コンテンツの由来と真正性に関する連合

コンテンツ認証情報

C2PA技術仕様書

2.2、2025年5月1日：

目次

本資料は「現状有姿」で提供されます。当事者は、本資料に関連する商品性、非侵害性、特定目的適合性、または所有権に関する黙示の保証を含む、あらゆる保証（明示、黙示その他を問わず）を明示的に否認します。本資料の実施またはその他の使用に関する全リスクは、実施者および使用者が負うものとします。いかなる場合においても、当事者は、本成果物またはその適用契約に関連するいかなる種類の請求原因に基づくものであれ、契約違反、不法行為（過失を含む）、その他を問わず、利益の損失、または間接的、特別、付随的、結果的損害を含むいかなる性質の間接的損害についても、他の当事者に対して責任を負わないものとします。（過失を含む）、その他を問わず、また、他のメンバーがそのような損害の可能性について通知されていたか否かを問わない。

第1章 はじめに

1.1. 概要

デジタルコンテンツの流通速度が加速し、強力な制作・編集技術が普及する中、メディアの出所を明らかにすることは、透明性、理解、そして最終的には信頼を確保するために極めて重要です。

メディアへの信頼は今、前例のない試練に直面している。ソーシャルプラットフォームが複雑で不透明なアルゴリズムを通じて特定コンテンツの拡散力と影響力を増幅させる中、誤った帰属や文脈で流通するコンテンツが急速に広がる。意図せぬ誤情報であれ、偽情報による意図的な欺瞞であれ、信憑性に欠けるコンテンツは増加の一途をたどっている。

現在、自身の作品に関するメタデータを付加したい場合、プラットフォームを横断して安全かつ改ざん防止機能を備えた標準化された方法でこれを行うことはできません。信頼できる情報源からのこの情報がなければ、出版社や消費者はメディアの真正性を判断するための重要な文脈を欠くことになります。

プロヴェナンスは、地理的位置や技術へのアクセス度合いに関わらず、コンテンツ制作者や編集者が、資産がどのように作成され、どのように変更され、何が変更されたかについての情報を開示することを可能にします。資産が変更されるたびに、既存のプロバンスは保持され、新たな変更がプロバンスに追加されます。このように、プロバンスを伴うコンテンツは真正性の指標を提供し、消費者が改変されたコンテンツを認識できるようにします。このようなプロバンスには、変更内容とその変更のソースが含まれる可能性があります。クリエイター、出版社、消費者に対してプロバンスを提供するこの能力は、オンライン上の信頼を促進するために不可欠です。

出版社、クリエイター、消費者向けにこの問題を大規模に解決するため、コンテンツの由来と真正性連合（C2PA）は、コンテンツの由来と真正性を提供する技術仕様を開発しました。本仕様は、適切なセキュリティ要件を満たしつつ、幅広い個人や組織向けのデジタル由来対応アプリケーションの豊かなエコシステムを構築することで、デジタル由来技術のグローバルなオプトイン型採用を可能にするよう設計されています。

本仕様は、Project Origin AllianceやContent Authenticity Initiative（CAI）を含む業界専門家やパートナー組織から収集したシナリオ、ワークフロー、要件に基づいて策定され、現在も継続的に更新されています。また、規制機関や政府機関がデジタル・プロバンスの基準を確立するために本仕様を活用する可能性もあります。

1.2. 適用範囲

本仕様書は、C2PAアーキテクチャの技術的側面を記述するものである。C2PAは、定義された信頼モデルに基づいて信頼性を評価可能な、暗号的に検証可能な情報を保存・アクセスするためのモデルである。本文書には、改ざん防止機能の実現および信頼の確立を可能とするデジタル署名技術の使用を含む、C2PAマニフェストとその構成要素の作成および処理方法に関する情報が含まれる。

本仕様の開発に先立ち、C2PAは「指針原則」を策定しました。これにより、仕様がプライバシーとデータの個人管理を尊重する形で利用され、潜在的な悪用や誤用に対して批判的な視点で臨むことに注力し続けられるようになりました。例えば、本仕様の実装者には、メディア資産の作成者および発行者が特定のプロバンスデータを含めるかどうかを制御する機能を提供することが強く推奨されます。

重要

指針原則の「包括的目標」セクションより：

> C2PA仕様は、特定のプロバンスデータセットが「良い」か「悪い」かについての価値判断を提供するべきではなく、単にその中に含まれるアサーションが、関連する基盤資産と関連付けられて検証可能か、正しく形成されているか、改ざんされていないかについてのみ判断すべきである。

仕様が消費者のコンテンツアクセシビリティに悪影響を及ぼさないことが重要である。

C2PA の他の文書では、期待されるユーザー体験や脅威および危害のモデリングの詳細など、具体的な実装上の考慮事項について取り上げます。

1.3. 技術概要

C2PA情報は、資産の作成、編集操作、キャプチャデバイスの詳細、コンテンツへのバインディングなど、様々な領域をカバーする一連のステートメントで構成されます。これらのステートメント（アサーションと呼ばれる）は、特定の資産の由来を構成し、人間がその資産の信頼性に関する見解を向上させるために利用できる一連の信頼シグナルを表します。アサーションは追加情報と共に「クレーム」と呼ばれるデジタル署名付きエンティティにまとめられます。このクレームは署名者の署名認証情報を使用し、署名者に代わってクレーム生成者によってデジタル署名され、クレーム署名が生成されます。

これらの主張、クレーム、およびクレーム署名はいずれも、クレームジェネレータと呼ばれるハードウェアまたはソフトウェアコンポーネントによって、検証可能な単位であるC2PAマニフェスト（図1「C2PAマニフェストとその構成要素」参照）にまとめられます。資産のコンテンツクレデンシャルに保存される一連のC2PAマニフェストは、その来歴データを表します。

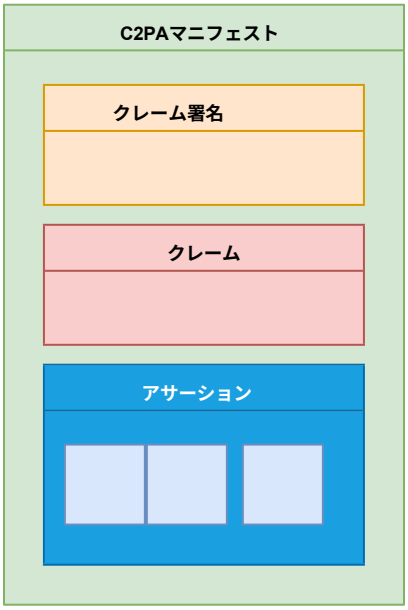

図1. C2PA マニフェストとその構成要素

1.3.1. 信頼の確立

C2PAにおける信頼決定の基盤となるのは、当社の信頼モデルにおいて、C2PAマニフェスト内のクレームに署名するために使用される暗号署名キーに関連付けられた署名者の身元です。C2PAマニフェストのクレーム署名は、信頼できるタイムスタンプと組み合わせることで、署名資格情報が有効かつ失効していない状態でクレームが署名されたかどうかを判断するための検証プロセスを無期限に実行できます。

1.3.2. 例

非常に一般的なシナリオとして、ユーザーがC2PA対応カメラ（またはスマートフォン）で写真を撮影する場合が挙げられます。この場合、カメラはマニフェストを作成します。このマニフェストには、カメラ自体に関する情報、画像のサムネイル、写真とマニフェストを結びつける暗号ハッシュなどのアサーションが含まれます。これらのアサーションはクレームに列挙され、デジタル署名された後、C2PAマニフェスト全体（図2「写真のC2PAマニフェスト例」参照）が出力JPEGに埋め込まれます。このC2PAマニフェストは永久に有効です。

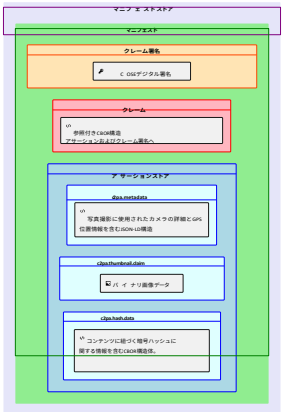

図2. 写真のC2PA マニフェスト例

C2PAバリデータなどのマニフェストコンシューマーは、まずデジタル署名と関連するクレデンシャルを検証することで、ユーザーが資産の信頼性を確立するのを支援します。また、各アサーションの有効性をチェックし、それらに含まれる情報と署名をユーザーに提示します。これにより、ユーザーはデジタルコンテンツの信頼性について情報に基づいた判断を下すことができます。

1.3.3. 設計目標

C2PAアーキテクチャの構築にあたっては、世界中の幅広いハードウェア・ソフトウェア実装で利用可能かつ誰もがアクセスできる技術とするため、明確な目標を設定することが重要でした。それらの目標は表1「C2PA設計目標」に示されています。

表1. C2PA設計目標

目標	説明
プライバシー	ユーザーが自身の情報のプライバシーを制御できるようにする。これには消費データやプロビネンスに記録された情報も含まれる
責任	消費者が資産の出所を特定できるようにする
拡張性	ウェブ上でのメディア作成・消費と同規模でのメディアプロビネンスの作成・消費・検証を可能にする
拡張性	将来のメタデータおよび認証情報プロバイダーが、C2PAからの入力や承認を必要とせずに情報を追加できるようにする
相互運用性	異なる実装が曖昧さなく相互に動作できることを保証する
ワークフロー全体の適用性	作成からその後のすべての修正、公開/配布に至るまで、複数のツールにわたって資産の出所情報を維持する
技術的ミニマリズム	既存の確立された技術に依存し、仕様内で必要最小限の新規技術のみを創出する
セキュリティ	消費者がプロバンスの完全性と出所を信頼できるように設計し、専門家による設計レビューを保証する
コンテンツの普遍性	文書を含む全ての一般的なメディアタイプに対して、出所の包含を可能にする
柔軟な局所性	オンラインとオフライン（資産のみ）の両方で、出所の保存と消費／検証を可能にする
グローバルな普遍性	世界中の関心を持つユーザーのニーズに応えるデザイン
アクセシビリティ	技術がWCAGなどの公認アクセシビリティ基準に準拠した方法で利用可能であることを保証する
危害と悪用	人権への脅威や脆弱なグループへの不均衡なリスクを含む、潜在的な危害を回避・軽減する設計
進化	これらの目標に対する仕様の継続的な見直しを行い、それらが優先事項であり続けることを保証する

第2章 用語集

2.1. 導入用語

2.1.1. アクター

C2PAエコシステムに参加する人間または非人間（ハードウェアまたはソフトウェア）。例：カメラ（キャプチャデバイス）、画像編集ソフトウェア、クラウドサービス、またはこれらのツールを使用する人物。

> **注記** 組織またはアクターのグループも、C2PAエコシステムにおけるアクターと見なされる場合があります。

2.1.2. クレーム生成者

資産に関するクレームおよびクレーム署名を生成する非人間（ハードウェアまたはソフトウェア）アクターであり、これにより資産に関連付けられたC2PAマニフェストが生成される。

2.1.3. 署名者

クレデンシャルの所有者であり、クレームの署名に使用される秘密鍵を保持する者。署名者はクレデンシャルの主体によって識別される。

2.1.4. マニフェストコンシューマー

関連付けられたC2PAマニフェストを持つ資産を消費し、C2PAマニフェストから出所データを取得することを目的とするアクター。

2.1.5. Validator

検証で記述されるアクションを実行する役割を担うマニフェスト消費者。

2.1.6. 操作

アクターがアセットに対して実行する操作。例：「作成」、「埋め込み」、「フィルターの適用」。

2.2. アセットとコンテンツ

2.2.1. デジタルコンテンツ

アセットのうち、画像のピクセルなどの実際のコンテンツ自体と、そのコンテンツを理解するために必要な追加の技術的メタデータ（例：カラープロファイルやエンコーディングパラメータ）で構成される部分。

2.2.2. アセットメタデータ

資産とそのデジタルコンテンツに関する非技術的な情報。

2.2.3. アセット

デジタルコンテンツ、アセットメタデータ、およびオプションでC2PAマニフェストを含むファイルまたはデータストリーム。

> **注記** この定義の目的上、「ファイル」の一般的な定義を拡張し、クラウドネイティブおよび動的に生成されたデータを含めるものとします。

2.2.4. 派生アセット

派生資産とは、既存の資産を起点として、そのデジタルコンテンツを変更する操作を施すことで作成される資産である。

> **例：**短縮されたオーディオストリームや、ページが追加された文書。

2.2.5. アセットのレンディション

デジタルコンテンツに「編集以外の変換」処理（例：再エンコードやスケーリング）が適用されたアセットの表現（アセットの一部として、または完全に新しいアセットとして）。

> **例：**画面解像度やネットワーク帯域幅を削減するために再エンコードされた動画ファイル。

2.2.6. 合成アセット

合成アセットとは、1つ以上の他のアセットからデジタルコンテンツの複数の部分や断片（素材と呼ばれる）を組み合わせて作成されるアセットです。既存のアセットから作成される場合は派生アセットの特殊なケースとなりますが、合成アセットは「白紙の状態」から作成される場合もあります。

> **例：**
>
> - 既存の動画クリップや音声セグメントを「白紙の状態」にインポートして作成された動画。
> - 別の画像をインポートし、元の画像の上に重ね合わせた画像。

2.2.7. 編集的変換

デジタルコンテンツの意図または意味、あるいはその両方を変更する変換の一種。

2.3. C2PAの核心的側面

2.3.1. アサーション

署名者によって作成された（または単にクレーム生成時に収集された）資産に関する声明を表すデータ構造。このデータはC2PAマニフェストの一部である。

2.3.2. クレーム

デジタル署名され改ざん防止機能を備えたデータ構造であり、資産に関する一連のアサーションと、コンテンツの拘束関係を表すために必要な情報を参照する。アサーションが削除された場合、その旨の宣言が含まれる。このデータはC2PAマニフェストの一部である。

2.3.3. クレーム署名

署名者が所有する秘密鍵を使用して作成されたクレーム上のデジタル署名。クレーム署名はC2PAマニフェストの一部である。

2.3.4. C2PAマニフェスト

1つ以上のアサーション（コンテンツバインディングを含む）、単一のクレーム、およびクレーム署名の組み合わせに基づく、資産*の出所*に関する情報の集合体。C2PAマニフェストはC2PAマニフェストストアの一部である。

注記	C2PAマニフェストは他のC2PAマニフェストを参照できます。

2.3.5. C2PAマニフェストストア

アセットに埋め込むことも、アセット外部に配置することも可能なC2PAマニフェストのコレクション。

2.3.6. コンテンツクレデンシャル

C2PAマニフェストを指す非技術的な推奨用語です。したがって、C2PAマニフェストストアはアセットのコンテンツ認証情報を表します。

コンテンツ認証情報はC2PA技術全体を指す場合もあり、基本的には複数名詞として扱われます。単一のC2PAマニフェストがコンテンツ認証情報であるのに対し、複数のC2PAマニフェストやより広範な普遍的概念はコンテンツ認証情報となります。

2.3.7. アクティブマニフェスト

C2PAマニフェストストア内のC2PAマニフェスト一覧において、検証可能なコンテンツバインディングのセットを持つ最後のマニフェスト。

2.3.8. 来歴

プロビネンスデータによって表される、資産の履歴およびアクターや他の資産との相互作用を理解する論理的概念。

2.3.9. 来歴データ

資産に対する一連のC2PAマニフェスト、および複合資産の場合はその構成要素。

> **注記** C2PAマニフェストは他のC2PAマニフェストを参照できます。

2.3.10. 真正性

デジタルコンテンツの特性であり、改ざんされていないことが暗号的に検証可能な一連の事実（出所データやハードバインディングなど）で構成される。

2.3.11. コンテンツバインディング

特定の資産に関連付けられた特定のC2PAマニフェストに、デジタルコンテンツをハードバインディングまたはソフトバインディングとして関連付ける情報。

2.3.12. ハードバインディング

資産全体またはその一部を一意に識別する、1つ以上の暗号ハッシュ。

2.3.13. ソフトバインディング

(a) フィンガープリントのように統計的に一意ではない、または (b) 識別されたデジタルコンテンツに不可視の透かしとして埋め込まれたコンテンツ識別子。

2.3.14. 信頼シグナル

マニフェスト消費者が資産の信頼性を判断する際に参考となる情報の集合体。これらは基本信頼モデルが依存する署名者に加えて存在する。

2.3.15. C2PA 信頼リスト

C2PAが管理するX.509証明書信頼アンカーのリスト。ハードウェアおよびソフトウェア署名者がクレーム署名に使用する証明書を発行する。

2.4. 追加条件

2.4.1. 耐久性のあるコンテンツ認証情報

耐久性のあるコンテンツ認証情報は、マニフェストリポジトリでの発見を可能にする 1 つ以上のソフトバインディングが存在するコンテンツ認証情報です。

2.4.2. フィンガープリント

デジタルコンテンツから計算可能な固有のプロパティの集合であり、そのコンテンツまたはそれに近い複製を識別する。

> **例：**アセットのメタデータが削除または破損した場合、そのアセットはC2PAマニフェストから分離される可能性があります。アセットのデジタルコンテンツのフィンガープリントを使用してデータベースを検索し、完全なC2PAマニフェストを持つアセットを復元できます。

2.4.3. 不可視透かし

資産のデジタルコンテンツに、実質的に人間の知覚を超えた方法で組み込まれた情報であり、例えば資産を一意に識別したり、C2PAマニフェストへの参照を保存したりするために使用できる。

2.4.4. 可視透かし

資産の出所に関する人間が認識可能な情報を保持する、デジタルコンテンツの知覚可能な構成要素。

2.4.5. マニフェストリポジトリ

C2PAマニフェストおよびC2PAマニフェストストアを格納可能なリポジトリであり、コンテンツバインディングを用いて検索可能である。

2.5. 概要

この図は、これら様々な要素がどのように組み合わさってC2PAアーキテクチャを構成しているかを示しています。

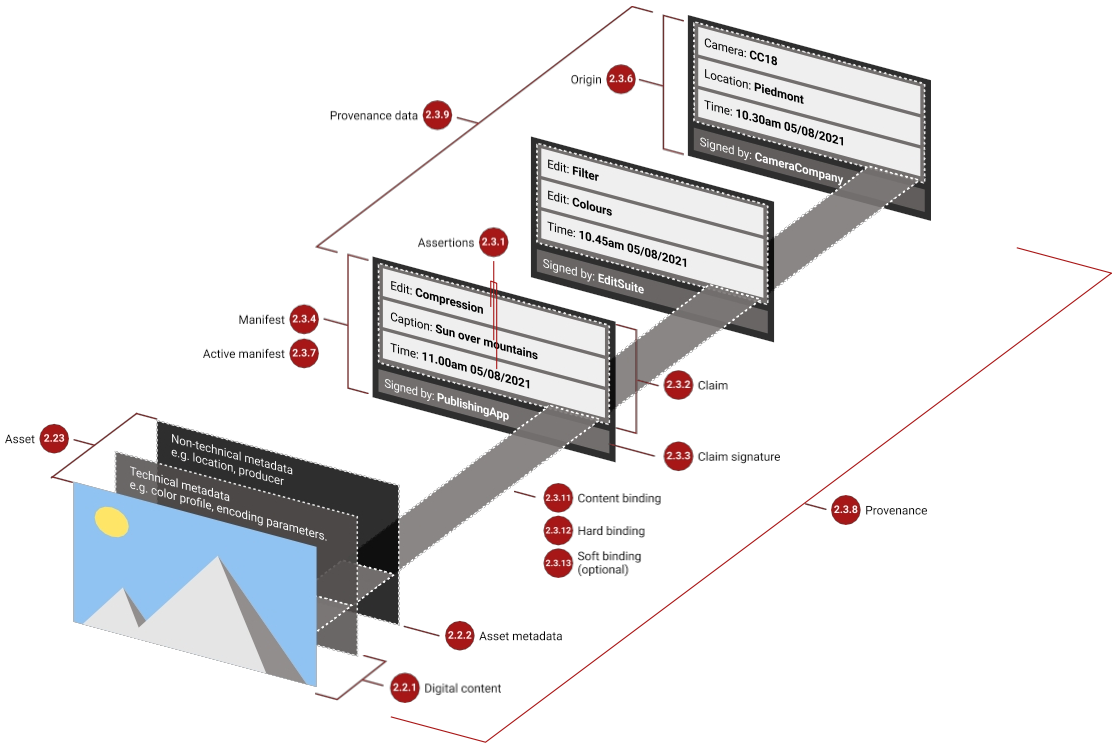

図3. C2PAの構成要素

第3章 規範的参照

3.1. コアフォーマット

- CBOR
- JSON
- JSON-LD
- JPEGユニバーサルメタデータボックスフォーマット（JUMBF）

3.2. スキーマ

- CDDL
- JSON Schema
- Dublin Core Metadata Initiative

3.3. デジタル署名および電子署名

- 暗号メッセージ構文 (CMS)
- インターネット X.509 PKI タイムスタンププロトコル
- インターネット X.509 公開鍵基盤証明書および証明書失効リスト (CRL) プロファイル用アルゴリズムおよび識別子
- インターネットX.509公開鍵基盤：DSAおよびECDSAのための追加アルゴリズムと識別子
- 米国安全ハッシュアルゴリズム
- オンライン証明書ステータスプロトコル（OCSP）
- JSON Web アルゴリズム (JWA)
- PKCS #1: RSA 暗号仕様バージョン 2.2
- Edwards曲線デジタル署名アルゴリズム（EdDSA）
- CBORオブジェクト署名および暗号化（COSE）
- COSEメッセージでのRSAアルゴリズムの使用
- インターネット X.509 公開鍵基盤で使用するための Ed25519、Ed448、X25519、および X448 のアルゴリズム識別子
- X.509証明書 汎用拡張キー使用法（EKU）: 文書署名用
- CBORオブジェクト署名および暗号化（COSE）：X.509証明書の伝送および参照のためのヘッダーパラメータ
- インターネット X.509 公開鍵基盤：X.509 証明書内のロゴタイプ

- JSON 先進電子署名 (JAdES)

3.4. 埋め込み可能フォーマット

- ISO ベースメディアファイルフォーマット (BMFF)
- PDF 1.7
- PDF 2.0
- JPEG 1
- JPEG XT、ISO/IEC 18477-3
- JPEG XL、ISO/IEC 18181-2:2024
- PNG
- SVG
- GIF
- ID3
- デジタルネガティブまたはDNG
- TIFF/EP
- TIFF v6)
- RIFF
- マルチピクチャーフォーマット (MPF)
- Open Font Format
- OpenType

3.5. その他

- eXtensible Metadata Platform (XMP)
- XMP の JSON-LD シリアライゼーション
- IPTC 写真メタデータ標準
- Exif
- UUID
- Uniform Resource Names (URNs)
- ユニバーサル一意識別子 (UUID)
- ISO 8601
- RFC 3339
- RFC 2326
- メディアフラグメント

- Web Annotation Data Model
- Brotli 圧縮データ形式
- RFC 5646、BCP 47

第4章 標準用語

キーワード「MUST」、「MUST NOT」、「REQUIRED」、「SHALL」、「SHALL NOT」、「SHOULD」、「SHOULD NOT」、

本文書における「推奨」、「非推奨」、「可能」、「オプション」は、大文字・小文字・混合表記を問わず、BCP 14、RFC 2119、およびRFC 8174に記述されている通りに解釈される。

第5章 バージョン管理

5.1. 互換性

コンテンツ認証情報の仕様が進化するにつれて、ボックスラベル、アサーション（およびそのフィールド）、クレーム、タイムスタンプなどの構造も進化してきた。新しいアサーションが追加され、既存のアサーションやクレームには追加フィールドを持つ新しいバージョンが存在する。さらに、一部の構造は非推奨となった。本仕様において構造が非推奨とマークされている場合、それはクレーム生成者がその構造（または値）を記述してはならないことを意味するが、検証者はそれを読み取るべきである。

クレーム生成器と検証器間の相互運用性を促進するため、クレーム生成器はクレーム生成に使用する仕様のバージョンを宣言する。クレーム生成者が仕様のバージョンを使用していると宣言する場合、それは当該資産のアクティブマニフェストがその仕様バージョンに準拠して生成されており、したがって付録C「*廃止に関する考慮事項*」の表19「構文のステータス」に記載された当該仕様バージョンにおける非推奨構文を含まないことを宣言している。

注記	本仕様は、この宣言の具体的な技術的手法を規定するものではないが、他の手段を通じてガイダンスが提供されることが期待される。

バリデータは少なくとも1つの仕様バージョンと互換性を持つべきであるが、追加のバージョンとも互換性を持つ場合がある。特定の仕様バージョンと互換性を持つバリデータは、そのバージョンでリストされている非廃止の構文をすべてサポートしなければならない。バリデータが、サポートしていない仕様バージョン（廃止済みまたは未定義のため）の構文を使用するマニフェストを検出した場合、廃止済み構文を無視し、その構文が存在しないものとしてマニフェストの残りを処理してもよい。あるいは、バリデータはマニフェスト全体を未知の由来を持つものとして扱い、適切に応じて `ingredient.unknownProvenance` または `manifest.unknownProvenance` ステータスコードを返すこともできる。

5.2. バージョン履歴

5.2.1. 2.2 - 2025年5月

このバージョンでは、仕様の技術的および編集上の変更に焦点を当て、2.1の新機能の一部を明確化するとともに、実装者からの要望に対応しています。仕様は、この分野における最新のベストプラクティスを反映するように更新されました。

- ソフトバインディング解決APIの新たな補足仕様を追加
- ソフトバインディングマニフェスト復元を示すため、成分アサーションに新フィールドを追加
- Android Motion Photosなどのマルチパートアセットのサポートを追加
- 更新マニフェストへのタイムスタンプおよび失効情報の追加をサポートし、タイムスタンプマニフェストを置き換える
- 「主張された署名作成時刻」のサポートを追加

- 新しいc2pa-kp-claimSigning EKUのサポートを追加
- C2PAトラストリストの使用制限：c2pa-kp-claimSigning EKUを持つ証明書に限定
- 導入 digitalSourceType 値 http://c2pa.org/digitalsourcetype/trainedAlgorithmicData（c2pa.trainedAlgorithmicDataに代わる）およびhttp://c2pa.org/digitalsourcetype/empty
- データボックスを埋め込みデータアサーションに置き換え
- 編集済みアサーションのクリアに関する追加ガイダンスを提供
- Trust Model における created_assertions および gathered_assertions の使用を明確化
- 役割に関する「署名者」と「クレーム生成者」の用語を明確化
- 各種ハードバインディングアサーションの変更と改善
  - c2pa.hash.data がアセットのクラシックメタデータセクションを除外できるようにする
  - c2pa.hash.boxesアサーションにおける除外サポートを追加
  - 更新マニフェストにおけるc2pa.hash.bmffアサーションの使用をサポート
- アサーションストアで許可されるJUMBFボックスを明確化
- 証明書失効処理を明確化
- タイムスタンプ検証の明確化
- アクションアサーションの改善と明確化
- ソフトバインディングアサーションの改善
- BMFFハッシュ図を明確化および正確化のために再作成
- マニフェストストア内のすべてのマニフェストを参照する必要があるという要件を削除

5.2.2. 2.1 - 2024年9月

本バージョンでは、コンテンツ認証情報のセキュリティと信頼性を向上させるため、仕様の技術的および編集上の変更に焦点を当てています。公開されているすべてのセキュリティ脆弱性に対処し、分野における最新のベストプラクティスを反映するよう仕様を更新しました。

- マニフェストとアセットの状態に関する明確な定義
  - 整形式のマニフェスト
  - 有効なマニフェスト
  - 信頼されたマニフェスト
  - 有効なアセット

- 廃止とバージョン管理の明確な定義とプロセス
- 新しい`c2pa` URN名前空間がマニフェストのラベル付けに追加されました！

        - 完全に仕様化されたABNFを含む
            - 新成分`v3`アサーション
    - 成分ベースのワークフローのより豊富なモデルをサポート。
    - `dataTypes` および `claimSignature` のサポート。
    - 他のアサーションとの整合性を高めるためフィールド名を変更。
    - 新しいステータス情報に対応する新しい検証ステータスフィールドを追加
    - `dc:title` および `dc:format` はオプションになりました
- 新しい `c2pa.hash.bmff.v3` アサーション
    - BMFF ベースの資産の固定および可変ブロックサイズのハッシュをサポート
- 新しいタイムスタンプマニフェスト
    - 特定の資産の「存在期間」を設定すること。
    - 更新マニフェストに類似するが、署名者がTSAとなる
- 標準的なRFC 3161タイムスタンプ処理のための改良モデル。
    - `sigTst2` および CTT タイムスタンプ
    - 新しいC2PA TSAトラストリストを導入
- 検証の改善
    - すべての標準アサーションに対する詳細な検証手順
    - 成分アサーション使用時の成分検証が必須化
    - より詳細なステータス情報を提供するための成分検証の拡張
    - 原材料における編集済み表示内容の検証支援
    - タイムスタンプの検証に関する詳細な要件の追加
    - データボックスおよびカスタムボックスへのハッシュ付きURIの検証を実施
    - 一意IDが重複するマニフェストの処理手順を定義
    - 検証プロセスにおける「孤立したマニフェスト」への対応
    - 新規検証ステータスコードを多数追加（新たな「情報提供用」コードタイプを含む）
- ドキュメントの改善とハッシュ手法のセキュリティ強化
    - BMFFベースのアセット
    - 「汎用ボックス」
    - ZIP

- フォーマット埋め込みセクションは独立した附属書に移動されました
    - JPEG-XLのサポートを追加しました

- ソフトバインディングの改善
- アクションアサーションの改善
    - 標準マニフェストにおいて、`c2pa.created` または `c2pa.opened` のいずれかが必須となりました
    - いくつかの新しい標準アクションタイプが追加されました
    - 単一のマニフェスト内で複数のアクションアサーションを記述可能になりました
    - アクションテンプレートの説明が改善され、より多くの例が追加されました
    - RFC 3339 に基づく関心領域
- 資産の各種固有識別子の種類/形式が明確化されました。
- JPEG Trustの互換性サポートで不足していた部分を追加
- すべてのCDDLを整理し、規範的表現を削除
- 編集上の改善を様々な領域で行いました
    - カスタムラベルをカスタム命名規則に再定義しました。
    - PDFへの埋め込み
    - ISOによる標準化に向けた文書の準備として、編集上の改善を多数実施

5.2.3. 2.0 - 2024年1月

このバージョンは、以前のバージョンから大きく方向転換したものです。「アクター」という用語の使用を減らし、もはや人間や組織を表すものではありません。バリデータ設定の信頼リストに加え、ハードウェアおよびソフトウェアに発行される証明書を対象とする新しいデフォルト信頼リスト「C2PA信頼リスト」を導入します。この理念的な変更により、仕様には以下の機能的な変更が生じました：

- 署名に使用できるのはX.509証明書のみとする。
- 検証および信頼モデルセクションの改善
    - 「整形式」および「有効な」C2PAマニフェストの概念を導入
    - 検証プロセスの様々な側面を明確化
- メタデータ処理の精緻化
    - 廃止予定のExif、IPTC、Schema.orgメタデータアサーションを削除しました
    - 新しい汎用的な「メタデータアサーション」概念を定義しました
    - `c2pa.metadata`では固定されたスキーマと値のみを許可
    - `c2pa.metadata`の作成プロセスが詳細に文書化されました
    - 関連する変更を反映するため、XMP処理セクションを刷新しました
    - マニフェスト外の標準メタデータ位置のハッシュ化に関する推奨事項を改善

- 「W3C Verifiable Credentials」セクションを削除

    - それおよびVCストアへの言及をすべて削除しました。
    - actionsアサーションからactorsフィールドを削除しました
    - 特定された人間をアサーションメタデータから削除
- 「トレーニングとデータマイニング」アサーションを削除
- 「推奨事項」アサーションを削除

さらに、仕様の様々な側面を改善するため、以下の変更が加えられました：

- クレームのバージョンをv2に変更
    - 廃止済みおよび未使用フィールドの削除
    - アサーションを `created_assertions` と `gathered_assertions` に分割
    - クレーム生成器は署名者である単一のみ許可
    - `claim-generator-info` に専用の `operating_system` フィールドを追加
- ボックスベースのハッシュ処理は、これをサポートするあらゆる形式において強く推奨されるようになりました
- 非推奨の`c2pa.hash.`bmffアサーションを削除しました
- 新しい`c2pa.watermarked`アクションを追加しました
- `c2pa.font`アクションは、単に`font`アクションとなりました
    - また、`c2pa.font.info` は `font.info` に統合されました
- CDDLスキーマのレンダリングを整理しました
- 規範的参照を更新し、将来のバージョンに関する注記を削除
- リンク修正を含む編集上の改善を多数実施

5.2.4. 1.4 - 2023年11月

- ZIPベースのフォーマット（例：EPUB、OOXML、ODF、OpenXPS）へのC2PAマニフェストの埋め込みをサポートしました
- マニフェストを特別なブロブボックスに圧縮できるようになりました。
- 複数ファイル（コレクション）のハッシュ化をサポート
- テキストベースのフォーマット（例：PDF、Office、EPUBなど）向けに新たな関心領域を追加
- Exif、IPTC、Schema.org、XMPをサポートする新しい`c2pa.`metadataアサーションを追加
- TIFF 埋め込みサポートの大幅な改訂
- OpenTypeおよびTrueTypeフォント内にC2PAマニフェストを埋め込む機能を追加
- PDFにおけるオブジェクトレベルのマニフェストのサポートを導入

- 埋め込みマニフェストに対するLinkヘッダーのサポートを拡張
- ボックスハッシュに関する問題を明確化

- タイムスタンプ、PKIStatus、文書署名 EKU を含む署名に関する問題を明確化
- Exif 3.0 との整合性を確保
- CDDLスキーマの改善
- 編集上の改善を多数実施

5.2.5. 1.3 - 2023年4月

- アクションアサーションの新しいv2バージョン（多くの新オプションをサポート）
- 新v2バージョンの成分アサーション（埋め込みデータ対応）
- 新しいアセット参照およびアセットタイプアサーション
- マニフェスト内に任意のデータを保存するための新しいデータボックス
- より包括的なバイト範囲ハッシュを実現する新汎用ボックスハッシュ手法
- 各種アサーションに適用可能な新「関心領域」データ構造
- 検証者がEKUリストを設定していない場合のC2PA署名者向け代替デフォルトEKUとして文書署名EKUを追加
- C2PAで使用するための新しいdigitalSourceTypeフィールドを追加
- 多くの新フォーマットに対応しました：MPF、WebP、AIFF、AVI、GIF
- エンティティ図を更新し、バージョン1.0以降の追加内容を反映
- X.509証明書用のCOSEヘッダー定義をRFC 9360に更新
- PDF埋め込みに関するガイダンスとPDF署名との関係を更新
- JUMBFハッシュおよびJUMBFボックストグルに関する情報を更新
- BMFFハッシュのv1を非推奨化
- C2PAマニフェストにおけるJUMBF保護ボックスの使用方法を明確化
- COSE署名にはすべての中間X.509証明書を含めるというC2PA固有の要件を明確化
- タイムスタンプは有効期限なく有効であることを明確化
- 編集上の改善が多数!!

5.2.6. 1.2 - 2022年10月

- DNGまたはTIFFにC2PAマニフェストを埋め込む方法の詳細を追加
- Actionsに新しいdigitalSourceTypeフィールドを追加
- IPTC動画メタデータに対応するため、`stds.iptc.photometadata` → `stds.iptc`に変更
- オプションフィールド追加時のアサーションのバージョン管理を明確化

5.2.7. 1.1 - 2022年9月

- ソルト付きボックスハッシュをサポートするメカニズムを定義
- 新しい`c2pa.hash.bmff.`v2アサーション。セキュリティ向上のためハッシュモデルを変更
- クレームに対するアサーションメタデータの有効化
- `claim_generator_hints` を `claim_generator_info` に置き換え
- エンドースメントの概念をサポートする新アサーションを追加
- `c2pa.`actionsアサーションの改善
- すべてのエラー＆ステータスコードに`c2pa`を接頭辞として付与
- W3C VCの編集メカニズムを定義
- 証明書内のEKUの検証を明確化
- 技術的変更を反映するため検証アルゴリズムを改訂
- CDDLおよびJSONスキーマの規範的テキストとの整合のための修正
- 図表の変更を反映した改訂
- 各種編集上および誤植の修正
- 規範的参照の更新（JUMBF および W3C VC データモデルを含む）

5.2.8. 1.0 - 2021年12月

- 初回リリース

第6章 断言

6.1. 一般

システム内で資産を作成または処理する主体が使用する各クレーム生成器は、資産がいつ、どこで、どのように生成または変換されたかに関する一つ以上のアサーションを作成または組み立てるものと期待される。アサーションとは、資産に関する宣言を表すラベル付きデータであり、通常（必須ではないが）CBORベースの構造で記述される。これらのアサーションの一部は人間が生成した情報（例：アクセシビリティのための代替テキスト）を含み、他は生成した情報を提供する機械（ソフトウェア／ハードウェア）から得られる（例：カメラの種類）。

アサーションの例：

- メタデータ（例：メーカーやレンズなどのカメラ情報）；
- アセットに対して行われた操作（例：トリミング、カラー補正）；
- アセットまたはその構成要素のサムネイル；
- コンテンツバインディング（例：暗号ハッシュ）。

特定のアサーションは、後続のクレームによって編集される場合があります（セクション 6.8「アサーションの編集」を参照）。ただし、クレームの一部として一度作成されたアサーションは、変更することはできません。

6.2. ラベル

6.2.1. 名前空間

C2PAデータ構造内の文字列値は、ピリオド(`.`)を区切り文字として名前空間に整理できる。C2PA名前空間`c2pa`は、本仕様で定義される文字列値の先頭に置かなければならない。エンティティ固有の名前空間は、Javaパッケージの定義と同様に、エンティティのインターネットドメイン名で始まるものとする（例：`com.litware`、`net.fineartschool`）。

エンティティ固有のネームスペースのピリオド区切りコンポーネントは、以下のABNF（ネームスペースのABNF）で定義されるPOSIXまたはCロケールで指定される変数命名規則（`[a-zA-Z0-9][a-zA-Z0-9_-]*`）に従うものとする。

名前空間のABNF

```
qualified-namespace = "c2pa" / entity
entity = entity-component *( "." entity-component )
entity-component = 1( 数字 / アルファベット ) *( 数字 / アルファベット / "-" / "_" )
```

6.2.2. ラベル命名

各アサーションには、C2PA仕様または外部エンティティによって定義されたラベルが付与されます。これらのラベルは、前項で説明した名前空間を持つ文字列、またはエンティティによって定義されます。最も一般的なラベルはc2pa名前空間で定義されますが、規約に従う任意の名前空間を使用可能です。ラベルは単純な整数増分方式（例：`c2pa.actions.v2`）でバージョン管理されます。バージョンが指定されない場合、`v1` と見なされます。公開されているラベルの一覧は第18章「*C2PA標準アサーション*」に記載されています。

> **注記**
>
> 本ドキュメントの旧版では、確立された標準規格向けの名前空間も規定していた。
> しかし、これはエンティティ固有のネームスペース（例：`org.iso`、`org.w3`）を介して単純に提供される方式に取って代わられました。

アサーションラベルのABNF

```
namespaced-label = qualified-namespace label qualified-
namespace = "c2pa" / entity
エンティティ = エンティティコンポーネント *( "." エンティティコンポーネント )
エンティティコンポーネント = 1( 数字 / アルファベット ) *( 数字 / アルファベット / "-" / "_" )ラベル
= 1*( "." ラベルコンポーネント )
ラベルコンポーネント = 1( 数字 / アルファベット ) *( 数字 / アルファベット / "-" / "_" )
```

ラベルのピリオド区切りコンポーネントは、POSIX または C ロケールで指定される変数命名規則 (`[a-zA-Z][a-zA-Z0-9_-]*`) に従う。ただし、繰り返しアンダースコア文字 (`___`) の使用は、同一タイプの複数アサーションのラベル付けに予約されている。

6.3. バージョン管理

アサーションのスキーマを変更する場合、下位互換性を保つ方法で実施すべきである。これは、新規フィールドの追加や既存フィールドの非推奨化（読み取りは可能だが書き込み不可）を意味する。既存フィールドは削除してはならない。ラベルは増分されたバージョン番号で構成される。例：`c2pa.action`（非推奨）から `c2pa.action.v2` への移行。

オプションフィールドの追加は下位互換性を維持したまま行えるため、バージョン番号を変更せずに既存のアサーションスキーマに追加することが可能です。

C2PA標準アサーションの非推奨フィールドは、第18章「*C2PA標準アサーション*」で明示される。クレーム生成者は、アサーション作成時に非推奨フィールドへデータを挿入してはならない。

後方互換性のない変更が必要な場合、ラベルのバージョン番号を増加させる代わりに、アサーションに新しいラベルを付与する。

> **注記**
>
> たとえば、`c2pa.ingredient` は架空の `c2pa.component` に変更することができます。

6.4. 複数のインスタンス

同一タイプのアサーションが同一マニフェスト内に複数存在することは可能ですが、アサーションはクレームによって

ラベルによって参照されるため、アサーションラベルは一意である必要があります。これは、ラベルに二重アンダースコアと単調増加するインデックスを追加することで実現されます。例えば、マニフェストに `c2pa.metadata` 型のアサーションが 1 つだけ含まれる場合、アサーションラベルは `c2pa.metadata` となります。マニフェストにこのタイプの アサーションが 3 つ含まれる場合、ラベルは `c2pa.metadata`、`c2pa.metadata__1`、`c2pa.metadata__2` となります。

ラベルにバージョン番号が含まれる場合、そのバージョン番号はラベル自体の一部となります。したがって、複数のインスタンスが存在する場合、インスタンス番号はラベルの後に引き続き付加されます。例：`c2pa.ingredient.v2__2`。

6.5. スキーマ検証

本文書で提供されるスキーマ、およびC2PAウェブサイトからダウンロード可能な機械可読スキーマは、読み書きされる構文を理解するための補助としてのみ使用してください。バリデータがスキーマ検証を実行することは必要ではなく、推奨もされません。

6.6. アサーションストア

マニフェスト内のクレームによって参照されるアサーションの集合は、*アサーションストア*と呼ばれる論理的構造体にまとめられる。アサーションおよびアサーションストアは、セクション11.1「JUMBFの使用」に記載された方法で保存されるものとする。特に、クレームの `created_assertions` または `gathered_assertions`（ただし `redacted_assertions` は除く）で参照される各アサーションは、当該クレームと同じC2PAマニフェスト内に配置されたアサーションストアに存在しなければならない。

各マニフェストには単一のアサーションストアが存在する。ただし、1つの資産に複数のマニフェストが関連付けられる場合があり、それぞれが特定のアサーションの系列を表すため、1つの資産に関連付けられるアサーションストアは複数存在し得る。

6.7. 埋め込みデータと外部保存データ

一部の断言データは、そのサイズや使用頻度の低さから、外部でホストされる場合があります。このようなデータは断言ストアに埋め込まれず、代わりにURIによって参照されます。これはクラウドデータ断言を通じて実現されます（セクション18.11「クラウドデータ」参照）。埋め込みアサーションデータとは異なり、クラウドデータはマニフェスト検証の一部として取得・検証されず、セクション15.10「アサーションの検証」で説明される別の検証ルールセットに基づき、アプリケーションが特に必要とする場合にのみ取得・検証されます。

6.8. アサーションの削除

アセットに埋め込まれたマニフェストに存在するアサーションは、そのアセットが成分として使用される際に、当該アセットのマニフェストから削除される場合があります。このプロセスは編集と呼ばれます。

編集処理には、マニフェストのアサーションストアからアサーション全体を削除する方法、またはラベル付きアサーションコンテナは保持しつつ、そのアサーション内のJUMBFコンテンツボックスを単一のUUIDコンテンツボックスに置き換える方法があります。置き換え後の `UUID` コンテンツボックスのIDフィールドには、`CAA98EEE-9D4D-F80E-86AD-4DFFCA263973`（C2PA編集用UUIDと呼ばれる）という値を設定します。

さらに、DATAフィールドにはゼロ（バイナリ0x00値）のみが含まれる。

さらに、削除された事実を示す記録を、クレームの redacted_assertions フィールドに、編集済みアサーションへの URI 参照として追加する。また、クレーム生成者は、セクション 18.14.4.7「パラメータ」で説明されているように、redacted フィールドを持つ c2pa.redacted アクションアサーションを追加することが強く推奨される。

C2PAマニフェストを参照する成分アサーションを編集する場合、編集後に当該マニフェストへの他の参照が残存しないときは、関連するマニフェストをC2PAマニフェストストアから削除しなければならない。

注記

各アサーションのURI参照にはアサーションラベルが含まれるため、どのタイプのアサーションであるかも判明する。（例：サムネイル、メタデータなど）が削除されたかどうかも把握できます。これにより、人間と機械の両方がルールを適用し、削除が許容可能かどうかを判断できます。

アサーションの編集がデジタルコンテンツの変更を必要としない限り、コンテンツの変更がないことを示すため、編集内容を記録する更新マニフェストを使用しなければならない。

クレーム生成者は、c2pa.actions または c2pa.actions.v2 のラベルを持つアサーションを編集してはならない。このアサーションタイプは、資産の履歴を理解する上で不可欠な情報を表している。また、コンテンツアサーションへのハードバインディング（c2pa.hash.data、c2pa.hash.boxes、c2pa.hash.collection.data、c2pa.hash.bmff.v2（非推奨）、c2pa.hash.bmff.v3）も編集してはならない。これらのアサーションは、資産の完全性を判断するために必要である。

NOTE

廃止予定の成分アサーション（c2pa.ingredient または c2pa.ingredient.v2）のいずれかを通じて参照される成分マニフェストにおいてアサーションが編集された場合、
そのアサーションの検証は失敗します（セクション15.11.3「成分アサーションの検証」で説明）。なぜなら、セクション15.11.3.3.1「クレーム署名ハッシュ検証方法」で説明されているクレーム署名ハッシュ検証方法をサポートしているのは c2pa.ingredient.v3アサーションのみだからです。

6.9. アサーションにおける時刻の仕様

アサーション内の日付および/または時刻値のデフォルトの指定は、日付/時刻形式であり、CBOR ではタグ番号 0 (RFC 8949、3.4.1) としてシリアル化され、CDDL では型 tdate として表現されます。

署名時刻の主張を追加する際に説明したように、時刻が特殊なタイプのCBOR日付/時刻として表現されるケースが一つあります。

さらに、署名プロセスで説明されている標準的なタイムスタンプ形式を使用するタイムスタンプアサーションがあります。

日付と時刻の表現に異なるタイプが存在する理由は、既存の標準に基づいて、それぞれの特定のユースケースに最も適切な表現を可能にするためです。

第7章 データボックス

重要

このセクションは、歴史的な目的のために残されています。データボックスの概念は標準のJUMBF埋め込みファイルコンテンツタイプボックスを使用してデータを格納する標準アサーションに置き換えられ、非推奨となりました。詳細については、[_data_box]を参照してください。

7.1. 一般

データボックスは、アサーションから参照される任意のデータをC2PAマニフェストに含める手段を提供します。これにより、データをバイナリ文字列としてアサーションのフィールドに直接埋め込む必要がなくなります。これらのデータボックスはデータボックスストアに配置され、それぞれが単一のCBORコンテンツタイプボックス（`cbor`）となります。

データボックスのデータは、データフィールドの値として直接提供されます。このフィールドはbstr型であるため、あらゆるバイナリデータを提供できます。データのタイプは`dc:format`フィールドを用いて、標準的なIANAメディアタイプで識別する必要があります。

注記

IANA structured suffixes (https://www.iana.org/assignments/media-type-structured-suffix/media-type-structured-suffix.xhtml）も、`dc`:format フィールドの値としてサポートされる。`dc:format`フィールドの値としてもサポートされています。

場合によっては、データの形式や使用方法をより明確にするため、data_typesフィールドの値として1つ以上のアセットタイプを指定する必要があることもある。

データボックスはラベル `c2pa.data` を持ち、複数のインスタンスに関してはアサーションラベルの規則に従う。

7.2. スキーマと例

このタイプのスキーマは、CDDL for data box の CDDL 定義における `data-box-map` ルールによって定義されます。

データボックス用CDDL

```
; 任意のデータを格納できるボックス

data-box-map = {
   "dc:format": format-string, ; データのIANAメディアタイプ "data" : bstr, ; 任意のテキスト/バイナリデータ
   ? "data_types": [1* $asset-type-map], ; データのタイプに関する追加情報
}
```

第8章. 一意の識別子

8.1. C2PAマニフェストと資産の固有識別

各C2PAマニフェストは、`c2pa` URN名前空間からのRFC 8141に準拠した一意のリソース名（URN）によって一意に識別され参照される。また、C2PAアセットは、そのアクティブなマニフェストの`c2pa` URN値によって一意に識別される。C2PA URNのABNFは、C2PA URNのABNFで記述される。

`c2pa` URNは、以下の順序で構成され、各セクション間にコロン(`:`)を挟む、2つの必須コンポーネントと2つのオプションコンポーネントから成るものとする。

- URN識別子（`urn:c2pa`）：必須。
- UUID v4（RFC 9562セクション4に準拠した文字列表現）：必須。
- クレーム生成器識別子文字列：オプション。
- バージョンと理由の文字列（以下で説明）：オプション。

存在する場合、「クレーム生成者識別子」文字列は、ASCII 範囲（RFC 20 による）の 32 文字以下で構成され、制御文字（RFC 20、5.2）またはグラフィック文字（RFC 20、5.3）ではないものとします。

「バージョンと理由」文字列が存在する場合、それは正の整数、アンダースコア（_）、そして別の正の整数で構成されるものとする。これらの各値の詳細と使用方法は、競合によるバージョン管理マニフェストで記述されている。さらに、「バージョンと理由」文字列が存在する場合、「クレーム生成器識別子」文字列も存在しなければならないが、空でもよい。

C2PA URN のABNF

```
c2pa_urn = c2pa-namespace UUID [claim-generator [version-reason]]c2pa-namespace =

"urn:c2pa:"

; この定義はRFC 9562より引用 UUID      = 4hexOctet "-"
                2hexOctet     "-"
                2hexOctet     "-"
                2hexOctet     "-"
                6hexOctet
hexOctet = HEXDIG HEXDIG DIGIT
           = %x30-39
HEXDIG      = 数字 / "A" / "B" / "C" / "D" / "E" / "F"

; ASCII 文字（制御文字およびグラフィック文字を除く）可視文字（スペースを除く） =
%x21-7E / %x80-FF

; クレーム生成識別子は、0から32文字の可視文字（スペースを除く）の文字列である
; これは空文字列が有効であることを意味する
claim-generator = ":" claim-generator-identifier
claim-generator-identifier = 0*32visible-char-except-space

; version-reason は正の整数で構成される文字列
; アンダースコアと正の整数が続く
```

```
version-reason = ":" version "_" reasonversion = 1*DIGIT
reason = 1*DIGIT
```

例:

- urn:c2pa:F9168C5E-CEB2-4FAA-B6BF-329BF39FA1E4
- urn:c2pa:F9168C5E-CEB2-4FAA-B6BF-329BF39FA1E4:acme
- urn:c2pa:F9168C5E-CEB2-4FAA-B6BF-329BF39FA1E4:acme:2_1
- urn:c2pa:F9168C5E-CEB2-4FAA-B6BF-329BF39FA1E4::2_1

注記 この仕様の以前のバージョンでは、RFC 9562、UUID URN を使用し、クレームの識別子はURNの先頭にジェネレータを配置すること。しかし、これはRFC 9562（UUID）にもRFC 8141（URN）にも準拠していないことが判明した。

このc2pa URN識別子は、派生資産や合成資産の構成要素として資産を識別する場合など、C2PA対応ワークフローの様々な部分で使用されます。

8.2. 競合によるバージョン管理マニフェスト

識別子の競合により、C2PAマニフェストの再ラベル付けが必要となる状況が発生する場合があります。例えば、クレーム生成者が既に成分マニフェストをアセットのC2PAマニフェストストアに追加した後、別の成分を追加し、そのマニフェストストアに同じラベルを持つマニフェストが存在する場合です。ただし、後者のマニフェストは、例えばアサーション値のいずれかが操作されたために異なる内容となっています。このような場合、変更されたバージョンの成分マニフェストをアセットのC2PAマニフェストストアにコピーし、再ラベル付けを行う必要があります。

マニフェストの再ラベル付け:

- 現在の URN に「クレーム生成器識別子文字列」が含まれていない場合、クレーム生成器は : を付加する。
- いずれの場合も、クレーム生成者は URN の末尾に : を付加し、その後ろに 1 から始まる単調増加整数、アンダースコア (_)、そして再ラベル付けの理由を示す以下のリストから選択した整数を付加しなければならない。
    - 1: 他のC2PAマニフェストとの競合

例えば、クレーム生成者が競合によりC2PAマニフェストを2度目に再ラベル付けする必要がある場合、付加される文字列は :2_1 となる。

8.3. 非C2PA資産の識別

C2PAマニフェストを含まない資産を扱う場合、その資産にXMP仕様パート2の2.2で定義されるxmpMM:DocumentIDおよび/またはxmpMM:InstanceIDの値を含む埋め込みXMPが含まれている場合、それらの値を資産の識別子として使用するものとする。

C2PAマニフェストを含まず、埋め込みXMPも含まないアセットを扱う場合、クレーム生成器は任意の方法で一意の識別子を提供できます。

8.4. URI リファレンス

8.4.1. 標準URI

マニフェスト内の情報への参照は、資産内部に保存されている場合（つまり埋め込み）でも、資産外部に保存されている場合（例：クラウド内）でも、ISO 19566-5:2023、C.2 で定義される JUMBF URI 参照を介して参照されなければならない。これらの URI は通常、`hashed_uri` または `hashed_ext_uri` データ構造の一部として使用される。

圧縮されたマニフェストを参照する場合、JUMBF URIはブロブボックスに関する情報を含んではならないが、マニフェストへのURIはマニフェストが圧縮されていないものとして扱われる。これは、URIがc2maまたはc2umボックスのラベルを含み、`c2cm`ボックスのラベルを含まないことを意味する。さらに、圧縮されたマニフェストへのURI参照には、ブロブボックスのラベルを含めてはならず、圧縮されたマニフェスト自体のラベルのみを含めるものとする。

内部のJUMBF URI参照を解決する際、複数の子ボックスが同一のラベルを持つことによりパス内のいずれかのラベルが曖昧な場合、バリデータはその参照を未解決として扱うものとする。

8.4.2. ハッシュ付きURI

8.4.2.1. 埋め込み

`hashed_uri` は、URI が同じ C2PA マニフェストストアに埋め込まれたものに対する場合に使用されます。

この仕様は、CDDL 定義を使用するスキーマ向けに、同等の `hashed-uri-map` データ構造（ハッシュ付き URI 用の CDDL）を提供します。

ハッシュ付きURI用CDDL

```
; 同一JUMBF内のURLへの参照とそのハッシュを格納するために使用されるデータ構造。ここではソケット/プラグを使用し、マップを同一ファイル内で定義せずに個別のファイルでハッシュドURIマップを利用できるようにする
$hashed-uri-map /= {
   "url": jumbf-uri-type, ; JUMBF URI参照
   ? "alg": tstr .size (1..max-tstr-length), ; このクレーム内の全ハッシュ計算に使用される暗号ハッシュアルゴリズムを識別する文字列。C2PAハッシュアルゴリズム識別子リストから取得される。このフィールドが存在しない場合、ハッシュアルゴリズムは包含構造体で定義されたものから取得される。両方が存在する場合、この構造体のフィールドが使用される。いずれの場所にも値が存在しない場合、この構造体は無効となる。デフォルト値は存在しない。
   "hash": bstr, ; ハッシュ値を含むバイト文字列
}

; CBORヘッダー(#)とテール($)は正規表現で導入されるため明示的に不要 jumbf-uri-type /= tstr .regexp "self#jumbf=[\\w\\d\/][\\w\\d\\.\/:-]+[\\w\\d]"
```

アサーションストアは、それらが参照するクレームと同じC2PAマニフェストボックス内に配置される必要があるため、

`self#jumbf` URIのみが許可される。これらの`self#jumbf` URIは、C2PAマニフェストストア全体を基準とする相対URIである場合、

この場合、URI は `/` (U+002F, スラッシュ) で始まる必要があります。または、現在の C2PA マニフェストに対する相対パスでも構いません。URI は `..` (U+002E, フルストップのペア) のシーケンスを含んではなりません。

例1. `self#jumbf` URIの例

以下の例は有効な `self#jumbf` URI です：

- `self#jumbf=/c2pa/urn:c2pa:F095F30E-6CD5-4BF7-8C44-CE8420CA9FB7/c2pa.assertions/c2pa.thumbnail.claim` はストア全体を基準とする相対パスです（`/` で始まるため）。
- `self#jumbf=c2pa.assertions/c2pa.thumbnail.claim` は、URI を含むボックスのマニフェストに対する相対パスとなります。

8.4.2.2. 外部

C2PAマニフェストストアの外部に存在するリソースを参照する場合、`hashed-ext-uri`-mapデータ構造が使用されます。これは`hashed-uri`の変種であり、`self#jumbf`ではなく外部URIを参照します。`hashed`-ext-uriデータ構造は、以下のCDDLにおける`hashed`-ext-uri-mapルールで定義されています：

ハッシュ外部URI用CDDL

```
; 外部URLとそのハッシュへの参照を格納するために使用されるデータ構造。
; ここではソケット/プラグを使用し、個々のファイル内でハッシュ付き拡張URIマップが使用できるようにしています
; マップを同じファイル内で定義せずに
$hashed-ext-uri-map /= {
   "url": ext-url-type, ; http/https URI参照
   "alg": tstr .size (1..max-tstr-length), ; このURIのデータに対するハッシュ計算に使用される暗号ハッシュアルゴリズムを識別する文字列。
C2PAハッシュアルゴリズム識別子リストから取得。他のタイプのalgフィールドとは異なり、ここでは必須フィールド。
   "hash": bstr, ; ハッシュ値を含むバイト文字列
   ? "dc:format": format-string, ; データのIANAメディアタイプ
   ? "size": size-type, ; データのバイト数
   ? "data_types": [1* $asset-type-map], ; データタイプに関する追加情報
}

; CBORヘッダー(#)とテール($)は正規表現で導入されるため、明示的に必要ない
ext-url-type /= tstr .regexp "https?:\/\/[-a-zA-Z0-9@:%._\\+~#=]{2,256}\\.[a-z]{2,6}\\b[-a-zA-Z0-9@:%_\\+.~#?&//=]*"
```

重要

一般的な慣行に従い、転送中のデータのプライバシー保護のため、アサーションデータの取得には `https` スキームの使用が推奨されます。ただし、データの完全性は hash フィールドによって保護されており、このプライバシーが常に必要とは限らないため、`http` も許可されています。外部 URI を含むマニフェストの作成者は、必要に応じてスキームを選択してください。

オプションの`dc:format`フィールドが存在する場合、http(s)ヘッダーの`Content-Type`フィールドの代替として機能します。存在する場合、このフィールドはコンテンツネゴシエーション/リクエスト時に取得される必須フォーマットとして使用されます。

場合によっては、データの形式と使用法をより明確にするため、data_typesフィールドの値として1つ以上のアセットタイプを指定する必要が生じることもあります。

オプションのサイズフィールドも提供されており、取得するデータのサイズを指定できます。これはハッシュに加えて、バリデータがヒントとして活用できる場合があります。

注記 ダウンロードと検証のいずれか、あるいは両方を実行するかどうかに関する情報を提供するために使用できます。

8.4.2.3. JUMBFボックスのハッシュ化

JUMBFボックス（例：アサーションボックスやデータボックス）へのURI参照を作成する際、ハッシュ処理は構造体のJUMBFスーパーボックスの内容に対して行わなければならない。このスーパーボックスには、JUMBF記述ボックスと、その内部にある全てのコンテンツボックスが含まれる（ただし、構造体のJUMBFスーパーボックスヘッダーは含まない）。

注記 ハッシュ処理の詳細については、セクション13.1「ハッシュ処理」を参照のこと。

最新版のJUMBF（ISO 19566-5:2023）に記載され、図4「`c2pa.actions`アサーションの例」に示すように、新しいプライベートフィールドが任意のJUMBF記述ボックスの一部として存在し得る。本C2PA仕様では、C2PAソルトを以下の構成要素からなる標準ボックスとして定義するプライベートフィールドとする：

- ボックス長（LBox：4バイトのビッグエンディアン符号なし整数）
- ボックスタイプ（TBox：4バイトビッグエンディアン符号なし整数、値は`c2sh`（C2PAソルトハッシュ用））
- ペイロードデータ（長さ16バイトまたは32バイトのランダムに生成されたバイナリデータで構成される）。

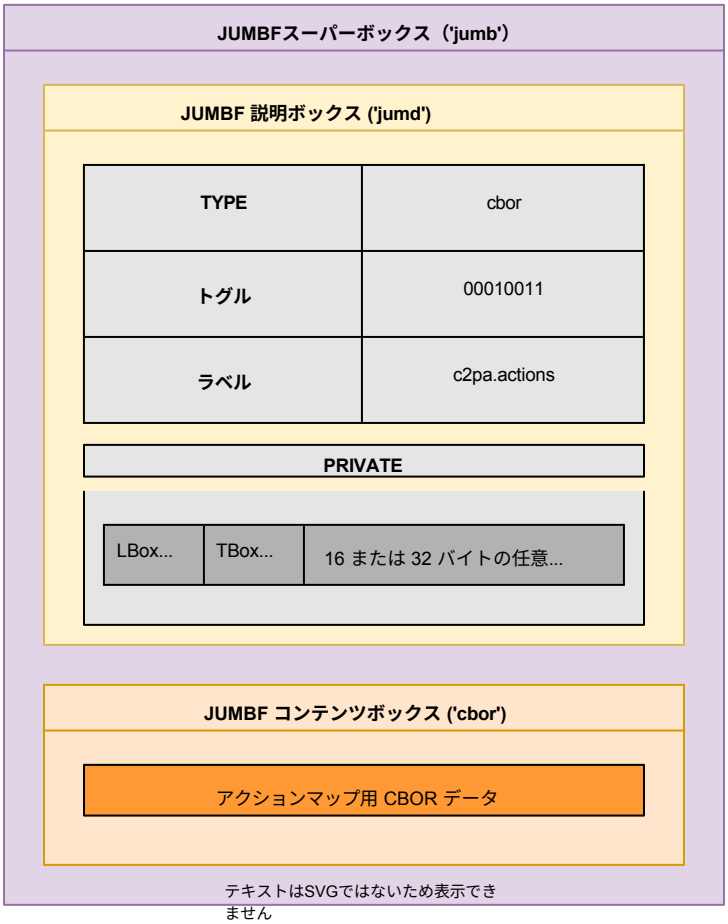

図4. `c2pa.actions` アサーションの例

第9章 コンテンツへのバインディング

9.1. 概要

標準的なC2PAマニフェストの重要な側面は、資産の一部を一意に識別できる「コンテンツバインディング」と呼ばれる1つ以上のデータ構造の存在です。C2PAがサポートするバインディングには、ハードバインディングとソフトバインディングの2種類があります。ハードバインディング（暗号バインディングとも呼ばれる）は、検証者が以下のことを保証できるようにします：(a) このマニフェストがこの資産に属すること、および (b) 資産が変更されていないこと。これは、この資産のみに一致し、他の資産（派生資産やレンディションを含む）には一致しない値を決定することで実現されます。ソフトバインディングは、資産の生のビットではなく、デジタルコンテンツから計算されます。ソフトバインディングは、派生資産や資産のレンディションを識別するのに有用です。

単一のマニフェストには、ハードバインディングを定義するアサーションを1つ以上含めることはできないが、ソフトバインディングを定義するアサーションを0個以上含めることができる。

9.2. ハードバインディング

9.2.1. バイト範囲を用いたハッシュ処理

改ざん検出に使用できる最も単純なハードバインディングは、セクション13.1「ハッシュ処理」で説明されているように、資産の全バイトまたは一部に対して暗号学的ハッシュアルゴリズムを適用する方法である。この手法はあらゆるタイプの資産に適用可能だが、ボックスベースのハッシュ処理のいずれかの形式をサポートしないフォーマットでのみ検討すべきである。

この形式のハードバインディングを使用する場合、データハッシュアサーションを用いてハッシュ処理されるバイト範囲（および処理されない範囲）を定義します。データハッシュアサーションはバイト範囲を定義するため、アセットが単一のバイナリである場合でも、複数のチャンクや部分で構成されている場合でも柔軟に使用できます。

9.2.2. 汎用ボックスハッシュを用いたハッシュ処理

アセットのフォーマットがJPEG、PNG、GIF、またはここに列挙されているその他の非BMFFベースのボックスフォーマットである場合、一般的なボックスハッシュアサーションを使用すべきである。このアサーションは構造体の配列で構成され、各構造体は 1 つ以上のボックス（名前/識別子で指定）と、それらのボックスのデータ（およびそれらの間にファイル内に存在する可能性のあるデータ）をカバーするハッシュ、およびハッシュに使用されるアルゴリズムをリストします。

9.2.3. BMFF形式の資産のハッシュ処理

アセットが ISO BMFF ベースの場合は、ボックスベースのフォーマットに最適化されたハードバインディング（BMFF ベースのハッシュアサーションと呼ばれる）を代わりに使用することができます。

mdatボックスが単位として検証されるモノリシックなMP4ファイルアセットの場合、アサーションの検証はデータハッシュアサーション

とほぼ同一の方法で行われる。ハッシュ処理の対象となるバイト範囲（および対象外となる範囲）を定義するために、バイト範囲の代わりにボックス除外リストを使用するだけである。

フラグメント化されたMP4（fMP4）ファイルの場合、アサーション自体はセクションA.5「BMFFベースのアセットへのマニフェスト埋め込み」で規定される位置にあるチャンク固有のハッシュ情報と組み合わせるものとする。

9.2.4. コレクションのハッシュ化

C2PAマニフェストが単一のアセットではなくアセットのコレクションを参照するワークフローでは、コレクション内のアセットに対するハードバインディングを指定する方法として、コレクションデータハッシュアサーションを使用するものとする。

> **注記** たとえば、AI/MLモデルのトレーニングデータセットの各フォルダを記述するために、コレクションデータハッシュアサーションを使用できます。

9.2.5. アセットメタデータバインディング

クレーム生成器は、コンテンツバインディングから資産メタデータ（例：EXIFやXMPなどC2PAマニフェスト外のメタデータ）を除外できる。その場合、データハッシュアサーション、汎用ボックスハッシュアサーション、またはBMFFベースのハッシュアサーションに対して適用可能な除外メカニズムを使用するものとする。

> **注記** 除外されたアセットメタデータは署名者に帰属しない。

付録 B「*c2pa.metadata の実装詳細*」に記載され、署名者によってアサーション可能な共通メタデータアサーションでサポートされる資産メタデータの値は、すべてそのようなアサーションにコピーされ、C2PA マニフェストに含まれるべきである。

9.3. ソフトバインディング

9.3.1. 一般

ソフトバインディングは、デジタルコンテンツから計算されたフィンガープリントやデジタルコンテンツ内に埋め込まれた不可視の透かしといったソフトバインディングアサーションを用いて記述される。これらのソフトバインディングにより、基盤となるビットが異なっていてもデジタルコンテンツを照合することが可能となる。

> **注記** 例えば、異なる解像度やエンコーディング形式のアセットレンディションなど。

さらに、アセットからC2PAマニフェストが削除された場合でも、そのマニフェストのコピーが別の場所のプロバンスストアに残っている場合、利用可能なソフトバインディングを用いてマニフェストとアセットを照合することが可能です。

異なる目的を果たすため、ソフトバインディングをハードバインディングとして使用してはならない。

9.3.2. 許可されるソフトバインディングアルゴリズムの一覧

すべてのソフトバインディングは、本仕様でサポートされるソフトバインディングアルゴリズムリストに記載されているアルゴリズムのいずれかを使用して生成されなければならない。

第10章 クレーム

10.1. 概要

クレームは、特定の時点における資産に関するすべてのアサーション（コンテンツへのバインディング用アサーションのセットを含む）を集約する。その後、クレームは第10.3.2.4節「クレームの署名」に記載される方法に従い、暗号的にハッシュ化され署名される。クレームは、ラベル (`c2pa.claim.v2`) の割り当てを含む、アサーションと全く同じ特性を有しますが、アサーションメタデータの使用はサポートしません。クレームは CBOR データとしてエンコードされ、そのように、CBOR のコア決定的エンコーディング要件（RFC 8949、4.2.1 項参照）に準拠しなければなりません。

> **注記**
>
> 以前のバージョンでは、クレームにアサーションメタデータを使用することがサポートされていましたが、これは非推奨となりました。

本仕様の以前のバージョンでは、Claimに対してラベル`c2pa.c`laimおよび関連する`claim-map`を使用していましたが、これらは非推奨となりました。バリデータは引き続きこのラベル（および関連する`claim-map`）を受け入れるべきですが、クレーム生成器はこのようなクレームを生成してはなりません。

10.2. 構文

10.2.1. スキーマ

このタイプのスキーマは、ラベル `c2pa.claim.v2` および `c2pa.claim` を持つクレームに対して、以下の CDDL 定義における `claim-map-v2` および `claim-map` ルールによって定義されます：

```
; C2PAにおけるクレームマップのCDDLスキーマclaim-map
= {
   "claim_generator": tstr, ; http://tools.ietf.org/html/rfc7231#section-5.5.3 に準拠した形式の User-Agent 文字列。クレームを生成したクレーム生成器の名前とバージョンを含める
   "claim_generator_info": [1* generator-info-map],
   "signature": jumbf-uri-type, ; このクレームの署名へのJUMBF URI参照"assertions": [1* $hashed-uri-map],
   "dc:format": tstr, ; アセットのメディアタイプ
   "instanceID": tstr .size (1..max-tstr-length), ; アセットの特定バージョンを一意に識別
   ? "dc:title": tstr .size (1..max-tstr-length), ; アセットの名前
   ? "redacted_assertions": [1* jumbf-uri-type], ; 編集対象となる構成要素マニフェストのアサーションへのJUMBF URI参照のリスト
   ? "alg": tstr .size (1..max-tstr-length), ; このクレームに記載された全てのデータハッシュアサーションの計算に使用される暗号ハッシュアルゴリズムを識別する文字列。特に上書きされない限り、C2PAデータハッシュアルゴリズム識別子レジストリから取得される。これは、このクレームに含まれるdata-hashおよびhashed-uri構造体の'alg'フィールドの値を提供する
   ? "alg_soft": tstr .size (1..max-tstr-length), ; 本クレームに記載される全てのソフトバインディングアサーションの計算に使用されるアルゴリズムを識別する文字列。上書きされない限り、C2PAソフトバインディングアルゴリズム識別子レジストリから取得される。
   ? "metadata": $assertion-metadata-map, ; アサーションに関する追加情報
}

; C2PAにおけるクレームマップのCDDLスキーマclaim-map-
v2 = {
```

```
   "instanceID": tstr .size (1..max-tstr-length), ; アセットの特定バージョンを一意に識別する
   "claim_generator_info": $generator-info-map, ; このクレームのクレーム生成元 "signature": jumbf-uri-type, ; このクレームの署名へのJUMBF URI参照 "created_assertions": [1* $hashed-uri-map],
   ? "gathered_assertions": [1* $hashed-uri-map],
   ? "dc:title": tstr .size (1..max-tstr-length), ; アセット名
   ? "redacted_assertions": [1* jumbf-uri-type], ; 編集対象となる成分マニフェストのアサーションへのJUMBF URI参照のリスト
   ? "alg": tstr .size (1..max-tstr-length), ; 本クレームに記載される全データハッシュアサーションの計算に使用される暗号ハッシュアルゴリズムを識別する文字列（C2PAデータハッシュアルゴリズム識別子レジストリから取得）。本クレームに含まれるdata-hashおよびhashed-uri構造体の'alg'フィールドの値を提供する
   ? "alg_soft": tstr .size (1..max-tstr-length), ; 本クレームに記載される全てのソフトバインディングアサーションの計算に使用されるアルゴリズムを識別する文字列。C2PAソフトバインディングアルゴリズム識別子レジストリから取得され、別段の定めがない限り優先される。
   ? "metadata": $assertion-metadata-map, ; （非推奨） アサーションに関する追加情報
}

generator-info-map = {
   "name": tstr.size(1..max_tstr_length), ; クレーム生成器を命名する人間が読める文字列
   ? "version": tstr, ; 製品のバージョンを表す人間が読める文字列
   ? "icon": $hashed-uri-map / $hashed-ext-uri-map, ; アイコンへのハッシュ化URI（埋め込みまたはリモート）
   ? "operating_system": tstr, ; クレーム生成者が動作しているオペレーティングシステムの読みやすい文字列
   * tstr => any
}
```

CBOR診断表記法（RFC 8949、第8条）におけるクレームマップv2構造の例：

```
{
   "alg" : "sha256", "claim_generator_info"
   : {
      "name": "Joe's Photo Editor",
      "version": "2.0",
      "operating_system": "Windows 10"
   },
   "signature" :
   "self#jumbf=c2pa.signature","created_assertions" : [
      {
         "url": "self#jumbf=c2pa.assertions/c2pa.hash.data", "hash":
         b64'U9Gyz05tmpftkoEYP6XYNsMnUbnS/KcktAg2vv7n1n8='
      },
      {
         "url": "self#jumbf=c2pa.assertions/c2pa.thumbnail.claim", "hash":
         b64'G5hfJwYeWTlflxOhmfCO9xDAK52aKQ+YbKNhRZeq92c='
      },
      {
         "url": "self#jumbf=c2pa.assertions/c2pa.ingredient.v3", "hash":
         b64'Yzag4o5jO4xPyfANVtw7ETlbFSWZNfeM78qbSi8Abkk='
      }
   ],
   "redacted_assertions" : [ "self#jumbf=/c2pa/urn:c2pa:5E7B01FC-4932-
      4BAB-AB32-
D4F12A8AA322/c2pa.assertions/c2pa.metadata"
   ]
}
```

10.2.2. フィールド

存在する場合、dc:titleの値はアセットの人間が読める名前であること。

注記 c2pa.claim には dc:format フィールドが存在しますが、c2pa.claim.v2 ではこのフィールドは廃止されました。

アセットにXMPが含まれる場合、アセットのxmpMM:InstanceIDをinstanceIDとして使用すべきである。XMPが利用できない場合、アセットの他の一意の識別子をinstanceIDの値として使用しなければならない。

注記 dc:titleなどの一部のフィールド名は、XMP標準から直接定義された名前空間プレフィックスを名前として持っています。ただし、C2PAでの使用にはXMPの使用は必須ではありません。

署名フィールドは必須であり、クレーム署名へのURI参照を含めるものとする。

created_assertionsフィールドは存在し、このクレームによって行われるアサーションへの1つ以上のURI参照を含まなければならない。標準マニフェストでは、少なくともハードバインディングを表すアサーションへの参照と、アクションアサーションへの参照を含まなければならない。

注記 すべての created_assertions は、信頼モデルが署名者の信頼に根ざしているため、署名者に帰属する。

存在する場合、gathered_assertions フィールドは、ワークフロー内の他のコンポーネントからクレーム生成器に提供されたアサーションへの 1 つ以上の URI 参照を含めるものとする。

注記 このリストにアサーションを追加することにより、クレーム生成者はそのアサーションがクレームの一部であるが、クレーム生成器から提供されたものではなく、署名者に帰属しないことを宣言します。例えば、人間アクターが入力した情報を含むアサーションは gathered_assertions にリストされます。

存在する場合、redacted_assertions フィールドは、編集済みアサーションへの 1 つ以上の URI 参照を含むものとします。

10.2.3. クレーム生成元情報

10.2.3.1. 一般

クレーム生成器に関する詳細情報は、claim_generator_info の値として存在しなければならない。マニフェストコンシューマーは、自身のため、またはユーザーエクスペリエンス（UX）での表示のために、クレーム生成器に関する情報を決定する際に、claim_generator_info の値を使用しなければならない。

注記 c2pa.claim には claim_generator フィールドがあり、その値は単純な文字列であったが、c2pa.claim.v2 では存在しなくなった。

10.2.3.2. 生成元情報マップ

claim_generator_infoフィールドを追加する場合、その値はgenerator-info-mapオブジェクトであり、nameフィールドを含めるものとする

。また、versionフィールドまたはiconフィールドのいずれか、あるいは両方のいずれかを含む場合がある。さらに、その他の任意のフィールドを含む場合がある。

`name`フィールドを含むものとする。また、versionフィールドまたはiconフィールド、あるいはその両方を追加で含むことができる。さらに、セクション6.2.1「名前空間」で規定される標準的なエンティティ固有の名前空間を使用し、その他の任意のフィールドを追加することも

セクション6.2.1「名前空間」で説明される標準的なエンティティ固有の名前空間を使用して許可されます。このオブジェクトのデータは、クレームを実際に生成した非人間（ハードウェアまたはソフトウェア）のアクター（別名クレーム生成器自体）を表すものとします。

クレーム生成者は、ユーザー体験を提供するマニフェストコンシューマーに対して、自身をグラフィカルに表現する手段（ここではアイコンと呼ぶ）を提供したい場合がある。アイコンフィールドが存在する場合、その値はハッシュ化されたURIでなければならない。このハッシュ化されたURIは、ラベルが`c2pa.icon`であり、複数インスタンスに関するアサーションラベルの規則に従う埋め込みデータアサーションを指すものとする。マニフェストコンシューマーは、本仕様の以前のバージョンで推奨されていたデータボックス方式もサポートすべきである。

注記 アサーション配列と同様に、ハッシュURIに使用されるハッシュアルゴリズムは、ハッシュURI内のalgフィールドによって決定される。algフィールドが存在しない場合は、クレーム内の`alg`フィールドによって決定される。

クレーム生成情報を使用した例

```
{
      "claim_generator_info" : { "name": "Joe's
            Photo Editor", "version": "2.0",
            "operating_system": "Windows 10", "icon": {
                  "url": "http://cdn.examplephotoagency.com/logo.svg", "hash":
                  "5bdec8169b4e4484b79aba44cee5c6bd"
            }
      }
}
```

10.3. クレームの作成

10.3.1. アサーションの作成

クレームを確定する前に、すべてのアサーションを作成し、後述する新規作成のC2PAアサーションストアに保存する必要があります。

標準マニフェストを作成する場合、クレーム作成時に必要なバインディング情報をすべて把握できない可能性があります。その場合は、複数ステップ処理方式を使用して設定を行い、後で情報を入力してください。

10.3.2. クレームの準備

10.3.2.1. アサーションと編集の追加

クレームには `created_assertions` フィールドを含み、`gather_assertions` フィールドを含む場合があります。これら2つのフィールドの値を組み合わせたものは、このクレームによって「主張」されている、アサーションストアに追加されたすべてのアサーションのURI参照のリストを表します。標準マニフェストでは、`created_assertions` フィールドの値には、ハードバインディングを表す少なくとも1つのアサーションが含まれている必要があります。

成分表示における主張が削除される場合、そのURI参照は削除された主張フィールドの値であるリストに追加されるものとする。

10.3.2.2. 成分の追加

多くの制作シナリオにおいて、アクターは完全に新規のアセットを作成するのではなく、既存のアセットを組み込んで作品を作成します。具体的には派生アセット、合成アセット、またはアセットレンディションとして活用します。これらの既存アセットは「材料」と呼ばれ、その使用状況は「材料アサーション」を通じてプロバンスデータに記録されます。

ある構成要素が1つ以上のC2PAマニフェストを含む場合、それらのマニフェストは本資産のC2PAマニフェストストアに挿入され、出所データが保持されることを保証する。このような構成要素マニフェストは、セクション11.1.4「C2PAボックスの詳細」に記載される通りJUMBFに追加される。C2PAマニフェストストアに同一の固有識別子を持つマニフェストが既に存在する場合、両者は（ハッシュ処理により）比較される。同一である場合、新規マニフェストは無視される。異なる場合、新規マニフェストは第8章「*固有識別子*」に記載の方法で固有識別子を新規値に変更した後、ストアに追加される。

成分のマニフェストがリモートであり、クレーム生成者がマニフェストを取得できない場合、その状態を反映するためにエラーコード `manifest.inaccessible` を使用すべきである。

10.3.2.3. 署名の接続

署名は署名済みペイロードの一部にはなりえないが、そのラベルは事前定義されているため、完全なURI参照も既知である。したがって、クレームの署名フィールドの値をそのURI参照に設定することで、それをクレームに含めることができる。

> **注記** これにより、クレームとその署名が明示的にバインドされます。

10.3.2.4. クレームの署名

署名の生成は、セクション13.2「デジタル署名」で規定されている。

標準マニフェストと更新マニフェストの両タイプにおいて、`Sig_structure`のペイロードフィールドはクレーム文書のシリアライズされたCBORとし、分離コンテンツモードを使用するものとする。

デジタル署名手順の結果としてシリアル化された `COSE_Sign1_Tagged` 構造体は、C2PA クレーム署名ボックスに書き込まれます。

10.3.2.5. タイムスタンプ

10.3.2.5.1. RFC 3161 の使用

可能であれば、クレーム生成者はRFC 3161準拠のタイムスタンプ機関（TSA）（RFC 3161）を使用して、署名自体が特定の日時に実際に存在したことを証明する信頼できるタイムスタンプを取得し、それをCOSE_Sign1_Tagged構造体に副署として組み込むべきである。

クレーム生成者は、自身のマニフェストが有効性を維持することを保証するため、タイムスタンプを取得し含めることが推奨される。

第15章「*検証*」で説明されているように、タイムスタンプのないマニフェストは、署名資格情報が失効または取り消された時点で無効となる。マニフェストには1つのタイムスタンプのみを含めること。

注記	本仕様の以前のバージョンでは、マニフェストに複数のタイムスタンプを含めることが許可されていた。

10.3.2.5.2. ペイロードの選択

本仕様の以前のバージョンでは、タイムスタンプのペイロードフィールドに、セクション10.3.2.4「クレームの署名」で説明されているSig_signatureで使用されたのと同じ値を使用していました。このペイロードは今後「v1タイムスタンプ」内の「v1ペイロード」と呼ばれ、非推奨と見なされます。クレーム生成者はこれを生成してはならないが、検証者は存在する場合に処理しなければならない。

「v2タイムスタンプ」の「v2ペイロード」は、セクション10.3.2.4「クレームの署名」の一部として作成されるCOSE_Sign1_Tagged構造体の署名フィールドの値である。タイムスタンプ操作を実行するクレーム生成者は「v2ペイロード」を使用しなければならない。

注	署名フィールドの値には、主要なタイプと長さを示すバイト（文字列自体だけでなく）を含む、シリアル化された `bstr` 全体が含まれます。

10.3.2.5.3. タイムスタンプの取得

すべてのタイムスタンプは、RFC 3161 に記載された方法に従い、以下の追加要件を満たす形で取得されなければならない：

- TimeStampReq構造体（RFC 3161、セクション2.4.1）のMessageImprintは、RFC 8152、セクション4.4のToBeSigned値を、`Sig_structure`の要素に対して以下の値を用いて作成することにより計算されるものとする：
  - `context`要素は`CounterSignature`とする。
  - payload要素は、セクション10.3.2.5.2「ペイロードの選択」で記述される値とする。
  - `Sig_structure`の残りの要素は、セクション13.2.3「署名の計算」に記載されている通りとする。
- 次に、`ToBeSigned`の値は、TSAがサポートするセクション13.1「ハッシュ処理」の許可リストに記載されたハッシュアルゴリズムを用いてハッシュ化され、そのハッシュ値と元の値が`MessageImprint`に格納される。TSAが許可リストのハッシュアルゴリズムを一切サポートしない場合、タイムスタンプ処理に使用することはできない。
  - 可能な限り、ハッシュアルゴリズムはクレームのデジタル署名で使用されたものと同じアルゴリズムを使用すべきである。
- タイムスタンプ要求構造体の `certReq` ブール値は、TSA への要求においてアサートされ、応答に証明書チェーンが提供されることを保証する。

10.3.2.5.4. タイムスタンプの保存

v1タイムスタンプ（非推奨）は、ラベルが文字列`sigTst`であるCOSE非保護ヘッダーに格納される。存在する場合、このヘッダーの値は例2「`tstContainer`用CDDL」で定義される`tstContainer`でなければならない。TSAからの応答で受信したTimeStampResp構造体の内容は、`tstTokens`要素の`val`プロパティの値として格納される。

v2タイムスタンプは、ラベルが文字列sigTst2であるCOSE非保護ヘッダーに格納されるものとする。存在する場合、このヘッダーの値は例2「tstContainer用CDDL」で定義されるtstContainerであるものとする。TSAからの応答で受信したTimeStampResp構造体のtimeStampTokenフィールドの値は、tstTokens要素のvalプロパティの値として格納される。これは、CBORバイト文字列でラップされたDERエンコードされたRFC 3161 TimeStampTokenとしてフォーマットされる。

注記

v2タイムスタンプは、RFC 3161タイムスタンプに対するCOSEヘッダーパラメータの「CTT」モデルに相当する。スタンプトークン草案におけるCOSEヘッダーパラメータの「CTT」モデルに相当する。タイムスタンプ付与前に署名構造全体が完成していることを要求するため、タイムスタンプは実際の証明書を含む署名構造全体に対する副署として機能する。

タイムスタンプが含まれていない場合、COSE非保護ヘッダーにはいずれのヘッダー（sigTstもsigTst2も）も存在してはならない。

例2. tstContainer のCDDL

```
; JSONスキーマに基づくtstContainerおよび関連構造体のCBORバージョン
; https://forge.etsi.org/rep/esi/x19_182_JAdES/raw/v1.1.1/19182-jsonSchema.json tstContainer = {
   "tstTokens": [1* tstToken]
}

tstToken = { "val":
   bstr
}
```

注記

上記の定義は、JAdESのセクション5.3.4およびそのJSONスキーマからのスキーマのサブセットをCBORに適応させたものであり、ただしvalの内容がBase64エンコードされた文字列ではなくバイト列である点が変更されています。

10.3.2.6. 認証情報の失効情報

署名者の認証情報がオンライン認証ステータスの照会をサポートしており、かつ認証情報にタイムスタンプ付き認証ステータス情報を提供するサービスへのポインタが含まれている場合、クレーム生成者は当該サービスを照会し、応答を取得し、信頼モデルで認証情報について記述されている方法で保存すべきである。この方法で認証情報の失効情報が添付されている場合、署名後にセクション10.3.2.5「タイムスタンプ」で説明されている信頼できるタイムスタンプも取得しなければならない。

10.3.3. 請求の例

10.3.3.1. 単一クレーム

以下は、単一のクレーム内に複数のアサーションが埋め込まれた画像の視覚的表現です。

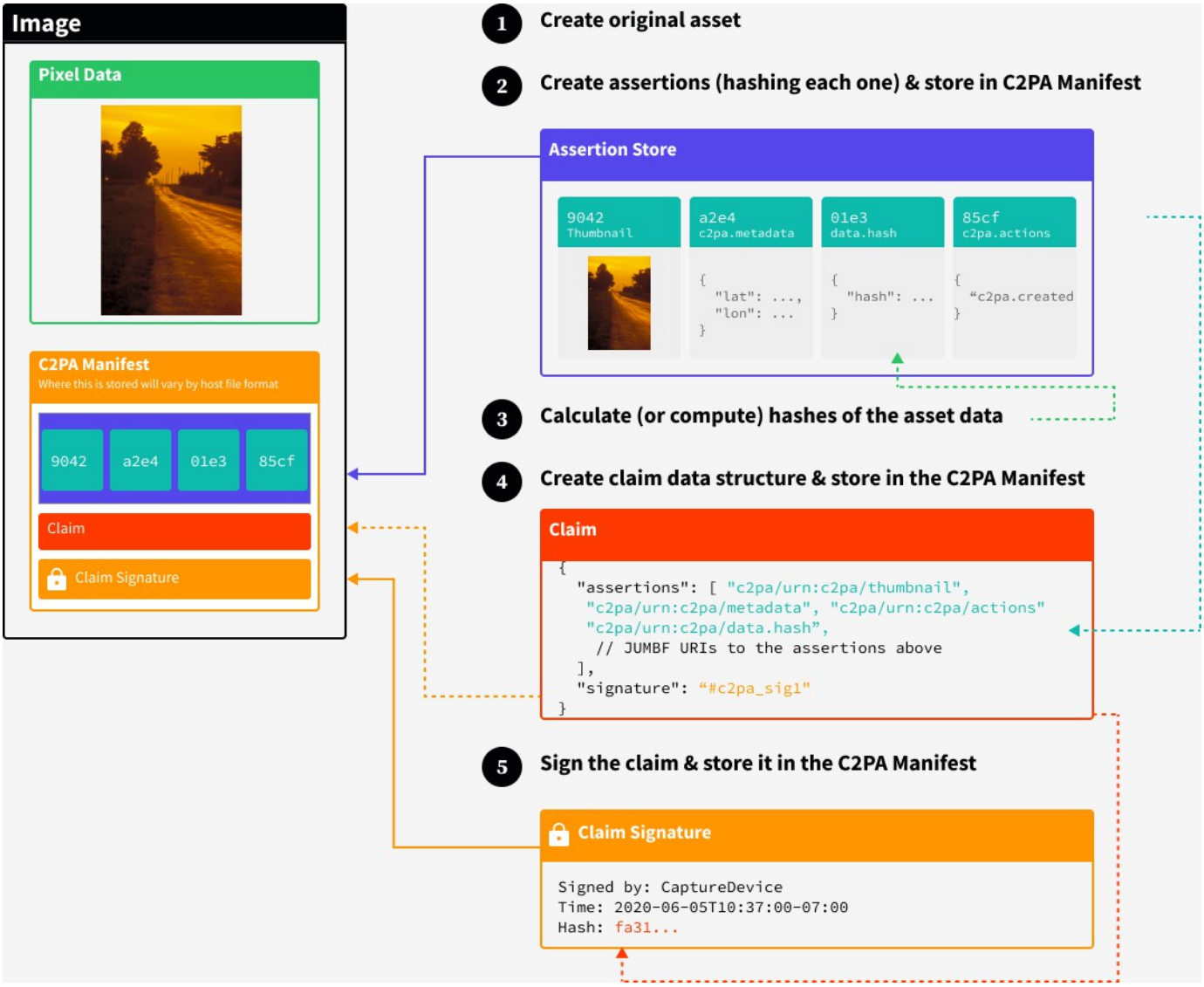

図5. 複数の主張を含む単一クレーム

10.3.3.2. 複数のクレーム

前の例に続く2つ目のクレームを作成するこの例では、元のクレームから1つのアサーションが削除されています。このシナリオの視覚的表現は次のようになります：

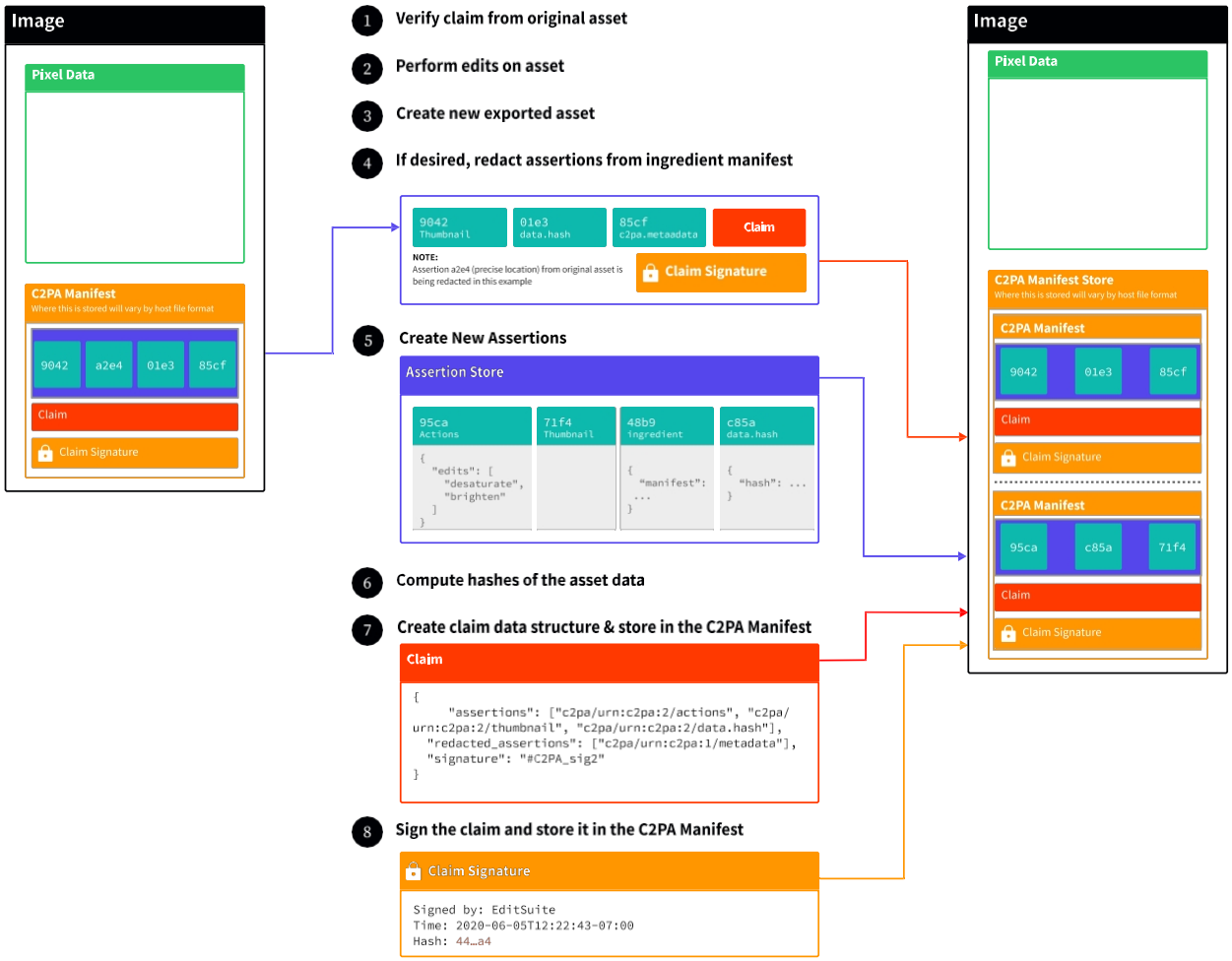

図6. 二次クレームにおける主張の削除

10.4. 複数ステップ処理

一部のアセットファイル形式では、マニフェスト署名前にC2PAマニフェストストアとアセットコンテンツのファイルオフセットを固定する必要があります。これにより、コンテンツバインディングが認証対象のコンテンツと正しく位置合わせされます。残念ながら、マニフェストとその署名のサイズは署名完了まで正確に把握できず、ファイルオフセットが変動する可能性があります。

例として、JPEG 1ファイルでは、画像データより前にC2PAマニフェストストア全体がファイル内に存在する必要があるため、そのサイズが認証対象コンテンツのファイルオフセットに影響を与えます。

これを達成するには、PDFでの署名の方法と同様に、複数のステップからなるアプローチを採用する。

10.4.1. コンテンツバインディングの作成

標準マニフェストを作成する際、そのクレームは、資産が改ざん防止であることを保証するために、アサーションのリストに１つ以上のコンテンツバインディングアサーションを含めるものとします。

データハッシュアサーションを作成し、以下の考慮事項を踏まえてアサーションストアに追加する。

多くの場合、JPEG 1のように、マニフェストがファイルの中間に埋め込まれるため、アセット全体をハッシュ化することは不可能です。つまり、アセットハッシュを計算する時点では、マニフェストデータのサイズや位置が不明となるためです。この循環依存関係は、ハッシュ化時に除外範囲を指定できるようにすることで回避されます。除外範囲が指定された場合、単一のハッシュ処理は実行されますが、対象となるのは除外範囲に含まれないアセット領域のみとなります。

JPEG 1ファイルの中央にAPP11セグメントとしてマニフェストが埋め込まれている場合、クレーム作成者はハッシュ計算からAPP11セグメントを除外できます。

挿入攻撃を防ぐため、可能な限り単一の除外範囲のみを設定することが望ましい。アセット内のマニフェストのサイズまたは位置（あるいはその両方）が不明な場合、データハッシュアサーションの開始位置と長さの値は両方ともゼロとし、パディング値のサイズは第2パスで値を書き込むのに十分な大きさとする。少なくとも16バイトが推奨される。パディングキーの値は全て0x00で構成されるものとする。

パディングが使用される場合、検証失敗を招かずにパディングデータが変更される可能性がある。クレーム生成者は、パディングデータ（またはその他の除外されたアセットデータ）の変更がアセットの解釈方法に影響を与えないことを保証しなければならない。

注記

JPEG 1ファイルの場合、これを達成するには、パディングを排除するか、または `JFIF` APP11/C2PAセグメントが短縮または異なるセグメントタイプに変更されないことを保証することで達成できる。これはすべてのC2PAマニフェストセグメントヘッダー（APP11）と2バイトの長さフィールドを、マニフェストを含む全セグメントのデータハッシュマップに含めることで実現される。これにより、除外領域で変更されたデータがJPEGプロセッサによって誤解釈されることを防ぐ。

10.4.2. 一時的なクレームと署名の作成

新たに作成したデータハッシュアサーション参照を、一時的なハッシュ値（空白など）を提供するクレームのアサーションリストに追加します。

この時点で一時クレームは完成し、作成中のC2PAマニフェストに追加可能となる。

現時点ではクレームが一時的なものであるため、署名することはできません。クレーム署名ボックスに有効なCBOR構造が含まれるようにするには、RFC 8152のセクション4.2に記載されているように、一時的なCOSE_Sign1_Tagged構造を作成します。`COSE_Sign1_Tagged`は、タグバイトに続いてCOSE_Sign1構造（4要素のCBOR配列）が続くものです。配列は以下のように構築します：

- 最初の要素は保護されたヘッダーバケット（RFC 8152、セクション3）です。この位置にサイズ0のbstrを配置して空のバケットを作成します。`bstr` of size 0 を配置して空のバケットを作成します。
- 第二の要素は保護されていないヘッダーバケットであり、CBORマップです。1組のマップを作成します。ラベルとして文字列パディングを使用し、値としてゼロバイト（0x00）で埋められた希望のパディングサイズのbstrを配置します。このパディングの初期サイズには25キロバイトが推奨されます。
- 第三の要素はペイロードです。ここに`nil`（CBORメジャータップ7、値22）を配置します。

- 第四の要素は署名です。ここにサイズ0のbstrを配置してください。

10.4.3. C2PAマニフェストの完成

この時点で、アセットのC2PAマニフェスト全体を構成するすべてのボックスが完成し、最終形態に構築可能となります（未構築の場合）。アセットのC2PAマニフェストは、構成要素のマニフェストと共に結合され、完全なC2PAマニフェストストアを形成します。アクティブなマニフェストは、C2PAマニフェストストアのスーパーボックス内で最後のC2PAマニフェストスーパーボックスである必要があります。C2PAマニフェストストアは、セクション11.3「各種ファイル形式へのマニフェスト埋め込み」で説明されているように、アセットに埋め込むことができます。

10.4.4. 戻って記入する

C2PAマニフェストストアがアセットに埋め込まれたため、アクティブなマニフェストの開始オフセットと長さは、そのデータハッシュアサーション内で更新可能となりました。この際、アサーションのボックスサイズを変更せず、データのみを変更する必要があります。これは、残りのバイトを「埋める」ために必要な長さにpadフィールドの値を調整することで実現されます。

注記

推奨される/決定論的なCBORシリアライゼーションにおけるパディングは、可変長整数を用いてエンコード済みバイナリデータの長さを指定します。長さが0から1バイト、または1から2バイト（以下同様）に変化する場合、結果のパディング長は2バイトずつ飛躍します。これは、全てのパディングが
単一のパディングフィールド。例えば、24バイトおよび26バイトのパディングは作成可能だが、25バイトのパディングは作成できない。この状況が発生した場合、必要なパディングはpadとpad2に分割できます。例えば、25バイトのパディングを作成するには、クレーム生成器が19バイトをpadにエンコード（結果としてエンコード長は20バイト）、4バイトをpad2にエンコード（結果として5バイト）します。

データハッシュアサーションの更新後、ハッシュを生成し、以前位置情報を保持していた空き領域に書き込む。

これでクレームは完成し、セクション10.3.2.4「クレームの署名」に記載の方法でハッシュ化および署名が可能となる。生成された署名は事前割り当て領域に格納される。その後、クレーム署名ボックスのサイズを維持するため、必要に応じてパディングヘッダーを縮小できる。このヘッダーは保護されていないため、変更してもクレーム署名の有効性は失われない。

シリアライズされたCOSE_Sign1_Tagged構造体がC2PAクレーム署名ボックスの予約サイズを超える場合、セクション10.4.2「一時的なクレームと署名の作成」で選択されたより大きなパディングサイズを用いて、複数ステップ処理を繰り返すものとする。前回の試行で取得した失効情報は、その有効期間内（RFC 6960、セクション 4 .2.2.1）であれば再利用可能であるが、追加されたパディングの結果としてファイルオフセットが変更された新しいクレームには、新しいタイムスタンプが必要となる。

C2PAマニフェストには、本仕様で定義されていないアサーションが含まれる可能性があり、それらはファイルレイアウトに依存する場合がある。そのため、クレーム生成器はデータハッシュアサーションにおいてファイルレイアウトやオフセットを変更できなくなる可能性がある。この場合、クレーム生成器はアサーション作成前にパディングを使用し、アサーションが確定した後にファイルレイアウトを変更する必要がないようにすべきである。

第11章 マニフェスト

11.1. JUMBFの使用

11.1.1. 理由

C2PAの多くの要件をサポートするため、C2PAマニフェストは構造化されたバイナリデータストアに保存（シリアライズ）される必要がありました。これにより、以下の特定の機能が可能となります：

- 単一のコンテナ内に複数のマニフェスト（例：親製品と原材料）を保存する機能。
- URIを介して個々の要素（マニフェスト内およびマニフェスト間）を参照する機能。
- 要素のハッシュ化対象部分を明確に識別する機能。
- C2PAで使用される事前定義データ型（例：JSON、CBOR）を保存する機能。
- 任意のデータ形式（例：XML、JPEGなど）を保存する機能。

上記の要件をすべてサポートするだけでなく、当社が選択したコンテナ形式であるISO 19566-5:2023（JUMBF）は、JPEGファミリーのフォーマットによってネイティブにサポートされており、多くの一般的な画像および動画ファイル形式で使用されているボックスベースモデル（ISOBMFF、ISO 14496-12など）とも互換性があります。JUMBFを使用することで、JPEG/JFIFやPNGといった従来の画像フォーマット、3Dフォーマット、ドキュメントフォーマット（例：PDF）などとも連携可能でありながら、同じ利点（URI参照などの追加機能も含む）をすべて享受できます。このシリアル化されたフォーマットは、JUMBF をネイティブでサポートしていないフォーマットでも、また C2PA マニフェストストアが別ファイルや URI 場所など、アセットとは別に保存される場合にも使用されます。

注記

標準的なアサーションの大半およびクレーム署名もCBORでシリアライズされるため、C2PAマニフェスト全体にCBORを使用することも検討されましたが、CBORがコンテナ形式ではないため採用されませんでした。

例えば、CBOR内部に「JSONの塊」を格納し、それがJSON（他の形式ではない）であることを認識するには、そのようなものを格納するためのデータ構造を設計する必要がある。次に、その構造をどのように運ぶかについて、親構造を定義しなければならない。この同じ概念は、JUMBFの各ネイティブ機能に対しても同様に適用される必要がある。

必要な機能をすべてCBORで完全に再実装することは確かに可能ですが、膨大な作業量となり、すべての実装においてJUMBF/BMFFパーサーの必要性を完全に排除することはできません。

11.1.2. 処理規則

C2PAマニフェストコンシューマーは、C2PAマニフェストストア内に含まれていないアサーション、アサーションストア、クレーム、クレーム署名、またはC2PAマニフェストを決して処理してはならない。さらに、C2PAマニフェストコンシューマーが認識できないJUMBFタイプUUIDを持つJUMBFボックスまたはスーパーボックスに遭遇した場合、その内容をスキップ（および無視）しなければならない。

注記 これは、C2PAマニフェストコンシューマーが認識しているプライベートボックスは処理できるが、認識していないものは無視することを意味する。

いずれかのJUMBFボックスまたはスーパーボックスのJUMBF説明ボックスにおいて、[*リクエスト可能*]および[*ラベル存在*]の両トグルが設定されている場合、当該ボックスまたはスーパーボックスは更新されたC2PAマニフェストストアに維持されるものとする。

注記 これらのトグルが設定されたボックスは、JUMBF URI 経由で参照されることを意図しており、それらを削除すると下流のワークフローが失敗する可能性があります。

11.1.3. 拡張機能

11.1.3.1. 一般

このセクションでは、本仕様で要求される JUMBF 仕様（ISO 19566-5:2023）の拡張について記述します。

11.1.3.2. 圧縮ボックス

マニフェストの圧縮をサポートするため、C2PAでは新しいブロブコンテンツボックスがサポートされる。JPEG-XL（ISO/IEC 18181-2:2024）の類似ボックスに基づき、ブロブボックスは圧縮マニフェスト条項で規定される標準マニフェストまたは更新マニフェストのいずれかのBrotli圧縮バイトを内容とするコンテンツボックスである。brobボックスのボックスIDは`0x62726F62`（`brob`）とする。

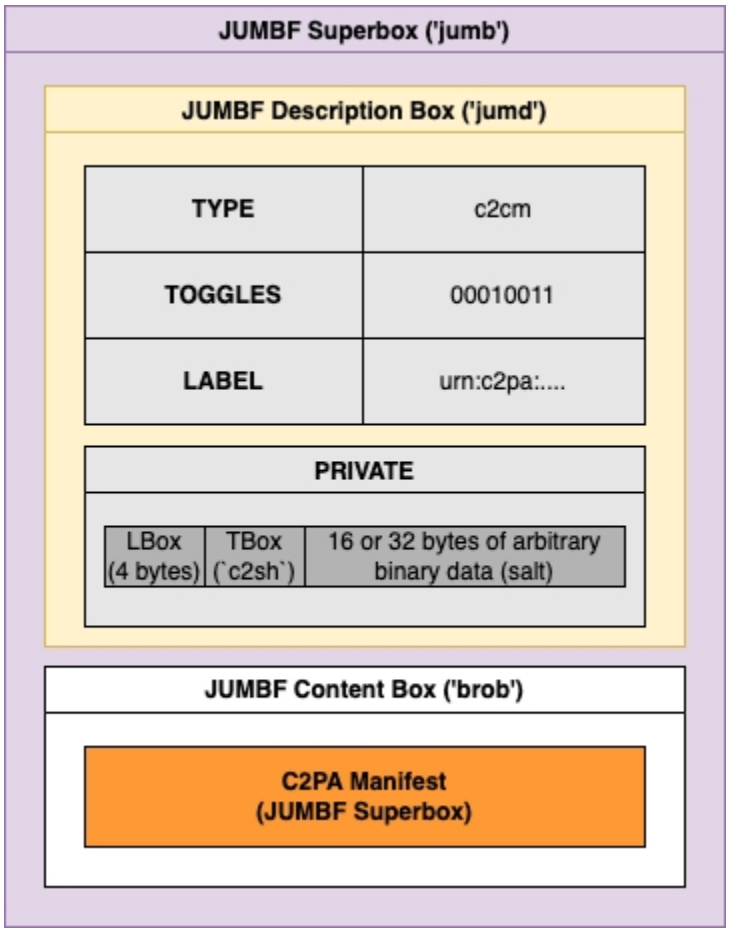

図7. 圧縮マニフェストの例

圧縮されたボックスのハッシュ処理は、セクション8.4.2.3「JUMBFボックスのハッシュ処理」で説明されている他のボックスと同様の方法で行われます。

注記

これは、activeManifestフィールドを介してC2PAマニフェストへの参照を含むingredientアサーションから取得したhashed_uriに対して、ハッシュが他のJUMBFスーパーボックスと同様のプロセスで計算されることを意味します。具体的には、JUMBF記述ボックスと圧縮ペイロードを含むブロブボックスを対象とし、スーパーボックスのヘッダーは除外します。ブロブボックスの内容はハッシュ計算前に解凍されません。

11.1.4. C2PAボックスの詳細

11.1.4.1. JUMBF 記述ボックス

11.1.4.1.1. ラベル

JUMBF仕様（ISO 19566-5:2023、A.3）に記載されている通り、ラベルはUTF-8エンコーディングによるISO/IEC 10646文字として格納されるものとする。ラベルには、U+0000 から U+001F までの範囲、U+007F から U+009F までの範囲の文字、および特定の文字 '/'、';'、'?'、'#' は使用できません。ラベルはヌル終端でなければなりません。

JUMBF URIの一部として使用されるラベルとして、U+FEFF、U+FFFF、およびU+D800-U+DFFFの文字も使用してはならない。

11.1.4.1.2. トグル

C2PAマニフェストで使用される全てのJUMBF説明ボックス（ISO 19566-5:2023、A.3）にはラベルが必要であり、*ラベル存在*トグル（`xxxxxx1x`）を設定しなければならない。さらに、JUMBF URI はシステム全体でボックスを参照するために使用されるため（例：アサーションのリスト、成分への参照など）、*要求可能*トグル（`xxxxxx11`）を設定しなければならない。

セクション8.4.2.3「JUMBFボックスのハッシュ化」に記載されているように、*PRIVATE*ボックスにソルトを含める場合、*Private*トグル（`xxx1xxxx`）も設定しなければならない。

11.1.4.2. マニフェストストア

C2PAデータはJUMBF互換のボックス構造にシリアライズされます。最外層のボックスはC2PAマニフェストストア（別名：コンテンツクレデンシャル）と呼ばれます。図8「C2PAマニフェストストア」は単一のC2PAマニフェストを含むC2PAマニフェストストアの例です：

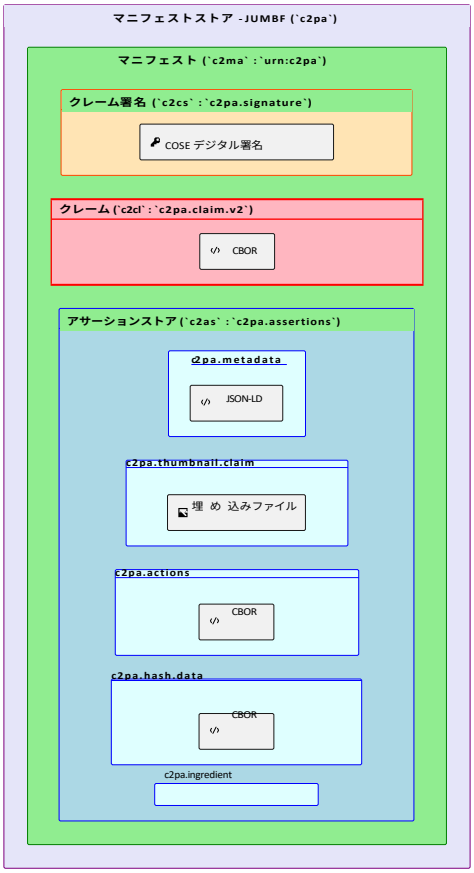

図8. C2PA マニフェストストア

C2PAマニフェストストアは、一連の他のJUMBFボックスおよびスーパーボックスで構成されるJUMBFスーパーボックスであり、各ボックスは自身のJUMBFタイプUUIDおよびJUMBF説明ボックス内のラベルによって識別される。C2PAマニフェストストアは、ラベル`c2pa`、JUMBFタイプUUID `63327061-0011-0010-8000-00AA00389B71`（`c2pa`）を持ち、1つ以上のC2PAマニフェストスーパーボックス（別名C2PAマニフェスト）を含むものとします。C2PA マニフェストストアは、本仕様で定義されていない JUMBF タイプ UUID を持つ JUMBF ボックスおよびスーパーボックスも含む場合がある。

注記

他のボックスおよびスーパーボックスを許可することで、C2PAへのカスタム拡張が可能となり、また将来の仕様バージョンにおいて互換性を損なうことなく新たなボックスを追加できるようになります。

各C2PAマニフェストには、クレーム発行時に生成されたデータ（C2PAアサーションストア、C2PAクレーム、C2PAクレーム署名を含む）を含めるものとする。C2PAマニフェストには、本仕様で定義されていないJUMBFタイプUUIDを持つJUMBFボックスおよびスーパーボックスを含めることもできる。

各C2PAマニフェストのJUMBFタイプUUIDは、`63326D61-0011-0010-8000-00AA00389B71`（`c2ma`）、`6332756D-0011-0010-8000-00AA00389B71`（`c2um`）のいずれかとする。マニフェストの種類に応じて。C2PAマニフェストボックスには、一意の識別子で説明されている方法で計算された`urn:c2pa`値でラベル付けされる。

11.1.4.3. アサーションストア

C2PA アサーションストアは、`c2pa.assertions` のラベルと、JUMBF タイプの UUID

`63326173-0011-0010-8000-00AA00389B71` (`c2as`) を持つ。これには一つ以上の JUMBF スーパーボックス（以下

C2PAアサーションボックス）のJUMBFタイプは、アサーションデータを含むサブボックスのタイプを定義する（ISO 19566-5:2023、附属書B）。これらのスーパーボックスは、それぞれ標準アサーションで定義されるラベルを有し、JUMBF記述ボックス、1つ以上のJUMBF内容ボックス、および場合によってはパディングボックスを含むものとする（ISO 19566-5:2023、A.4）。

JUMBFコンテンツタイプ（ISO 19566-5:2023、附属書B）の各アサーションスーパーボックスに含まれるボックスは、CBORコンテンツタイプ（`cbor`）、JSONコンテンツタイプ（`json`）、埋め込みファイルコンテンツタイプ（`bfdb` & `bidb`）、またはUUIDコンテンツタイプ（`uuid`）であるべきであるが、JUMBF（ISO 19566-5:2023）およびその改正で定義される任意のコンテンツタイプが許可される。さらに、ISO 19566-4:2020 に記載される JUMBF 保護ボックスも使用できる。

注記	暗号化データなど、他の形式/シリアライゼーションのデータを含むカスタムアサーションは カスタムUUIDに続いてデータが続くUUIDコンテンツボックスの使用を通じてサポートされる（ISO 19566-5:2023、B.5）。

11.1.4.4. クレームおよびクレーム署名

C2PAクレームボックスは、`c2pa.claim.v2`というラベル、JUMBFタイプのUUID `6332636C-0011-0010-8000-00AA00389B71`（`c2cl`）を持ち、単一のCBORコンテンツタイプボックス（`cbor`）で構成される。

C2PA クレーム署名ボックスは、`c2pa.signature` というラベル、`63326373-0011-0010-8000-00AA00389B71` (`c2cs`) という JUMBF タイプの UUID を持つものとし、単一の CBOR コンテンツタイプボックス (`cbor`) で構成されるものとする。

11.1.4.5. 成分の保存

C2PAマニフェストに成分アサーションが含まれ、かつその成分がC2PAマニフェストを含む場合、当該C2PAマニフェストは、来歴データが保持されるよう含めるものとする。このような成分マニフェストは、当該資産自体のC2PAマニフェストと同等のものとしてC2PAマニフェストストアに追加される。

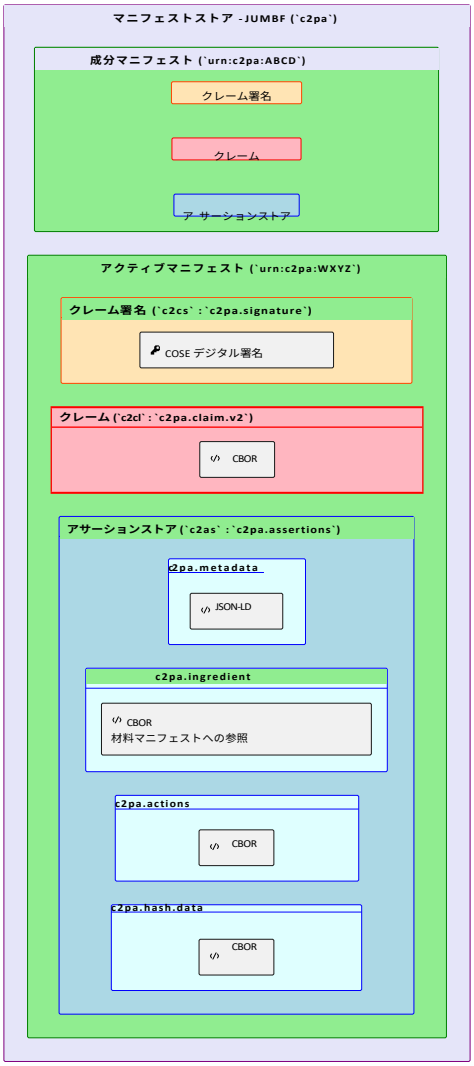

図9. 成分を含むC2PA マニフェストストア

11.1.4.6. データストレージ

重要

このセクションは歴史的記録として残されています。データボックスの概念は廃止されました。

廃止され、標準のJUMBF埋め込みファイルコンテンツタイプボックスを使用してデータを格納する標準アサーションに置き換えられました。埋め込みデータアサーションの詳細については、セクション18.12「埋め込みデータ」を参照してください。

C2PA データボックスストアは、1 つ以上の CBOR コンテンツタイプボックス (`cbor`) のみを含む JUMBF スーパーボックスである。他の種類の JUMBF ボックスやスーパーボックスは含んではならない。ラベルは `c2pa.databoxes`、JUMBF タイプ UUID は `63326462-0011-0010-8000-00AA00389B71` (`c2db`) とする。

CBOR コンテンツタイプボックスは、`c2pa.data`（埋め込みデータ用）のラベルを持つ。

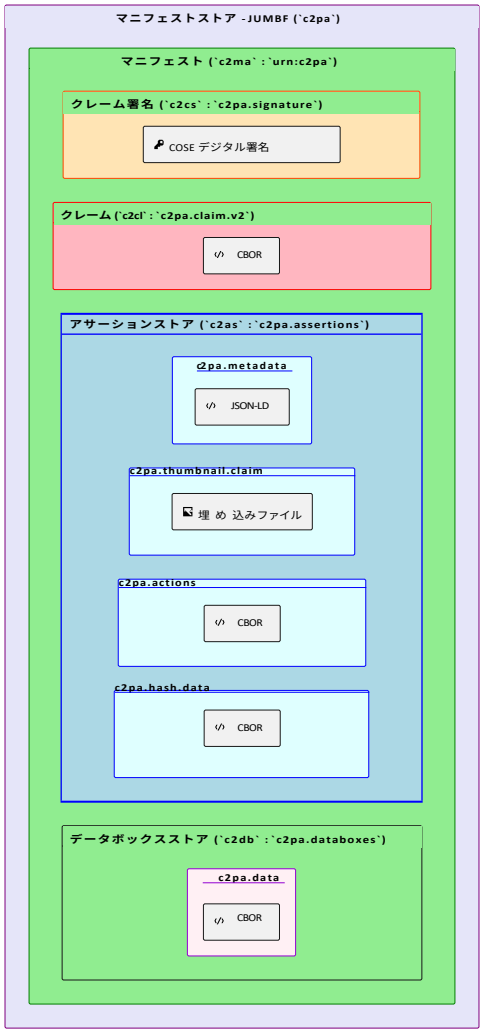

図10. データボックス付きC2PA マニフェストストア

11.2. マニフェストの種類

11.2.1. 共通点

すべてのC2PAマニフェストは、少なくとも1つのアサーション、クレーム、およびクレーム署名を含むアサーションストアを含まなければならない。

11.2.2. 標準マニフェスト

標準 C2PA マニフェスト (JUMBF タイプ UUID: `63326D61-0011-0010-8000-00AA00389B71(c2ma)`) は、マニフェストの対象となるアセットのタイプとバージョンに基づいて、`c2pa.hash.data`、`c2pa.hash.boxes`、`c2pa.hash.collection.data`、`c2pa.hash.bmff.v2` (非推奨)、または `c2pa.hash.bmff.v3` のいずれか、コンテンツアサーションへのハードバインディングを 1 つだけ含めるものとする。この要件により、C2PA 由来データにはこのタイプのマニフェストが主に存在することになります。

マニフェストコンシューマーは、JUMBF タイプ UUID `63326D64-0011-0010-8000-00AA00389B71` (c2md) で指定された標準 C2PA マニフェストも受け入れるものとしますが、クレームジェネレータは、この JUMBF タイプ UUID を持つマニフェストを作成してはなりません。

注記 標準 C2PA マニフェストは、アクティブマニフェストまたは成分マニフェストのいずれかとして配置することができます。

11.2.3. マニフェストの更新

ただし、追加のアサーションが必要だがデジタルコンテンツ自体は変更されないプロバンスワークフローも存在する。これらのワークフローでは、更新マニフェスト（JUMBFタイプUUID: `6332756D-0011-0010-8000-00AA00389B71` (`c2um`)）を使用すべきである。

更新マニフェストには、`c2pa.hash.data`、`c2pa.hash.boxes`、`c2pa.hash.collection.data`、`c2pa.hash.bmff.v2`（非推奨）、`c2pa.hash.bmff.v3` タイプの断言を含めてはならない。コンテンツが変更されていないため、バインディングを更新する必要がないからである。ファイルオフセットハッシュ（`c2pa.hash.data`）の場合、C2PAマニフェストストアは更新後も同じファイルオフセットから開始されなければならず、長さが変更される可能性があるのみである。さらに、`c2pa.hash.multi-asset`アサーションを含んではならない。

更新マニフェストには `c2pa.actions` または `c2pa.actions.v2` タイプのアサーションを含めることができます。ただし、これらのアサーションの `actions` 配列に含まれる各アクションの `action` フィールドの値は、以下のいずれかのみである必要があります：

- `c2pa.edited.metadata`
- `c2pa.opened`
- `c2pa.published`
- `c2pa.redacted`

更新マニフェストは、以下の値を含む`c2pa.`actionsまたは`c2pa.actions.v2`型の断言を含んではならない：
このリスト外のアクションフィールド。

更新マニフェストには、タイムスタンプアサーション、証明書ステータスアサーション、またはその両方が含まれる場合があります。

注 これは廃止予定のタイムスタンプマニフェスト機能に代わるアプローチです。

更新マニフェストは、サムネイルアサーションを含んではならない。

注 これらの要件が設けられている理由は、このリストにないアクションフィールドやサムネイルは、デジタルコンテンツに変更があったことを意味するためです。

更新マニフェストには、更新対象のアセットについて、(a) `activeManifest` および `claimSignature` フィールドの両方を含み、それぞれの値が C2PA マニフェストおよびクレーム署名への URI 参照である（または `c2pa.ingredient.v3` フィールドを含む `c2pa.ingredient.`v3 または `c2pa.ingredient.v3` フィールドを含む `c2pa.ingredient.`v3 フィールドを含む `c2pa.ingredient.v3` フィールドを含む `c2pa.ingredient.`v3 フィールドを含む `c2pa.ingredient.v3` フィールドを含む `c2pa.ingredient.v3` フィールドを含む `c2pa.ingredient.v3` フィールドを含む `c2pa.ingredient.v3` フィールドを含む `c2pa.ingredient.v3` フィールドを含む `c2pa.ingredientv2` または `c2pa_manifest` フィールドを含む `c2pa.ingredient`）への URI 参照であり、かつ (b) relationship フィールドの値が `parentOf` である。

注記 成分の C2PA マニフェストは、標準マニフェストまたは更新マニフェストのいずれかです。

11.2.4. 圧縮マニフェスト

標準マニフェストおよび更新マニフェストは、前述のBrotli圧縮アルゴリズムを用いて全体を圧縮できる。いずれのマニフェストタイプにおいても、TYPEフィールドの値はc2cmとし、labelフィールドの値は圧縮マニフェストスーパーボックスのラベルと同一とし、brobコンテンツボックスの内容はマニフェストスーパーボックス全体の圧縮バイトデータとする。圧縮された標準マニフェストの例については、図7「圧縮マニフェストの例」を参照のこと。

重要 本仕様書において標準マニフェストまたは更新マニフェストが参照される箇所は、圧縮標準マニフェストまたは圧縮更新マニフェストも同様に有効である。

11.2.5. タイムスタンプマニフェスト（歴史的）

重要 この機能はタイムスタンプアサーションに取って代わられ、非推奨となりました。記述しないでください
クレーム生成者によって記述されることも、マニフェスト消費者によって読み取られることもありません。代わりに、タイムスタンプアサーションが同じ目的を達成するために使用されます。

注記 以下の情報は、過去の記録として残されています。

一部のプロバンスワークフローでは、署名時にTSAから信頼できるタイムスタンプ（RFC 3161に準拠）を取得できないオフライン環境で、標準または更新マニフェストが作成される。この状況に対応するため、タイムスタンプマニフェスト（JUMBF タイプ UUID: `6332746D-0011-0010-8000-00AA00389B71 (c2tm)`）を使用し、TSA に連絡が取れる後の操作でタイムスタンプを追加することが可能です。

11.3. 各種ファイル形式へのマニフェスト埋め込み

C2PAマニフェストは、画像、動画、音声、フォント、文書など、様々なメディアタイプを含む多様なファイル形式に埋め込むことができます。付録A「*マニフェストの埋め込み*」では、各サポート対象ファイル形式へのC2PAマニフェストの埋め込み方法に関する技術的詳細を説明しています。

注記 BMPなどの多くの古典的な画像フォーマットは、任意のデータの埋め込みをサポートしていないため、外部マニフェストの使用が必要となります。

11.4. 外部マニフェスト

場合によっては、C2PAマニフェストストアをアセットに埋め込むことが不可能（または非現実的）なことがあります。そのような場合、アセットの外部にC2PAマニフェストを保持することは、アセットの由来情報を提供する上で許容されるモデルです。C2PAマニフェストは、参照やURIなどによって、その資産を扱うマニフェストコンシューマーが容易に発見できる場所（マニフェストリポジトリと呼ばれる）に保存されるべきです。C2PAマニフェストストアはJUMBFボックスであるため、JUMBFメディアタイプであるapplication/c2paで提供されるものとします。

注記 この仕様の以前のバージョンでは、C2PA マニフェスト ストアに対してメディアタイプ `application/x-c2pa-manifest-store` を使用していました。このメディアタイプは非推奨です。

外部マニフェストを使用する一般的な理由は以下の通りです：

- 技術的に不可能な場合（例：.txtファイルの場合）。
- C2PAマニフェストストアのサイズがアセットのデジタルコンテンツよりも大きい場合など、現実的でない場合がある。
- 変更すべきでない資産を変更してしまう可能性がある場合など、適切でない場合。

> **注記** 既存のアセットに対してマニフェストを作成する場合が良い例です。

11.5. 外部マニフェストへの参照の埋め込み

アセットに XMP が埋め込まれており、C2PA マニフェストが外部に保存される場合、クレームジェネレータは XMP に `dcterms:provenance` キーを追加し、その値（URI 参照）をアクティブなマニフェストの場所とすることをお勧めします。

> **注記** この仕様の以前のバージョンでは、埋め込みマニフェストへの参照にもこの方法を使用することを推奨していました。現在では、このメカニズムは外部マニフェスト専用です。

フォントはXMPをサポートしないため、リモートC2PAマニフェストストアへのURIを指定する同等の方法については、本項のフォントに関する記述を参照のこと。

第12章 エンティティ図

図11「C2PAエンティティ図」は、C2PAシステムのすべての構成要素がどのように統合され、相互に関連しているかを示しています。

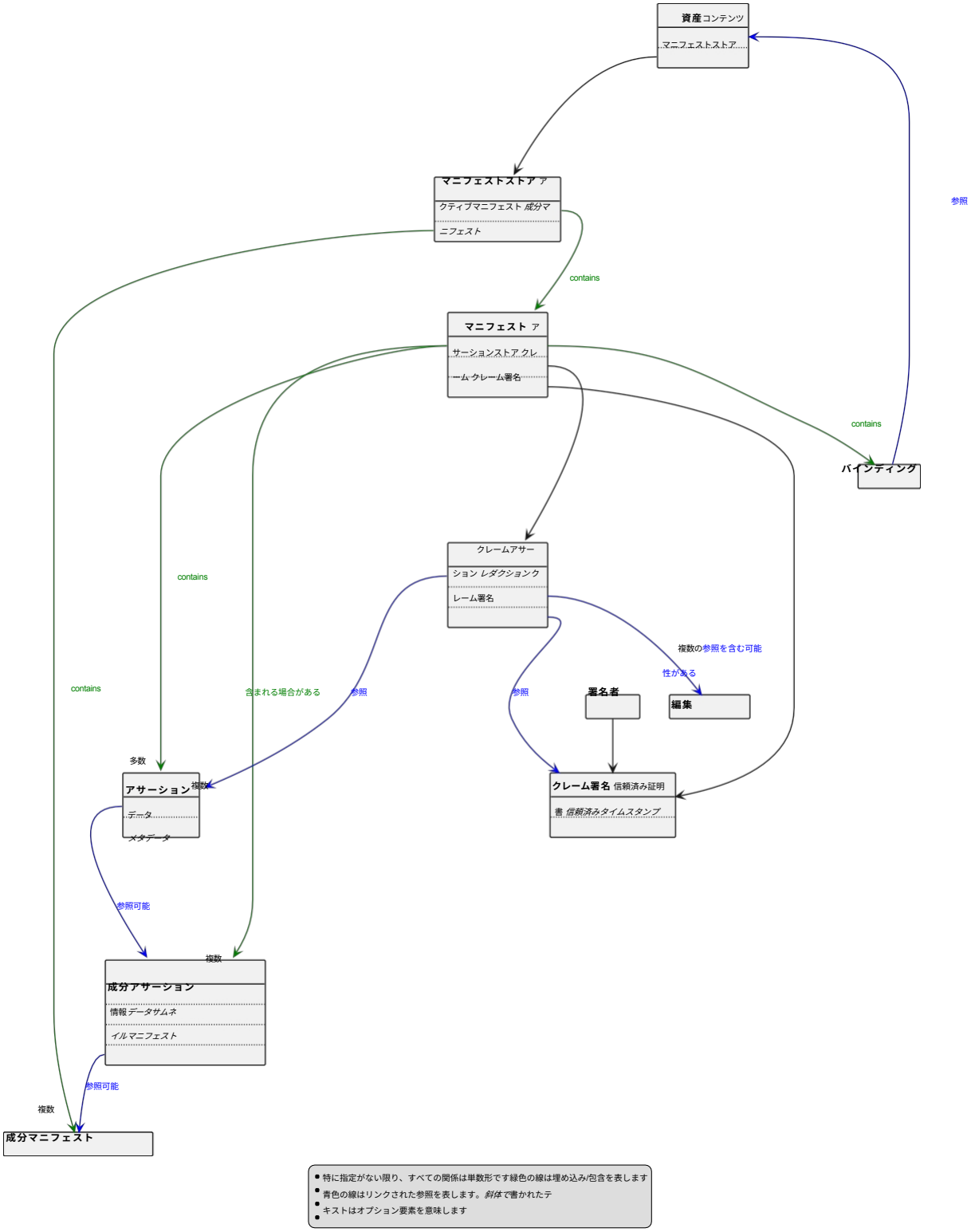

図11. C2PAエンティティ図

第13章 暗号技術

13.1. ハッシュ

本仕様の技術要件に従って適用されるすべての暗号ハッシュは、本節で説明するハッシュアルゴリズムのいずれかを使用して生成されなければならない。本節では以下の両方を定義する：

- 新規コンテンツのハッシュ生成に使用が許可され、かつ既存コンテンツのハッシュ検証に必須とされるハッシュアルゴリズムの一覧（許可リスト）；
- 既存コンテンツのハッシュ検証にはサポートが必須であるが、新規コンテンツのハッシュ生成には使用が許可されないハッシュアルゴリズムの一覧（非推奨リスト）。

注記 本節は、セクション18.10「ソフトバインディング」で説明されるソフトバインディングに使用されるアルゴリズムを規定するものではない。

注記 本セクションは、本仕様の外部で定義されるカスタムアサーションで使用されるアルゴリズムを規定するものではない。

アルゴリズムは、いずれかのリストにのみ記載されるものとする。複数の出力長（SHA2の各種長など）でインスタンス化されるアルゴリズムは、それぞれ異なるアルゴリズムと見なされ、各インスタンス化は個別に記載されるものとする。いずれのリストにも記載されていないアルゴリズムは禁止され、使用またはサポートしてはならない。アルゴリズムを禁止するために、リストからアルゴリズムを削除することができる。このため、実装はオプションベースで追加のアルゴリズムをサポートしてはならない。

実装者は、ソフトウェア更新をリリースする際、仕様の現行版における本節を参照し、サポートするアルゴリズムがこれに準拠していることを確認すべきである。

これらのリストは、ハッシュ生成に許可されるアルゴリズムと、C2PAデータ構造の対応するフィールドにおけるアルゴリズム識別子（通常は`alg`と呼ばれる）として使用する文字列アルゴリズム識別子を規定する。ハッシュ関数の出力は、宣言された長さを伴うバイト列（メジャータイプ2）としてCBORにエンコードされたバイナリ値として格納されなければならない。ハッシュ関数の出力を含むフィールドがある場合、同じ構造体内、または包含構造体内、あるいはclaim-mapまたはclaim-map-v2構造体内に、使用されたアルゴリズムを宣言するアルゴリズム識別子文字列フィールドが存在しなければならない。ハッシュアルゴリズム識別子フィールドはこれらの場所のいずれか1か所にのみ存在すべきであるが、構造体とその包含構造体内に複数存在する場合、最も近い識別子を使用する。「最も近い」とは、まずハッシュ値の兄弟フィールドである識別子を指し、次にルート構造体までの直近の囲み構造体を指す。これらのいずれの場所にも識別子が存在しない場合、`claim-map` または `claim-map-v2` 構造体の `alg` フィールドを使用する。

許可リストは以下の通りです:

- SHA2-256 ("sha256");
- SHA2-384（「sha384」）；

- SHA2-512（「sha512」）。

注記

SHA-3ファミリーのハッシュアルゴリズムは、デジタル署名アルゴリズムの許可リストとの整合性を保つため、許可リストに含まれていません。これは、COSEがSHA-3アルゴリズムをハッシュアルゴリズムとして使用するデジタル署名アルゴリズムをまだ確立していないためです。

非推奨リストは空です。

13.2. デジタル署名

本仕様の技術要件に従って適用されるすべてのデジタル署名は、本節に記載されたデジタル署名アルゴリズムおよび鍵タイプのうちいずれかを使用して生成されなければならない。本節では以下の両方を定義する：

- 新しいクレーム署名の生成に許可され、かつ既存のクレーム署名の検証に必要なデジタル署名アルゴリズムおよび鍵タイプのリスト（許可リスト）；
- 既存のクレーム署名の検証にはサポートが必須であるが、新規クレーム署名の生成には許可されないデジタル署名アルゴリズムおよび鍵タイプのリスト（非推奨リスト）。

注記

このセクションは、本仕様の外部で定義されるカスタムアサーションで使用されるデジタル署名には適用されません。

これらのリストは、COSE のアルゴリズムおよび CBOR 識別子へのマッピングを定義する関連規格（RFC 8152 および RFC 8230 を含むがこれらに限定されない）からのアルゴリズム識別子を参照することにより、許可されるアルゴリズムおよび鍵タイプを確立する。これらの規格は、署名方式で使用されるハッシュアルゴリズムも規定する。第13.1節「ハッシュ処理」の規定は、本仕様におけるハッシュアルゴリズムの使用には適用されない。以下のデジタル署名アルゴリズムおよび鍵タイプにデジタル署名アルゴリズムが含まれる場合、署名方式におけるその指定ハッシュアルゴリズムの使用は許可され、従うものとする。

注記

以下のリスト内の括弧書きは、読者の理解を助けるための補足説明です。

13.2.1. 署名アルゴリズム

許可リストは以下の通りです：

- ES256（SHA-256を用いたECDSA）；
- ES384（SHA-384を用いたECDSA）；
- ES512（SHA-512を用いたECDSA）；
- PS256 (SHA-256を使用したRSASSA-PSSおよびSHA-256を使用したMGF1);
- PS384 (SHA-384を使用したRSASSA-PSSおよびSHA-384を使用したMGF1);
- PS512 (SHA-512を使用したRSASSA-PSSおよびSHA-512を使用したMGF1);
- EdDSA（エドワーズ曲線DSA）。

- Ed25519インスタンスのみ。他のEdDSAインスタンスは許可されません。

非推奨リストは空です。

実装は、署名または検証操作に提供される鍵が選択されたアルゴリズムに対して正しいことを確認する必要があります。これは、RFC 8152 のセクション 8.1（ECDSA）、RFC 8152 のセクション 8.2（EdDSA）、および RFC 8230 のセクション 2 およびセクション 4（RSASSA-PSS）で要求されているとおりです。

これらの要件を便宜上以下に要約します：

- ECDSAでは、P-256、P-384、またはP-521楕円曲線上の楕円曲線鍵が必要です。
    - `ES256`にはP-256鍵、`ES384`にはP-384鍵、`ES512`にはP-521鍵の使用が推奨されますが、必須ではありません。実装は、すべてのECDSAアルゴリズム選択において、これらの曲線上の鍵を受け入れるものとします。
- Ed25519はedwards25519楕円曲線上の楕円曲線鍵を必要とします。
- RSASSA-PSSでは、少なくとも2048ビットのモジュラス長を持つRSA鍵が必要です。

実装は、アルゴリズム選択に適合しない鍵を用いた署名の生成または検証を拒否しなければならない。実装は、係数長が16384ビットを超えるRSA鍵を拒否してもよい。

13.2.2. COSEの使用

CBOR エンコードされたクレームの署名は、RFC 8152 のセクション 4.2 および 4.4 で説明されている CBOR オブジェクト署名および暗号化 (COSE) によって生成されます。

> **NOTE**
>
> ペイロードは、COSE署名内に存在するか、別個に輸送される（「分離型」RFC 8152のセクション4.1で説明されている「分離されたコンテンツ」として）。「分離されたコンテンツ」モードでは、署名付きデータはCOSE_Sign1_Tagged構造体の外部に保存され、COSE_Sign1_Tagged構造体のペイロードフィールドは常に`nil`となる。

ペイロードが COSE_Sign1_Tagged 署名内に存在するかどうか、あるいは分離されているかに関わらず、デジタル署名の計算または検証のために構築されるメモリ内の `Sig_structure` のペイロードフィールドの内容は、本仕様におけるデジタル署名の特定の用途で記述されている外部データで埋められるものとする。`Sig_structure` のペイロードフィールドが `nil` になることは決してない。

標準または更新マニフェストの署名を計算または検証する場合、`Sig_structure`のペイロードフィールドには、セクション10.3.2.4「クレームへの署名」およびセクション11.1「JUMBFの使用」で説明されているクレームJUMBFボックスの内容が含まれる。

13.2.3. 署名の計算

署名はRFC 8152のセクション4.4に記載されている方法で計算または検証される。`Sig_structure`の構築には以下の追加要件が適用される：

- context要素の値は、本仕様におけるデジタル署名の特定の用途で`CounterSignature`の使用が指定されている場合を除き、`Signature1`とする。`Signature`は使用してはならない。
- ペイロード要素の値は、本仕様におけるデジタル署名の各用途ごとに指定される。

- external_aad要素は長さゼロのbstrとする。外部認証データは使用しない。
- 署名アルゴリズムを指定するalgヘッダーは、RFC 8152のセクション3.1で定義されるbody_protected要素内に存在しなければならない。

> **注記**
>
> algヘッダーは標準的なCOSEヘッダーであるため、常に保護されたIANA COSEヘッダーパラメータレジストリで規定されているように、ラベルとして整数1を持つヘッダーマップ。リテラル文字列algはラベルとして使用されない。COSE_Sign1を使用する場合、sign_protected要素は常に省略される。

C2PA構造内の全てのデジタル署名は、RFC 8152セクションに定義されるCOSE_Sign1_Tagged構造とする。
4.2 に定義される COSE_Sign1_Tagged 構造体とする。COSE_Sign1_Tagged の構築には、以下の追加要件が適用される：

- 上記Sig_structure内のalgヘッダーは、保護ヘッダーバケットにも存在しなければならない。
- ペイロードフィールドの値およびペイロードが署名内に存在するか分離されているかは、本仕様におけるデジタル署名の各使用例で指定される。ペイロードが分離と指定された場合、ここでの値はnilとする。逆に、ペイロードが署名内に存在する場合は、ペイロードのバイナリ内容がbstrとしてこのフィールドに格納される。

> **注記**
>
> COSEはRFC 8152のセクション1.3においてnilをメジャータイプ7、値22と定義し、この値を使用する。長さゼロのバイト配列（メジャータイプ2）は、分離コンテンツを示すために使用できません。

13.2.4. 署名時刻の主張の追加

署名生成者は、数値日付（NumericDate）を値とする IAT 保護ヘッダーを追加することで、「署名時刻の主張」を確立することもできる。存在する場合、これは署名が生成された時刻を表すものとする。

> **注記**
>
> NumericDate は、RFC 8949 のセクション 3.4.2 で説明されている CBOR 数値日付ですが、先頭のタグ 1（エポックベースの日付/時刻）が省略されています。この仕様の他の箇所では使用されません。

> **注記**
>
> 本勧告は、証明書有効性検証には使用されないがユーザーエクスペリエンスで利用可能な非信頼タイムスタンプを提供するためのJAdESの進行中更新に基づいています。クレーム生成者が信頼できる時刻ソースにアクセスできないが、署名時刻を提供したいシナリオで有用です。

13.2.5. 署名検証

署名生成時に、クレーム生成者が検証者としても機能する場合、クレーム生成者は第14章「*信頼モデル*」に従い署名認証情報が許容可能かどうかを検証し、許容できない場合は警告を生成すべきである。クレーム生成者は、必要に応じて当該認証情報による署名を許可してもよい。これは、ローカルのクレーム生成者の検証者が、資産の想定される対象者が使用する検証者とは異なる構成を有することが既知の場合に望ましい。

13.2.6. 暗号学的検証

署名の検証時には、メモリ内にSig_structureが生成される。そのbody_protectedフィールドには、COSE_Sign1_Tagged構造体（RFC 8152、セクション4.4）の保護されたヘッダーバケットの内容が格納される。ペイロードフィールドについては、署名内でペイロードの存在が指定されていた場合、COSE_Sign1_Tagged構造体のペイロードフィールドから値が取得される。ペイロードが分離されていると指定された場合、COSE_Sign1_Tagged構造体のペイロードフィールドはNULLとなります。この場合、Sig_structureのペイロードフィールドの内容は、署名の生成に使用されたのと同じ外部ソースから設定されるものとします。これらは、本仕様書内でデジタル署名を使用する箇所で定義されています。

13.2.7. 署名者アイコンの包含

C2PAマニフェストコンシューマーは、署名者のアイコンまたはロゴを表示したい場合がある。そのようなグラフィックを見つけるには、RFC 9399で定義されるロゴタイプを埋め込み証明書内で検索する。ロゴタイプが存在しない場合、マニフェストコンシューマーは実装依存の方法で他のソースからのアイコンまたはロゴを使用できる。

第14章 信頼モデル

注記 本節における「ユーザー」とは、消費および作成シナリオにおいてC2PA準拠バリデータを利用する人間の行為者を指す。

14.1. 概要

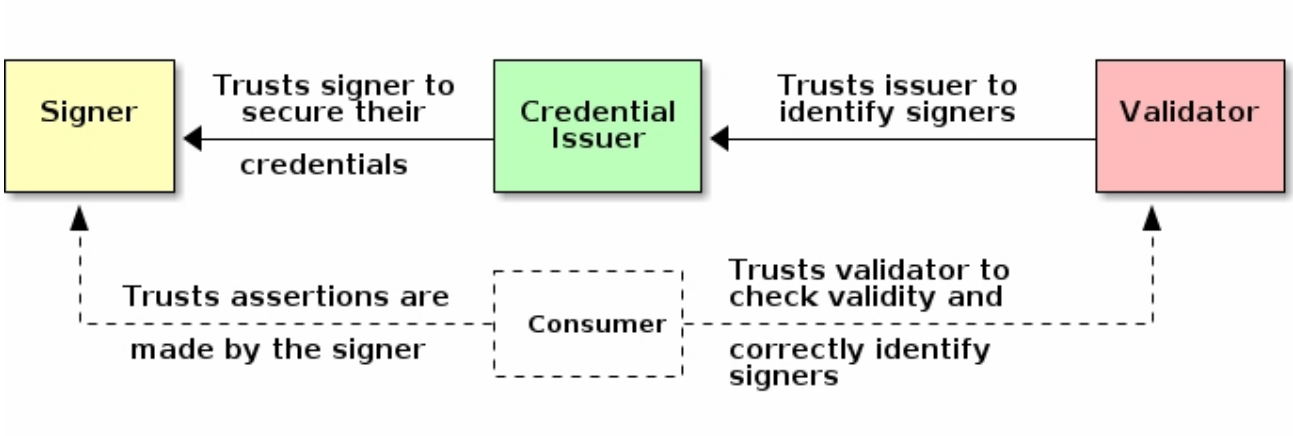

図12. C2PA信頼モデル図

図12「C2PA信頼モデル図」は、署名者の身元に対する信頼を扱う信頼モデルで規定された3つのエンティティを、黄色、緑色、赤色で示している。下部の破線は、署名者の身元とその他の信頼シグナルを用いて資産に関するアサーションの真偽を判断する消費者（信頼モデルでは規定されていない）を表す。

14.2. 署名者の身元

信頼モデルにおけるアイデンティティとは、暗号署名鍵（別名：クレデンシャル）を署名者と関連付ける手段であり、その鍵で署名されたクレーム署名または構造（アサーションやクレームを含むがこれらに限定されない）に基づいて信頼判断を行うための基盤となるものである。

認証情報は、すべてのC2PAマニフェストにおけるデジタル署名に使用されるCOSE_Sign1_Tagged構造体のCOSE保護ヘッダーにリストされるものとする。保護ヘッダーと非保護ヘッダーの結合集合には、同一の識別認証情報が厳密に1回のみ出現するものとする。認証情報を含まない、または2つ以上の認証情報を含むCOSE_Sign1_Tagged構造体は拒否される。同一の認証情報を複数回繰り返す場合（保護ヘッダーと非保護ヘッダーに別々に含める場合を含む）も、2つ以上の認証情報に該当し、拒否される。

注記 本仕様の旧バージョンでは、COSE非保護ヘッダーへのクレデンシャル記載も許可されていた。

クレデンシャルがヘッダ値にどのように格納されるか、信頼チェーンがどのように構築されるかは、セクション 14.5「X.509 証明書」で規定されており、追加情報も同セクションに記載されています。

14.3. 検証状態

14.3.1. 一般

バリデータは、その資産および関連するアクティブなマニフェストについて検証ステートメントを生成するマニフェストコンシューマーである。これらのステートメントを取得するプロセスは検証セクションで説明されている。資産を消費するアクター（通常はユーザーエージェントとそのユーザーインターフェースを介して）は、それらのステートメントを解釈し、消費している資産の出所に関する独自の結論のセットを導き出さなければならない。これらの結論は、ステートメントと資産自体の内容から導かれる。

14.3.2. マニフェストの状態

これらのステートメントに基づき、C2PAマニフェストは以下のいずれかの状態となる可能性があります：

- 整形式
- 有効
- 信頼済み

注記 信頼されたマニフェストはすべて有効であり、有効なマニフェストはすべて形式が正しい。

14.3.3. アセットの状態

バリデータが、アクティブなマニフェストが作成されてからコンテンツバインディングの対象となるアセットの部分が変更されていないことを報告し [セクション 15.12、「アセットのコンテンツの検証」]、そのアクティブなマニフェストが有効または信頼されている場合、アセット自体は有効なアセットとなります。

14.3.4. 整形式マニフェスト

C2PAマニフェストは、以下の各条件が検証により真であると判断された場合に、形式的に正しい（Well-Formed）とされる：

- マニフェストの内容が、本仕様の規範的要件に準拠していること（検証プロセスにより検証される）。
- マニフェストの特定のタイプに対して許可されているアサーションのみが存在すること［セクション 15.10.1「マニフェストのタイプに対する正しいアサーションの検証」］。
- マニフェストの主張は、主張に関するすべての要件を満たしている［セクション15.10.3「主張の検証」］。
- マニフェストに含まれるすべての成分は、成分に関するすべての要件を満たしている［セクション15.11「成分の検証」］。

14.3.5. 有効なマニフェスト

C2PAマニフェストは、以下の各項目が真であると検証によって判断された場合に有効である：

- マニフェストが適切に構成されている［セクション14.3.4「適切に構成されたマニフェスト」］。
- マニフェストは署名されて以来変更されていません［セクション13.2.6「暗号による検証」］。
- クレーム署名は、success code として `claimSignature.validated` を受け取ります [セクション 15.7、「署名の検証」]。
- 検証の署名期間内のクレーム署名は、成功コード `claimSignature.insideValidity` [セクション 15.8「タイムスタンプの検証」]。
- C2PA マニフェストの署名者の認証情報は、署名認証情報失効（`signingCredential.ocsp.revoked`）または署名認証情報不明（`signingCredential.ocsp.unknown`）の失敗コードで拒否されない [セクション 15.9、「認証情報失効情報の検証」]。

C2PAマニフェストが有効な場合、そのマニフェストのクレームは、クレームの
クレームの `claim_generator_info` フィールド [セクション 10.2.3「クレーム生成元情報」] で識別される。

14.3.6. 信頼されたマニフェスト

C2PAマニフェストは、以下の各条件が検証により真であると判断された場合に信頼される：

- マニフェストが有効である［セクション14.3.5「有効なマニフェスト」］。
- C2PAマニフェストの署名クレデンシャルが`signingCredential.trusted`の成功コードを受信している
[セクション15.7「署名の検証」]。

14.4. 信頼リスト

14.4.1. C2PA署名者

検証者は、C2PA署名者を評価するために以下の情報を維持しなければならない：

- 受け入れられる拡張キー使用法（EKU）値のリスト。
- 各受け入れられる EKU 値に対する X.509 証明書信頼アンカーのリスト。

`c2pa-kp-claimSigning` (1.3.6.1.4.1.62558.2.1) EKU については、信頼アンカーのリストには、C2PA が提供する署名者信頼アンカー（すなわち C2PA 信頼リスト）を含める必要があるが、これに限定されるものではない。

注記	これらのリストの一部は空でもよい。

C2PA Trust List で提供される `c2pa-kp-claimSigning` EKU の信頼アンカーリストに加えて、バリデータは、その EKU および/または他の EKU (`id-kp-emailProtection` (1.3.6.1.5.5.7.3.4）や `id-kp-documentSigning` (1.3.6.1.5.5.7.3.36) など）に対して、ユーザーが追加の信頼アンカーを設定できるようにすべきである。バリデータは、デフォルトのオプションを提供するか、ユーザーがオプトインできる外部団体が管理するリストを提供し、C2PA 署名者のバリデータの信頼アンカーストアに情報を登録できるようにすべきである。

注記	この仕様の以前のバージョンでは、`id-kp-emailProtection` または `id-kp-claimSigning` EKU の存在が要求されていました

。したがって、署名者の証明書にこれらの EKU の少なくとも 1 つを含めると、c2pa-kp-claimSigning と併せて、古いバリデータとの互換性を向上させることができます。

> kp-documentSigning EKUの存在を要求していました。したがって、署名者の証明書にこれらの2つのEKUのうち少なくとも1つをc2pa-kp-claimSigningとともに含めることで、古いバリデータとの互換性を向上させることができます。

14.4.2. タイムスタンプ機関

バリデータは、タイムスタンプ機関（TSA）用のX.509証明書信頼アンカーのリストを維持しなければならない。このリストはC2PA署名者用のリストとは別個のものでなければならない。このリストには、C2PAが提供するタイムスタンプ機関信頼アンカー（すなわちC2PA TSA信頼リスト）を含める必要があるが、これに限定されない。

注記	このリストは空でもよい。

C2PA TSA信頼リストで提供される信頼アンカーのリストに加え、バリデータはユーザーが追加のTSA信頼アンカーストアを設定できるようにすべきであり、デフォルトオプションを提供するか、ユーザーが選択可能な外部団体が管理するリストを提供し、タイムスタンプ機関向けのバリデータの信頼アンカーストアを充填できるようにすべきである。

14.4.3. プライベート認証情報ストレージ

バリデータは、ユーザーが署名用資格情報のプライベートな資格情報ストアを作成・維持することを許可する場合もある。このストアは、帯域外の関係に基づいて信頼することを選択した資格情報の「アドレス帳」として意図されている。プライベート認証情報ストアが存在する場合、署名付きC2PAマニフェストの検証にのみ適用され、タイムスタンプの検証には適用されない。また、プライベート認証情報ストアが存在する場合、署名者認証情報の直接的な信頼のみを許可する。プライベート認証情報ストアのエントリは認証情報を発行できず、検証時の信頼アンカーとして含めることはできない。

検証者は、プライベート認証情報ストアに事前設定されたエントリを一切保持してはならない。

バリデータは、ユーザーからの資格情報の信頼要求に応じてのみ、プライベート資格情報ストアにエントリを追加する。同様に、バリデータは、ユーザーからの資格情報の信頼停止要求に応じてのみ、プライベート資格情報ストアからエントリを削除する。

14.5. X.509 証明書

X.509証明書は、RFC 9360（*CBORオブジェクト署名および暗号化（COSE）：X.509証明書の伝送および参照のためのヘッダーパラメータ*）で定義されている形式で保存される。便宜上、x5chainの定義を以下に引用する。

重要	本仕様は、引用文の後に記載するRFC 9360を超える追加要件を規定します。特に、本仕様ではすべての中間証明書に対して以下を要求します 署名者の証明書チェーンに含まれるすべての中間認証局の証明書をx5chainヘッダーに含めること、およびクレーム生成者が常にx5chainヘッダーを保護されたヘッダーバケットに配置することを要求します。

> x5chain: このヘッダーパラメータは、X.509証明書の順序付き配列を含みます。証明書は、エンドエンティティ鍵を含む証明書から始まり、それを署名した証明書が続く順序で並べられます。全体の

> 要素内にチェーンが存在する場合、依存側が既に不足している証明書を所有しているか、またはその所在を特定できると信じる理由がある場合に限る。これは、依存側が依然としてパス構築を行う必要があるが、このヘッダパラメータで候補パスが提案されることを意味する。
>
> 信頼メカニズムは、このパラメータ内の証明書をすべて信頼できない入力として処理しなければならない。パラメータ内に自己署名証明書が存在しても、帯域外確認なしに信頼アンカーのセットを更新してはならない。このヘッダパラメータの内容は信頼できない入力であるため、ヘッダパラメータは保護されたヘッダバケットまたは保護されていないヘッダバケットのいずれかに配置できる。保護されていないヘッダバケットでヘッダパラメータを送信すると、中継者が証明書を削除または追加できる。
>
> エンドエンティティ証明書は、COSEによって完全性が保護されなければならない。これは例えば、保護されたヘッダーにヘッダーパラメータを送信する、保護されていないヘッダーに「x5chain」を送信すると同時に保護されたヘッダーに「x5t」を組み合わせる、あるいは外部AADにエンドエンティティ証明書を含めることで実現できる。
>
> このヘッダーパラメータにより、単一のX.509証明書またはX.509証明書のチェーンをメッセージ内で伝送することが可能となる。
>
> - 単一の証明書が伝送される場合、それはCBORバイト文字列に格納される。
> - 複数の証明書が伝送される場合、各証明書が個別のバイト文字列で構成される CBOR 配列が使用されます。

バリデータは、その信頼アンカーの証明書のみを保持することが想定される。したがって、署名の一部としてx5chainヘッダーを作成する際、クレーム生成器は署名者の証明書およびすべての中間認証局をヘッダーの値に含めなければならない。信頼アンカーの証明書（ルート証明書とも呼ばれる）は含めてはならない。

最初の、または唯一の証明書のsubjectPublicKeyInfo要素が、署名の検証に使用される公開鍵となる。tbsCertificateシーケンスのvalidity要素は、証明書の有効期間を提供する。

本仕様の旧版では、整数ラベル33が標準化されないという可能性が低いものの回避のため、クレーム生成者が文字列ラベルx5chainのみを記述することを要求していた。

整数ラベル33は標準化され、本仕様ではこれを標準として採用し、文字列ラベルの使用を非推奨とします。したがって：

- クレーム生成者は、このヘッダーをCOSE署名に挿入する際、ラベルとして整数33のみを使用すべきである。クレーム生成者は文字列ラベルx5chainの記述を継続できるが、この動作は非推奨となり、クレーム

生成者は整数ラベルのみを使用するよう更新すべきである。クレーム生成者は、上記で要求される通り、このヘッダーをCOSE署名の保護されたヘッダーバケットにのみ配置しなければならない。

- バリデータは、このヘッダーのラベルとして文字列 `x5chain` または整数 `33` のいずれかを受け入れるものとする。両方のラベルが存在する場合、バリデータは整数ラベル `33` を持つヘッダーを使用し、文字列 `x5chain` をラベルとするヘッダーを無視するものとする。バリデータは、この仕様の以前のバージョンとの互換性を維持するため、保護されたバケットまたは保護されていないバケットのいずれかからのヘッダーを受け入れるものとする。セクション14.2「署名者の同一性」に従い、このヘッダーが同一ラベルで保護バケットと非保護バケットの両方に存在する場合、バリデータは複数の認証情報が存在するため、クレーム署名を不正な形式として拒否しなければならない。

14.5.1. 証明書プロファイル

14.5.1.1. 一般要件

このセクションでは、セクション15.7「署名の検証」に記載されているように、X.509証明書が署名認証情報として受け入れ可能であることを検証するための要件を定義する。

すべての証明書は、以下の要件を満たさなければならない。

- signatureAlgorithmフィールドのalgorithmフィールドは、以下のいずれかの値でなければならない：

ecdsa-with-SHA256

RFC 5758、セクション3.2

ecdsa-with-SHA384

RFC 5758、セクション 3.2

ecdsa-with-SHA512

RFC 5758、セクション 3.2

sha256WithRSAEncryption

RFC 8017、付録 A.2.4

sha384WithRSAEncryption

RFC 8017、付録 A.2.4

sha512WithRSAEncryption

RFC 8017、付録 A.2.4

id-RSASSA-PSS

RFC 8017、付録 A.2.3

id-Ed25519

RFC 8410 セクション 3

- 署名アルゴリズムフィールドのアルゴリズムフィールドがid-RSASSA-PSSの場合、パラメータフィールドはRSASSA-PSS-params型である。そのフィールドはRFC 8017付録A.2.3で定義される以下の要件を満たすものとする：
  - hashAlgorithmフィールドが存在すること。
  - hashAlgorithmフィールドのalgorithmフィールドは、RFC 8017付録B.1で定義される以下の値のいずれかでなければならない：
    - id-sha256。
    - id-sha384.
    - id-sha512.
  - maskGenAlgorithm フィールドは存在しなければならない。
  - maskGenAlgorithm フィールドの parameters フィールドの algorithm フィールドは、hashAlgorithmフィールドのalgorithmフィールドと等しくなければならない。
- 証明書のsubjectPublicKeyInfoのalgorithmフィールドがid-ecPublicKeyである場合、parametersフィールドはRFC 5480セクション2.1.1.1で定義される以下の命名曲線の一つでなければならない：
  - prime256v1.
  - secp384r1.
  - secp521r1.
- 証明書の subjectPublicKeyInfo のアルゴリズムフィールドのアルゴリズムフィールドが rsaEncryption または id-RSASSA-PSS の場合、パラメータフィールドのモジュラスフィールドの長さは少なくとも 2048 ビットでなければならない。

X.509 証明書用のプライベート認証情報ストアにある証明書を除くすべての証明書は、受け入れられるために以下の追加要件を満たす必要があります。

- バージョンは、RFC 5280 のセクション 4.1.2.1 に準拠した v3 であること。
- TBSCertificateシーケンスのissuerUniqueIDおよびsubjectUniqueIDオプションフィールドは、RFC 5280セクション4.1.2.8に従い、存在してはならない。
- 基本制約拡張はRFC 5280セクション4.2.1.9に従うものとする。特に、以下のいずれかが真であること：
  - 認証済み公開鍵が証明書署名の検証に使用可能な場合、Basic Constraints拡張が存在し、cAブール値がアサートされなければならない。
  - 認証済み公開鍵が証明書署名の検証に使用できない場合、基本制約拡張は存在しないか、拡張内のcAブール値がアサートされず、かつキー使用法拡張内のkeyCertSignビットがアサートされない。
- RFC 5280の4.2.1.1節に従い、自己署名証明書以外の証明書にはすべて、権限キー識別子拡張が含まれていなければならない。

- RFC 5280の4.2.1.2節で規定されている通り、Subject Key Identifier拡張はCAとして機能するあらゆる証明書に存在しなければならない。エンドエンティティ証明書にも存在すべきである。
- RFC 5280の4.2.1.3節で規定されている通り、Key Usage拡張は存在し、かつクリティカルとしてマークされるべきである。C2PAマニフェストの署名に使用される証明書は、`digitalSignature`ビットをアサートしなければならない。`keyCertSign`ビットは、Basic Constraints拡張で`cA`ブール値がアサートされている場合にのみアサートされるべきである。
- 基本制約拡張が存在しない、または`cA`ブール値がアサートされていない証明書では、RFC 5280セクション4.2.1.12に従い、拡張キー使用法（EKU）拡張が存在し、かつ空でないものとする。これらは一般に「エンドエンティティ」または「リーフ」証明書と呼ばれる。
  - `anyExtendedKeyUsage` EKU（2.5.29.37.0）は存在してはならない。
  - タイムスタンピングの副署を署名する証明書は、`id-kp-timeStamping` (1.3.6.1.5.5.7.3.8) 目的に対して有効である。
  - 証明書に対する OCSP 応答に署名する証明書は、`id-kp-OCSPSigning` (1.3.6.1.5.5.7.3.9) 目的に対して有効である。
  - 証明書が `id-kp-timeStamping` または `id-kp-OCSPSigning` のいずれかに有効である場合、その証明書は、これら 2 つの目的のうち、正確に 1 つにのみ有効であり、他の目的には有効ではないものとします。
  - 証明書は、上記に列挙された目的以外には有効であってはならない。ただし、本プロファイルに記載されていない、かつ構成ストア内のEKUリストに含まれないEKUが存在しても、証明書が拒否されることはない。

14.5.1.2. 証明書信頼チェーン

署名認証情報として証明書を検証する場合、その証明書が
タイムスタンプの検証時には、プライベート認証情報ストアは参照されない。

証明書がプライベート認証情報ストアに存在しない場合、またはバリデータがそれを実装していない場合、信頼チェーンはRFC 5280セクション6の手順に従い、要求される特定の目的（署名、タイムスタンプ、またはOCSP署名）およびその目的に適した信頼アンカーストアに対して構築され検証されるものとする。当該検証アルゴリズムの失敗は、チェーンが拒否されることを意味する。証明書チェーン構築時には、プライベート認証情報ストアは決して含まれない。プライベート認証情報ストア内の証明書はCAとして機能できない。

C2PAクレームまたはタイムスタンプの署名には、エンドエンティティ証明書のみを使用すること。CA証明書はこれらの目的で使用してはならない。C2PAクレーム、タイムスタンプ、またはOCSP応答の署名を検証するために使用されるCA証明書（基本制約拡張機能内の`cA`ブール値がアサートされているもの）は、署名資格情報.信頼できない（`signingCredential.untrusted`）の失敗コードで拒否される。

検証者は、署名証明書が使用目的に対して認可されていることを確認し、認可されていない目的で使用される証明書を拒否しなければならない。証明書が特定の目的に対して認可されているとは、その目的の拡張鍵使用法（EKU）オブジェクト識別子（OID）が証明書の拡張鍵使用法拡張（RFC 5280、セクション4.2.1.12）に含まれている場合を指す。

C2PAクレームの署名に使用される証明書を検証する際、署名証明書は、検証者が関連付けられた信頼アンカーのリストを保持するEKU（セクション14.4.1「C2PA署名者」参照）を少なくとも1つ有している必要があり、検証者は

証明書に含まれるEKUに関連付けられた信頼アンカーのみを使用しなければならない。

タイムスタンプの署名に使用される証明書チェーンを検証する場合、署名証明書は `id-kp-timeStamping` (1.3.6.1.5.5.7.3.8) EKU を持つものとする。

OCSP応答の署名に使用される証明書チェーンを検証する場合、署名証明書は`id-kp-OCSPSigning`（1.3.6.1.5.5.7.3.9）EKUを有しなければならない。

X.509証明書用のプライベート認証情報ストアを通じて受け入れられる証明書を除き、検証者は証明書の証明書プロファイルへの準拠性を検証し、準拠しない証明書を拒否しなければならない。これには、拡張キー使用法拡張機能の存在を要求すること、および証明書が本節に記載された3つの目的（C2PA署名、タイムスタンプ署名、OCSP応答署名）のうち1つを超えて許可されていないことを要求することが含まれる。

証明書プロファイルに記載されている通り、証明書を発行する認証局（CA）証明書はEKU拡張を持つ必要がなく、通常は持たない。存在する場合、無視されるものとする。この要件は、C2PAマニフェスト、タイムスタンプ、またはOCSP応答に署名するエンドエンティティ証明書にのみ適用される。CA証明書は、C2PAマニフェスト、タイムスタンプ、またはOCSP応答の署名に使用してはならない。

14.5.2. 証明書の失効

X.509証明書は失効ステータス照会をサポートする。クレーム生成者は、失効処理を実装するためにオンライン証明書ステータスプロトコル（OCSP、RFC 6960）およびOCSPステープリング（RFC 6066セクション8で当初構想されたが、本項で記述される通り実装される）を使用すべきである。クレーム生成者は証明書失効リスト（CRL、RFC 5280）を使用してはならない。

注記

CRLを使用するには、各認証局の失効証明書リスト全体をダウンロードする必要があります
これは時間がかかる場合があります。CRLをOCSP応答と同様にステープル方式で含めることも可能ですが、CRLの潜在的なサイズがOCSP応答に比べて大きいことから、この方法も望ましくありません。

準拠した CA は、発行した証明書に AuthorityInfoAccess (AIA) 拡張 (RFC 5280、セクション 4.2.2.1) を含め、CA が運営する OCSP サービスへのアクセス情報を提供すべきである。

証明書にAIA拡張が含まれる場合、失効情報はCOSE_Sign1構造体の保護されていないヘッダーに、文字列ラベル`rVals`で格納され、値のスキーマは例3「`rVals`用CDDL」の`r`Valsルールに従うものとする：

例3. `rVals`のCDDL

```
; JSONスキーマに基づくrValsおよび関連構造のCBOR版（
https://www.etsi.org/deliver/etsi_ts/119100_119199/11918201/01.01.01_60/ts_11918201v010101p.pdfセクション5.3.5.2参照）
rVals = {
   "ocspVals": [1* bstr]
}
```

注記

上記の定義は、JAdESのセクション5.3.5.2にあるスキーマのサブセットをCBORに適応させたものであり、OCSP応答のみをバイナリ文字列として格納します。

クレーム署名前に、署名者の証明書にAIA拡張が含まれる場合、クレーム生成者は当該拡張で指定されたOCSPサービスを照会し、応答を取得してrValsヘッダーのocspVals配列要素に格納すべきである。クレーム署名に含める中間CA証明書についても同様の処理を行う必要がある。

クレーム受信時、ステープルされたOCSP応答はRFC 6960セクション3.2に従い検証される。

クレーム署名後の証明書失効状態の検証プロセスは、「認証情報の失効情報の検証」で詳細に記述されている。

第15章 検証

15.1. 検証プロセス

15.1.1. 説明

C2PAマニフェストの検証は、複数のステップからなるプロセスであり、マニフェスト内に含まれるアサーション、クレーム、および関連するクレーム署名の検証に加え、（アクティブなマニフェストの場合のみ）関連するハードバインディングの検証を含みます。この検証プロセスは、本条項で記述される検証アルゴリズムを実装するハードウェアまたはソフトウェアのアクターであるバリデータによって実行されます。

15.1.2. 検証のフェーズ

これらのフェーズは順不同で、以下の条項に記述されている：

- セクション15.10「アサーションの検証」：アサーションの検証。
- セクション15.11「成分の検証」：成分がある場合の検証。
- セクション15.8「タイムスタンプの検証」：タイムスタンプの検証。
- セクション 15.9、「認証情報の失効情報の検証」：認証情報の失効情報の検証。
- セクション 15.7、「署名の検証」：クレーム署名の検証。
- セクション 15.12、「アセットのコンテンツの検証」：アセットのコンテンツの検証。

セクション 14.3「検証状態」で説明されているように、C2PA マニフェストは、これらのステップの結果に基づいて、整形式、有効、または信頼済みと見なされる場合があります。

15.1.3. 視覚的表現

図13「クレームの検証」は、C2PAマニフェストを検証するプロセスの視覚的表現である。

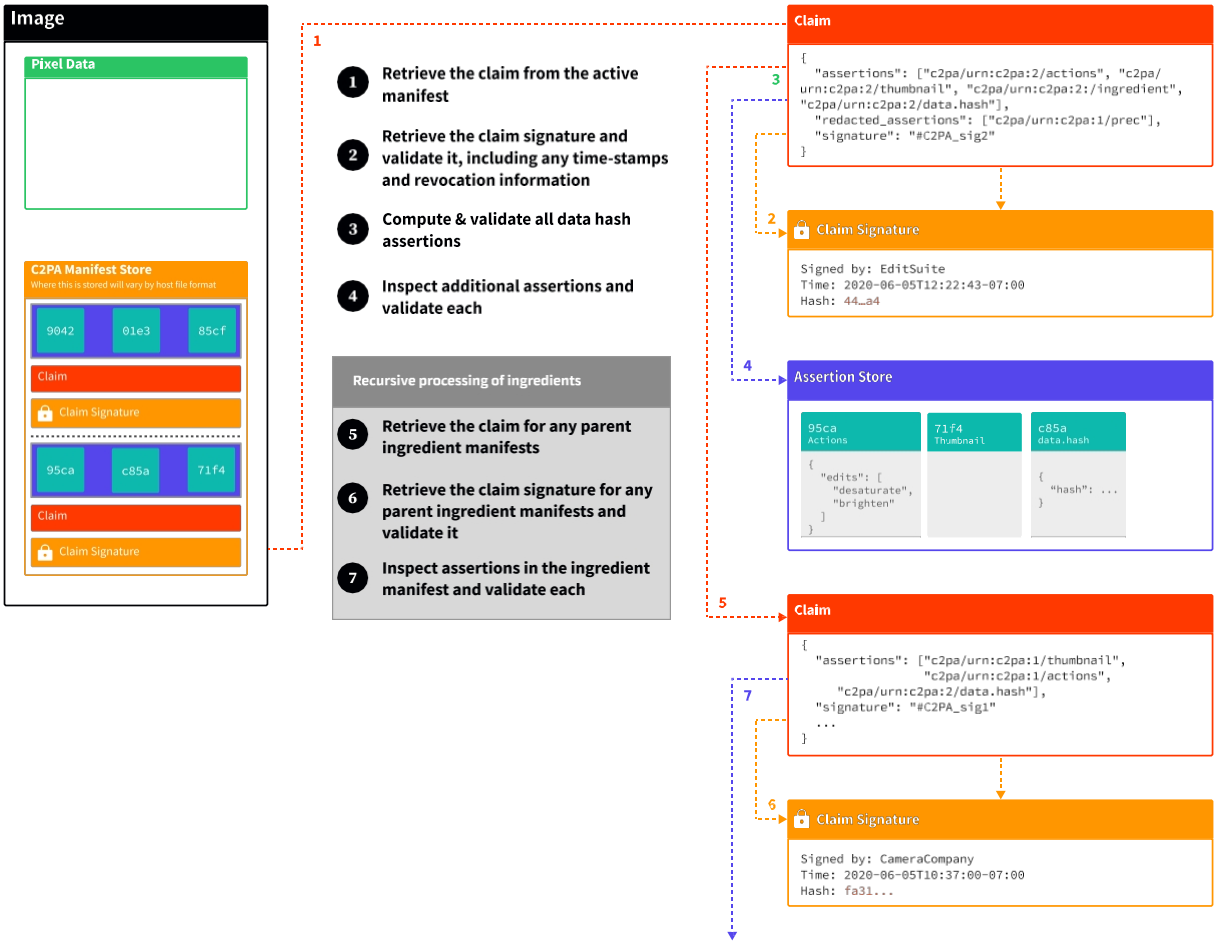

図13. クレームの検証

注記 図と本文に相違がある場合は、本文が優先されます。

15.2. 検証結果の返却

15.2.1. 一般

検証アルゴリズムは、資産のC2PAマニフェストストア内のすべてのマニフェスト（アクティブなマニフェストを含む）および成分アサーション経由で参照されるC2PAマニフェストストア内のその他すべてのマニフェストについて、統合された検証結果セットを返すものとする。

検証結果は、以下セクション15.2.2「標準ステータスコード」で定義される成功コード、情報コード、失敗コードの標準セットによって表現される。クレーム生成器がプロセス固有のステータス情報を記録する必要がある場合、カスタムステータスコードの使用も許可される。カスタムコードはエンティティ固有の名前空間（例：`com.litware`）と同じ構文に準拠しなければならない。

クレーム生成者が成分アサーションを介して成分アセットを追加する場合、検証者として機能し、本節で説明する完全な検証アルゴリズムを当該成分に対して実行しなければならない。クレーム生成者は検証結果を記録しなければならない。

検証結果　　成分、　　以下の　CDDL　定義　スキーマ、　　値

成分アサーション内のvalidationResultsフィールド。

検証結果のためのCDDL

```
; 検証コード

; 成功コード
$status-code /= "assertion.accessible"
$status-code /= "assertion.bmffHash.match"
$status-code /= "assertion.boxesHash.match"
$status-code /= "assertion.collectionHash.match"
$status-code /= "assertion.dataHash.match"
$status-code /= "assertion.hashedURI.match"
$status-code /= "claimSignature.insideValidity"
$status-code /= "claimSignature.validated"
$status-code /= "ingredient.claimSignature.validated"
$status-code /= "ingredient.manifest.validated"
$status-code /= "signingCredential.ocsp.notRevoked"
$status-code /= "signingCredential.trusted"
$status-code /= "timeStamp.trusted"
$status-code /= "timeStamp.validated"

; 情報コード
$status-code /= "ingredient.unknownProvenance"
$status-code /= "manifest.unknownProvenance"
$status-code /= "signingCredential.ocsp.inaccessible"
$status-code /= "署名認証情報.OCSP.スキップ済み"
$status-code /= "timeOfSigning.insideValidity"
$status-code /= "timeOfSigning.outsideValidity"
$status-code /= "タイムスタンプ不正"
$status-code /= "タイムスタンプ不一致"
$status-code /= "タイムスタンプが有効期間外"
$status-code /= "タイムスタンプが信頼できない"
$status-code /= "assertion.dataHash.additionalExclusionsPresent"

; 失敗コード
$status-code /= "algorithm.deprecated"
$status-code /= "algorithm.unsupported"
$status-code /= "assertion.action.ingredientMismatch"
$status-code /= "assertion.action.malformed"
$status-code /= "assertion.action.redacted"
$status-code /= "assertion.action.redactionMismatch"
$status-code /= "assertion.bmffHash.不正な形式"
$status-code /= "assertion.bmffHash.mismatch"
$status-code /= "assertion.boxesHash.mismatch"
$status-code /= "assertion.boxesHash.malformed"
$status-code /= "assertion.boxesHash.unknownBox"
$status-code /= "assertion.cloud-data.hardBinding"
$status-code /= "assertion.cloud-data.actions"
$status-code /= "assertion.cloud-data.hardBinding"
$status-code /= "assertion.cloud-data.不正な形式"
$status-code /= "assertion.collectionHash.incorrectFileCount"
$status-code /= "assertion.collectionHash.invalidURI"
$status-code /= "assertion.collectionHash.不正な形式"
$status-code /= "assertion.collectionHash.mismatch"
$status-code /= "assertion.dataHash.不正な形式"
$status-code /= "assertion.dataHash.不一致"
$status-code /= "assertion.hashedURI.mismatch"
$status-code /= "assertion.inaccessible"
$status-code /= "assertion.ingredient.malformed"
```

```
$status-code /= "assertion.json.invalid"
$status-code /= "assertion.missing"
$status-code /= "assertion.multipleHardBindings"
$status-code /= "assertion.notRedacted"
$status-code /= "assertion.outsideManifest"
$status-code /= "assertion.selfRedacted"
$status-code /= "assertion.undeclared"
$status-code /= "claim.cbor.invalid"
$status-code /= "claim.hardBindings.missing"
$status-code /= "claim.malformed"
$status-code /= "claim.missing"
$status-code /= "claim.multiple"
$status-code /= "claimSignature.missing"
$status-code /= "claimSignature.mismatch"
$status-code /= "claimSignature.outsideValidity"
$status-code /= "general.error" ; 他のいずれにも該当しない場合
$status-code /= "hashedURI.missing"
$status-code /= "hashedURI.mismatch"
$status-code /= "ingredient.claimSignature.missing"
$status-code /= "ingredient.claimSignature.mismatch"
$status-code /= "ingredient.manifest.missing"
$status-code /= "ingredient.manifest.mismatch"
$status-code /= "manifest.compressed.invalid"
$status-code /= "manifest.アクセス不可"
$status-code /= "manifest.multipleParents"
$status-code /= "manifest.timestamp.invalid"
$status-code /= "manifest.timestamp.wrongParents"
$status-code /= "manifest.update.invalid"
$status-code /= "manifest.update.wrongParents"
$status-code /= "署名資格情報が無効"
$status-code /= "署名認証情報.OCSP.失効済み"
$status-code /= "署名認証情報.OCSP.不明"
$status-code /= "署名認証情報.信頼できない"

; カスタムステータスコード
$status-code /= tstr .regexp "([\\da-zA-Z_-]+\\.)+[\\da-zA-Z_-]+"

status-map = {
   "code": $status-code,                  ; ステータスを説明するラベル形式の文字列
   ? "url": jumbf-uri-type,               ; このステータスコードが適用されるJUMBFボックスへのJUMBF URI参照
   ? "explanation": tstr .size (1..max-tstr-length), ; ステータスを説明する人間が読める文字列
   ? "success": bool                      ; 非推奨。コードは成功（true）または失敗（false）を反映していますか
}

status-codes-map = {
     "success": [* $status-map],                    ; 検証成功コードの配列。空の場合あり。
     "informational": [* $status-map],              ; 検証情報コードの配列。空の場合あり。
     "failure": [* $status-map]                     ; 検証失敗コードの配列。空の場合あり。
}

; マニフェストとその成分の検証結果を含むオブジェクト validation-results-map = {
     ? "activeManifest": $status-codes-map,; 構成要素のアクティブなマニフェストに対する検証ステータスコード。構成要素がC2PA資産の場合は存在し、そうでない場合は存在しない。
     ? "ingredientDeltas": [* $ingredient-delta-validation-result-map] ; 各原料のマニフェストについて、現在の検証結果と前回の検証結果との間の変更点/差分のリスト。
```

```
変更点/差分の一覧。当該成分がC2PAアセットである場合に存在。
}

ingredient-delta-validation-result-map = {
     "ingredientAssertionURI": jumbf-uri-type, ; 成分アサーションへのJUMBF URI参照
     "validationDeltas": $status-codes-map                    ; 成分のアクティブなマニフェストに対する検証結果
}
```

15.2.2. 標準ステータスコード

15.2.2.1. 成功コード

表 2. 検証成功コード

値	意味	`url` 使用
`assertion.accessible`	検証時点で非埋め込み（リモート）アサーションがアクセス可能であった。	C2PA アサーション
`assertion.bmffHash.match`	ボックスベースの資産のハッシュが、BMFF ハッシュアサーションで宣言されたハッシュと一致します。	C2PA アサーション
`assertion.boxesHash.match`	ボックスベースの資産のハッシュが、一般的なボックスハッシュアサーションで宣言されたハッシュと一致します。	C2PA アサーション
`assertion.collectionHash.match`	コレクションに含まれるすべてのアセットのハッシュが、コレクションデータハッシュアサーションで宣言されたハッシュと一致します。	C2PA アサーション
`assertion.dataHash.match`	アセットのバイト範囲のハッシュが、データハッシュアサーションで宣言されたハッシュと一致します。	C2PA アサーション
`assertion.hashedURI.match`	参照されたアサーションのハッシュが、クレーム内のアサーションのハッシュ付き URI に対応するハッシュと一致すること。	C2PA アサーション
`assertion.multiAssetHash.match`	マルチアセットハッシュアサーションの一部ハッシュが、アサーションのマルチアセットハッシュマップ内の対応するハッシュと一致する。	C2PA アサーション
`claimSignature.insideValidity`	クレームで参照されるクレーム署名は、署名認証情報の有効期間内に作成されました	C2PA クレーム署名ボックス

claimSignature.validated	クレームで参照されているクレーム署名が検証済みです。	C2PA クレーム署名ボックス
ingredient.claimSignature.validated	成分の C2PA クレーム署名ボックスのハッシュが正常に検証されました。	C2PA アサーション

値	意味	url 使用法
ingredient.manifest.validated	成分の C2PA マニフェスト ボックスのハッシュが正常に検証されました。	C2PA アサーション
署名認証情報.OCSP.非失効 OK	署名時点において署名認証情報は失効していませんでした。	C2PA クレーム署名ボックス
signingCredential.trusted	署名認証情報は信頼されています	C2PA クレーム署名ボックス
タイムスタンプが信頼されている	タイムスタンプ認証情報は、検証者のタイムスタンプ機関向け信頼アンカーリストに記載されています。	C2PAクレーム署名ボックス
timeStamp.validated	タイムスタンプは適切に形成されており、クレーム署名と一致するメッセージインプリントを持ち、タイムスタンプ認証情報の有効期間内に作成されました。	C2PA クレーム署名ボックス

15.2.2.2. 情報コード

表3. 検証情報コード

値	意味	url 使用法
algorithm.deprecated	アルゴリズムは非推奨となりました。	C2PA クレームボックスまたは C2PA アサーション
ingredient.unknownProvenance	この成分には C2PA マニフェストがありません。	C2PA アサーション
signingCredential.ocsp.inaccessible	バリデータはオンライン OCSP チェックを実行しようとしましたが、応答を受け取りませんでした。	C2PA クレーム署名ボックス
signingCredential.ocsp.スキップ	検証者はオンライン OCSP チェックを実行しないことを選択しました。	C2PA クレーム署名ボックス
timeOfSigning.insideValidity	署名時の主張時刻（署名のIATヘッダー内）は、署名者の証明書チェーンの有効期間内にあり、かつ対応する信頼できるタイムスタンプのいずれかの時刻よりも前である。	C2PA クレーム署名ボックス

timeOfSigning.outsideValidity	署名（署名の iat ヘッダー内）の主張された時刻は、署名者の証明書チェーンの有効期間外、または対応する信頼できるタイムスタンプの時刻よりも遅い。	C2PA クレーム署名ボックス
timeStamp.malformed	クレーム署名ヘッダーに含まれるタイムスタンプ応答が、RFC 3161 に準拠した適切な形式で構成されていません。	C2PA クレーム署名ボックス
値	**意味**	**url 使用法**
timeStamp.mismatch	タイムスタンプがクレームの内容と一致しません。	C2PA クレーム署名ボックス
timeStamp.outsideValidity	署名内の署名済みタイムスタンプ属性が、TSAの証明書の有効期間外に作成されました。	C2PA クレーム署名ボックス
timeStamp.untrusted	タイムスタンプ認証情報は、バリデータ側のTSA信頼リストに記載されていません。	C2PAクレーム署名ボックス

15.2.2.3. 失敗コード

表4. 検証失敗コード

値	意味	url 使用法
algorithm.unsupported	アルゴリズムが未指定またはサポートされていません。	C2PA クレームボックスまたは C2PA アサーション
assertion.action.ingredientMatch	関連する成分を必要とするアクションが、その成分を持たないか、指定された成分が見つかりません	C2PA アサーション
assertion.action.malformed	アクションのアサーションが不正な形式です。	C2PA アサーション
assertion.action.redacted	クレーム作成時にアクションアサーションが編集されました。	C2PA アサーション
assertion.action.redactionMis match	関連する編集が必要なアクションが、編集を伴っていないか、指定された編集が見つかりません	C2PA アサーション
assertion.bmffHash.malformed	BMFF ハッシュアサーションの形式が不正です。	C2PA アサーション
assertion.bmffHash.mismatch	ボックスベースの資産のハッシュが、BMFFハッシュアサーションで宣言されたハッシュと一致しません。	C2PA アサーション
assertion.boxesHash.malformed	一般的なボックスハッシュアサーションの形式が不正です	C2PA アサーション

	。	
assertion.boxesHash.mismatch	一般的なボックス型資産フォーマットのハッシュが、一般的なボックスハッシュアサーションで宣言されたハッシュと一致しません。	C2PA アサーション
assertion.boxesHash.unknownBox	予想外のボックスが見つかりました	C2PA アサーション
assertion.cloud-data.actions	更新マニフェストに、アクションアサーションを参照するクラウドデータアサーションが含まれています。	C2PA アサーション
assertion.cloud-data.hardBinding	ハードバインディングアサーションは、クラウドデータアサーション内にあります。	C2PA アサーション
値	**意味**	`url` **使用法**
assertion.cloud-data.malformed	クラウドデータアサーションが不完全でした	C2PA アサーション
assertion.cbor.invalid	アサーションのcborが無効です	C2PA アサーション
assertion.collectionHash.incorrectFileCount	コレクションデータハッシュアサーションにリストされていたアセットが、コレクションから欠落しています。	C2PA アサーション
assertion.collectionHash.invalidURI	コレクションデータハッシュアサーション内のアセットの URI に、ファイル部分「..」または「.」が含まれています。	C2PA アサーション
assertion.collectionHash.malformed	コレクションハッシュアサーションが不完全でした	C2PA アサーション
assertion.collectionHash.mismatch	コレクション内のアセットのハッシュが、コレクションデータハッシュアサーションで宣言されたハッシュと一致しません。	C2PA アサーション
assertion.dataHash.malformed	データハッシュアサーションが不正な形式です。	C2PA アサーション
assertion.dataHash.mismatch	アセットのバイト範囲のハッシュが、データハッシュアサーションで宣言されたハッシュと一致しません。	C2PA アサーション
assertion.dataHash.redacted	クレーム作成時にハードバインディングアサーションが編集されました。	C2PA アサーション
assertion.hashedURI.mismatch	マニフェストで参照されているアサーションのハッシュが、クレーム内のアサーションのハッシュ付き URI に対応するハッシュと一致しません。	C2PA アサーション
assertion.inaccessible	検証時に、埋め込まれていない（リモート）アサーションにアクセスできませんでした。	C2PA アサーション

`assertion.ingredient.malformed`	成分アサーションが不完全でした	C2PA アサーション
`assertion.json.invalid`	アサーションの JSON(-LD) が無効です	C2PA アサーション
`assertion.missing`	マニフェストのクレームにリストされているアサーションが、アセットのマニフェストから欠落しています。	C2PA クレームボックス
`assertion.multiAssetHash.malformed`	マルチアセットハッシュアサーションの形式が不正です。	C2PA アサーション
`assertion.multiAssetHash.miss ingPart`	マルチパートアセットの必須部分が検出できません。	C2PA アサーション
`assertion.multiAssetHash.mismatch`	マルチパーツ資産の一部ハッシュが、マルチアセットハッシュアサーションで宣言されたハッシュと一致しません。	C2PA アサーション
`assertion.multipleHardBinding s`	マニフェストに複数のハードバインディングアサーションが含まれています。	C2PA アサーションストアボックス
値	**意味**	`url` **使用法**
`assertion.notRedacted`	クレームではアサーションが編集済みと宣言されているが、マニフェストには依然として存在する。	C2PA アサーション
`assertion.outsideManifest`	クレームに記載されているアサーションが、そのクレームと同じ C2PA マニフェストに存在しません。	C2PA クレームボックス
`assertion.selfRedacted`	アサーションは、それ自身のクレームによって編集済みとして宣言されました。	C2PA クレームボックス
`assertion.timestamp.malformed`	タイムスタンプアサーションの形式が不正です。	C2PA アサーション
`assertion.undeclared`	マニフェスト内で、クレームに明示的に宣言されていないアサーションが検出されました。	C2PA アサーション
`claim.cbor.invalid`	クレームのcborが無効です。	C2PA クレームボックス
`claim.hardBindings.missing`	ハードバインディングが存在しません。	C2PA クレームボックス
`claim.malformed`	マニフェストで参照されているクレームのデータ/フィールドが正しくありません。	C2PA クレームボックス
`claim.missing`	マニフェストで参照されているクレームが見つかりません。	C2PA クレームボックス
`claim.multiple`	マニフェストに複数のクレームボックスが存在します。	C2PA クレームボックス
`claimSignature.missing`	クレームで参照されているクレーム署名が、そのマニフェスト内で見つかりません。	C2PA クレーム署名ボックス

claimSignature.mismatch	クレームで参照されているクレーム署名の検証に失敗しました。	C2PA クレーム署名ボックス
claimSignature.outsideValidit y	クレームで参照されているクレーム署名は、署名認証情報の有効期間外で作成されました。	C2PA クレーム署名ボックス
general.error	ここに具体的に記載されていないエラーが発生した場合に使用する値。	C2PA クレームボックスまたは C2PA アサーション
hashedURI.missing	hashed_uri が指すデータが見つかりません	C2PA アサーション
hashedURI.mismatch	指定された hashed_uri のハッシュが、宛先 URI のデータの対応するハッシュと一致しません	C2PA アサーション
ingredient.claimSignature.mis sing	参照された ingredient C2PA クレーム署名が見つかりませんでした。	C2PA アサーション
値	**意味**	**url 使用法**
ingredient.claimSignature.mis match	埋め込まれた C2PA マニフェストの C2PA クレーム署名のハッシュが、成分アサーションの claimSignature フィールドの hashed_uri 値で宣言されたハッシュと一致しません。	C2PA アサーション
ingredient.manifest.missing	参照された ingredient C2PA マニフェストが見つかりませんでした。	C2PA アサーション
ingredient.manifest.mismatch	埋め込まれた C2PA マニフェストのハッシュが、成分アサーションの activeManifest フィールドの hashed_uri 値で宣言されたハッシュと一致しません。	C2PA アサーション
manifest.compressed.invalid	圧縮されたマニフェストが無効でした。	C2PA クレームボックス
manifest.inaccessible	検証時に、非埋め込み（リモート）マニフェストにアクセスできませんでした。	C2PA クレームボックス
manifest.multipleParents	マニフェストには、parentOf 関係にある複数の成分が含まれています。	C2PA クレームボックス
manifest.timestamp.invalid	マニフェストはタイムスタンプマニフェストですが、許可されていない（非成分）アサーションが含まれています	C2PA クレームボックス

	。	
manifest.timestamp.wrongParen ts	マニフェストはタイムスタンプマニフェストですが、parentOf 成分が 0 個または複数含まれています。	C2PA クレームボックス
manifest.update.invalid	マニフェストは更新マニフェストですが、ハードバインディングやアクションアサーションなど、許可されていないアサーションが含まれています。	C2PA クレームボックス
manifest.update.wrongParents	マニフェストは更新マニフェストですが、parentOf 要素が 0 個、または複数含まれています。	C2PA クレームボックス
signingCredential.invalid	署名認証情報は署名に無効です。	C2PA クレーム署名ボックス
signingCredential.ocsp.revoked	OCSP 応答により、署名認証情報が発行者によって失効されていることが示されています。	C2PA クレーム署名ボックス
signingCredential.ocsp.unknown	OCSP 応答には、署名認証情報のステータスが含まれています	C2PAクレーム署名ボックス
signingCredential.untrusted	署名認証情報は、検証者の適用可能な信頼リストのいずれにも記載されていません。	C2PAクレーム署名ボックス

15.3. マニフェスト情報の表示

マニフェストコンシューマーは、有効でないマニフェストまたは有効でないアセットからのデータを表示してはなりません。マニフェストコンシューマーがそのようなデータを表示することを選択した場合、表示の一部として以下を含める必要があります：

- 有効性欠如に関する警告、
- 当該データがマニフェスト署名者に帰属しない旨の警告、および原材料マニフェストの場合は追加で、当該資産のマニフェスト署名者にも帰属しない旨の警告。

注意

シナリオ作成においては、作成者が判断できるよう警告をより目立つように表示することが望ましい。無効なアセットや欠陥のあるプロバンス履歴を持つアセットの処理方法について、十分な情報に基づいた判断を下せるようにすることである。

15.4. ハッシュアルゴリズムの決定

15.4.1. ハッシュ付きURIの場合

C2PAマニフェストの様々な部分では、URI、そのハッシュ、および（オプションで）ハッシュ計算に使用されるアルゴリズムをカプセル化するためにhashed_uri構造が利用される。hashed_uri構造内に`alg`フィールドが存在する場合、それはハッシュアルゴリズムとして使用される。`alg`フィールドがhashed_uri構造内に存在しない場合、ハッシュアルゴリズムはalgフィールドを含む最も近い囲み構造を評価することで決定される。これらのいずれの場所にも `alg` フィールドが見つからない場合、クレーム内の `alg` フィールドの値をハッシュアルゴリズムとして使用します。これらのいずれの場所にも `alg` フィールドが存在しない場合、クレームは `algorithm.unsupported` の失敗コードで拒否されます。

15.4.2. ハッシュ付き外部URIの場合

C2PAマニフェストの一部では、外部URI、そのハッシュ、およびハッシュ計算に使用されたアルゴリズムをカプセル化するためにhashed_ext_uri構造体が利用される。hashed_ext_uri構造体にalgフィールドが存在する場合、それをハッシュアルゴリズムとして使用しなければならない。algフィールドがhashed_ext_uri構造体に存在しない場合、失敗コード`algorithm.unsupported`を使用しなければならない。

注記

`alg`フィールドは`hashed_ext_uri`において必須であるため、ハッシュアルゴリズムを決定するための再帰的な手順は不要である。

15.4.3. アルゴリズムの検証

ハッシュアルゴリズムが決定されたら、セクション13.1「ハッシュ」の許可リストまたは非推奨リスト内の値と比較する。いずれのリストにも存在しない場合、クレームは失敗コード`algorithm.unsupported`で拒否される。アルゴリズムが非推奨リストに含まれる場合、クレームには情報コード`algorithm.deprecated`が発行される。

15.5. アクティブマニフェストの特定

15.5.1. 一般事項

C2PAマニフェストストアスーパーボックス内の最後のC2PAマニフェストスーパーボックスをアクティブマニフェストと見なす。ただし、C2PAマニフェストストアの特定には複数の可能性のある場所を調査する必要がある場合がある。

15.5.2. 埋め込み

15.5.2.1. 一般

C2PAマニフェストストアは、マニフェスト埋め込みの標準的な場所において、アセット内に埋め込まれたバリデータによって特定されるものとする。ただし、アセットがHTTP接続を介して取得された場合、バリデータは下記「リンクヘッダー条項」に記載されるリンクヘッダーを参照し、C2PAマニフェストストアが存在するか否かを判断することができる。

NOTE

リンクヘッダーが存在する場合、それを確認することでバリデータはC2PAマニフェストストアが存在するか否かを、アセット全体をダウンロードせずに判断できます。これは、大容量のアセットやストリーミングされるアセットにおいて有用です。

アセット内に複数のC2PAマニフェストストアが存在する場合、それら全てを無効と見なし、検証ではマニフェストが見つからなかった場合と同様に扱う必要があります。このアセットがイングリディエントとして追加される場合、埋め込まれたC2PAマニフェストはいずれもイングリディエントアサーションに含まれてはなりません。

15.5.2.2. PDFに関する特別な考慮事項

PDFファイルは「増分更新」と呼ばれる技術をサポートしており、これにより情報は元の文書を変更する代わりに文書の末尾に追加されます。このため、PDFファイルは複数のC2PAマニフェストストアをサポートする必要があります（ただし、更新セクションごとに1つだけ存在すること）。

単一の更新セクション内に複数のC2PAマニフェストストアが存在する場合、それらはすべて無効と見なされ、検証ではマニフェストが見つからなかった場合と同様に扱う必要があります。ただし、PDFの初期更新時または元のPDFに存在するC2PAマニフェストストアは、依然として有効と見なされ、それに応じて処理されます。

15.5.3. 参照またはURIによる

15.5.3.1. 参照による方法

埋め込みC2PAマニフェストストアが存在しない場合、リモート位置でマニフェストストアを特定するために以下の試行を行うべきである。

- アセットが HTTP 接続を介して取得された場合、以下の「リンクヘッダー」項で、リンクヘッダーを介してマニフェストを見つける方法について説明します。

- アセットが標準アセット位置（つまりC2PAマニフェスト外）にXMPを有し、かつそのXMPに`dcterms:provenance`キーが含まれている場合、提供されたURIを使用してアクティブなマニフェストを検索する必要があります。

- アセットがC2PAテーブルを持つフォントであり、そのactiveManifestUriLengthがゼロでない場合、指定されたURIを使用してアクティブなマニフェストを検索する必要があります。
- C2PAマニフェストストアが見つからない場合、バリデータは同じパスまたはURIで、ファイル名拡張子が.c2paのファイルを検索すべきです。C2PAマニフェストストアが見つからない場合、バリデータはそれを特定するために最も適切と判断する追加の場所（例：ファイルシステムの子フォルダ）を検索できます。

> **注記**
> バリデータは上記の場所に限定されず、追加の場所も選択して検索できます。

マニフェストがリモート場所に存在すると文書化されていたにもかかわらず、そこに存在しない場合、またはその場所が現在利用できない場合（オフラインシナリオなど）、この状況を報告するために `manifest.inaccessible` エラーコードを使用しなければならない。

C2PAマニフェストストアのIANAメディアタイプに関する情報は、外部マニフェストセクションに記載されています。

15.5.3.2. リンクヘッダーによる方法

アセットがHTTP接続経由で取得された場合、バリデータはRFC 8288で定義されるLinkヘッダーをHTTPレスポンスのヘッダー内で検索し、rel=c2pa-manifestパラメータを含む必要があります。存在する場合、そのURI参照からC2PAマニフェストストアを取得できます。URIは標準的なhttpまたはhttps URI（例：https://c2pa.org/image.c2pa）となります。

また、JUMBF URIフラグメントを使用して、アセット内に埋め込まれたC2PAマニフェストストアを参照するためにlink関係を使用することも可能です。この場合、URIにはC2PAマニフェストストアのスーパーボックスhttps://c2pa.org/image.jpg#jumbf=c2paへのJUMBF URIフラグメントが含まれます。C2PAマニフェストストア内の特定のC2PAマニフェストへの参照は許可されておらず、バリデータはJUMBF URIフラグメントの子ラベル部分を無視するものとします。

> **注記**
> HTTP は RFC 7230 で定義された*ハイパーテキスト転送プロトコル*を指し、特定の URL スキーム `http://` を指すものではない。

15.5.4. 解凍

前述のように、標準マニフェストと更新マニフェストの両方を圧縮することができます。圧縮されたマニフェストが検出された場合、バリデータは標準の検証プロセスを続行する前にそれを解凍しなければなりません。圧縮されたマニフェストのブロブボックスに含まれるデータが標準マニフェストでも更新マニフェストでもなかった場合、または解凍に失敗した場合、バリデータは失敗コード `manifest.compressed.invalid` でマニフェストを拒否しなければなりません。

15.5.5. 一致の検証

バリデータは、検出されたC2PAマニフェストストアが実際に当該アセットに関連付けられたものであることを検証したい場合があります。

C2PAマニフェストストアが検出された場合、そのアクティブマニフェスト内のハードバインディングアサーションを用いて、それが一致するマニフェストであること、およびマニフェスト更新なしにアセットが変更されたかどうかを検証する。ハードバインディングが一致しない場合、その原因が(a)アセットの変更か、(b)誤ったC2PA

マニフェストストアが検出されたためかは不明である。したがって、バリデータはこれを不一致のハードバインディングとして扱い、データハッシュアサーションが使用されている場合はassertion.dataHash.mismatch、一般的なボックスハッシュアサーションが使用されている場合は assertion.boxesHash.mismatch 、コレクションデータハッシュアサーションが使用されている場合は assertion.collectionHash.mismatch、BMFFハッシュアサーションが使用されている場合はassertion.bmffHash.mismatchの失敗コードでマニフェストを拒否する。

15.6. クレームの特定と検証

15.6.1. クレームの特定

検証対象のマニフェスト（以下「現在のマニフェスト」と呼ぶ）を特定した後、現在のマニフェスト内でラベルが c2pa.claim.v2（または古いクレーム構造を持つファイルの場合は c2pa.claim）で、JUMBF タイプ UUID が 6332636C-0011-0010-8000-00AA00389B71（c2cl）のラベルを持つJUMBFスーパーボックスを検索することでクレームを特定する。なお、JUMBFタイプUUIDは、新しい形式（c2pa.claim.v2ラベル付き）と古い形式（c2pa.claimラベル付き）の両方で同一であることに留意すること。現在のマニフェストには、このようなボックスは1つだけ存在しなければなりません。複数見つかった場合、C2PAマニフェストは失敗コードclaim.multipleで拒否されます。

15.6.2. 検証

クレームの内容が適切に構成されたCBORでない場合、当該クレームは失敗コード
claim.cbor.invalid

> **注記** 適切に構成されたCBORはRFC 8949付録Cで定義されている。

「c2pa.claim.v2」の場合、CBORオブジェクトには以下のフィールドが存在することが期待されます。いずれかが欠落している場合、クレームは失敗コードclaim.malformedで拒否されます。

- instanceID
- signature
- created_assertions
- claim_generator_info

claim_generator_info フィールドに name フィールドが含まれていない場合、クレームは失敗コード claim.malformed で拒否される。

claim-map または claim-map-v2 の claim_generator_info フィールドが参照する generator-info-map に icon フィールドが存在する場合、その値はセクション 15.10.3.3「参照の検証」に記載されている方法で検証されなければならない。

15.7. 署名の検証

クレームの署名フィールドの値から署名の URI 参照を取得し、その URI を解決する

COSE署名の取得に関する参照。署名は、セクション11.1.4「C2PAボックスの詳細」で説明されているのと同じマニフェストに埋め込まれるものとする。署名URIが同一のC2PAマニフェストボックス内（self#jumbfロケーション）を指していない場合、クレームは拒否される。そのようなフィールドが存在しない場合、またはURIが解決できない場合、クレームは失敗コード`claimSignature.missing`で拒否される。

署名とクレームが同一のC2PAマニフェストに含まれていない場合、当該C2PAマニフェストは有効とはみなされない。

すべての種類のC2PAマニフェストにおいて、署名に使用された認証情報の検証は、第14章「*信頼モデル*」に従って実施されるものとする。

クレデンシャルが当該クレデンシャルのタイプ要件を満たさない場合、署名クレデンシャル無効（`signingCredential.invalid`）の失敗コードでクレームを拒否する。署名アルゴリズムが第13.2節「デジタル署名」の許可リストまたは非推奨リストに含まれない場合、アルゴリズム非対応（`algorithm.unsupported`）の失敗コードでクレームを拒否する。

その後、当該認証情報から適用可能な信頼アンカーリストのいずれかのエントリに至る信頼の連鎖を検証する必要がある。この信頼の連鎖を検証できない場合、クレームは失敗コード `signingCredential.untrusted` で拒否される。それ以外の場合は、クレーム署名に成功コード `signingCredential.trusted` が割り当てられる。

クレームがまだ拒否されていない場合、第13.2節「デジタル署名」で規定された手順に従い検証を継続する。署名の検証に失敗した場合、クレームは失敗コード `claimSignature.mismatch` で拒否される。それ以外の場合は、クレーム署名に成功コード `claimSignature.validated` が割り当てられる。

本章の残りの部分において、ヘッダーとはCOSE署名における保護対象および非保護対象ヘッダーパラメータの集合の和集合を指す。第13.2節「デジタル署名」または第14.5節「X.509証明書」で特に指定されていない限り、ヘッダーはどちらのバケットにも出現し得る。COSEヘッダーはRFC 8152の第3節で規定されている。

15.8. タイムスタンプの検証

15.8.1. タイムスタンプトークンの取得

15.8.1.1. クレーム署名への埋め込み

`sigTst` または `sigTst2` ヘッダーのいずれかが存在する場合、`tstTokens` 配列には単一の `tstToken` が含まれていることが期待される。ヘッダーに複数の `tstToken` が含まれる場合、検証者は `timestamp.malformed` 情報コードを発行し、タイムスタンプを無視しなければならない。

`sigTst`をサポートするバリデータは、タイムスタンプ応答を検証するために以下の手順を実行する：

- `tstToken`から`val`プロパティを取得する。これはRFC3161準拠の`TimeStampResp`（タイムスタンプ応答）でなければならない。

- statusフィールドの値を確認する。PKIStatusInfoは、TimeStampRespのstatusフィールドの値である。
    - 0（承認済み）または1（修正付き承認済み）以外の値が含まれている場合、バリデータはタイムスタンプが不正な形式の場合、情報コードを発行し、そのタイムスタンプを無視する。
    - 値が 0 (承認済み) または 1 (条件付き承認済み) の場合は、以下に説明するタイムスタンプ検証プロセスの残りの処理を続行する。
- 検証プロセスの残りの部分で使用するために、TimeStampResp の timeStampToken フィールドの値を取得する。

sigTst2 のバリデータは、tstToken から val プロパティを取得する。これは RFC3161 準拠のタイムスタンプトークン（TimeStampToken、TST）である。

15.8.1.2. タイムスタンプアサーションによって参照される

バリデータが既にsigTstまたはsigTst2ヘッダー内でタイムスタンプトークンを検出し、かつその検証に合格した場合（セクション15.8.2「タイムスタンプトークンの検証」に従う）、このステップをスキップする。該当するヘッダーが存在しない場合、またはそこに検出されたタイムスタンプトークンの検証に合格しなかった場合、このステップを実行する。

検証者が以前にタイムスタンプアサーションを検出し、それをC2PAマニフェスト識別子とタイムスタンプトークンの対応付けに保持していた場合、検証者は現在のC2PAマニフェストの識別子がその対応付けに含まれているかを確認する。存在する場合、バリデータはセクション15.8.2「タイムスタンプトークンの検証」で説明される検証プロセスにおいて、マッピング内の識別子に関連付けられたタイムスタンプトークンを使用する。マッピング内で当該識別子に対応する複数のタイムスタンプトークンが見つかった場合、バリデータは検証に成功するまで各トークンを試行する（成功したトークン以外のものは無視する）。識別子がマッピングに存在しない場合、エラーは発生しない。これは単に、現在のコンテキストにおいてこのC2PAマニフェストに関連付けられたタイムスタンプトークンが存在しないことを意味する。

15.8.2. タイムスタンプトークンの検証

すべてのバリデータは、以下の手順に従って処理を継続しなければならない：

- タイムスタンプトークン内の署名アルゴリズムが、セクション13.2「デジタル署名」の許可リストまたは非推奨リストに含まれていない場合、検証者はタイムスタンプ.untrusted 情報コードを発行し、当該タイムスタンプを無視しなければならない。
- timeStampToken内の署名をRFC 2630セクション5.6に従って検証する。署名が有効でない場合、バリデータはtimestamp.mismatch情報コードを発行しタイムスタンプを無視する。
- timeStampToken does not contain a messageImprint field, the validator shall issue a timestamp.malformed 情報コードを発行し、タイムスタンプを無視する。
- メッセージインプリントハッシュアルゴリズムがセクション 13.1「ハッシュ」の許可リストまたは非推奨リストに含まれていない場合、バリデータは timestamp.untrusted 情報コードを発行し、タイムスタンプを無視する。
- メッセージインプリントフィールド（タイムスタンプトークン内）の値が、クレーム（v1、

検証対象のC2PAマニフェストのsigTstフィールド（v2）またはCOSE_Sign1_Tagged構造体の署名フィールド（v2、sigTst2）を、セクション10.3.2.5.2「ペイロードの選択」に記載の通り照合する。値が一致しない場合、検証者はtimestamp.mismatch情報コードを発行し、タイムスタンプを無視する。

- タイムスタンプトークンの証明書フィールドが存在すること、TSAの証明書が本フィールドで提供された証明書セット内に存在すること、およびTSAの証明書からC2PA TSA信頼リスト（またはこの目的のために検証器に存在するその他の信頼アンカーリスト）のエントリへの信頼チェーンを構築できることを検証する。証明書が見つからない場合、または信頼チェーンを構築できない場合、バリデータはtimestamp.untrusted情報コードを発行し、タイムスタンプを無視する。
- genTimeフィールド（timeStampToken内）に記載された認証時刻が、TSAの署名証明書および信頼アンカーまでの全CA証明書の有効期間内に収まっていることを検証する。該当しない場合、検証者はtimestamp.outsideValidity情報コードを発行し、タイムスタンプを無視する。
- 上記のいずれかの条件によりタイムスタンプ検証が停止または失敗しない場合、バリデータは成功コード timeStamp.trusted および timeStamp.validated を発行する。
- 検証者が timeStamp.trusted と timeStamp.validated の両方の成功コードを発行した場合、検証者はタイムスタンプ機関（TSA）によって証明された時刻（timeStampToken 内の genTime フィールドに記載）が、クレーム署名証明書および信頼アンカーまでのすべての CA 証明書の有効期間内にあることを検証しなければならない。そうでない場合、バリデータは claimSignature.outsideValidity失敗コードでクレームを拒否しなければならない。

注記

タイムスタンプは、タイムスタンプ機関の署名認証情報が失効した後も有効であり続けるため、証明された時刻がタイムスタンプ機関の証明書の有効期間内に収まる限り、有効です。これはタイムスタンプ機関にのみ適用される特別な信頼形態です。

検証時、タイムスタンプが存在し、信頼され、かつ検証された場合、検証者は署名証明書の有効期間およびタイムスタンプ機関の証明書の有効期間を判断する際、現在の時刻ではなく証明された時刻を使用しなければならない。

注記

本仕様は、タイムスタンプ機関の証明書の失効状態を署名時に取得すること、または検証時に確認することを要求しない。

sigTstヘッダーもsigTst2ヘッダーも存在しない場合、または少なくとも一方が存在してもそのタイムスタンプトークンが上記の要件を満たさない場合、検証時点の時刻が署名者の証明書および信頼アンカーまでの全てのCA証明書の有効期間内にある場合に限り、C2PAマニフェストは有効である。この場合、検証者は成功コード claimSignature.insideValidity を返す。そうでない場合、C2PAマニフェストは失敗コード claimSignature.outsideValidity で拒否される。

15.8.3. 「署名時刻の主張」の検証

検証者は、IAT保護ヘッダー内の値によって証明される「署名時刻」の検証を選択できる。IATヘッダーが存在する場合、検証者は証明時刻が署名者の証明書および信頼アンカーまでの全CA証明書の有効期間内にあり、関連する信頼できるタイムスタンプによって証明される時刻より遅くないことを検証できる。検証者がこの値の検証を行い、それが有効期間内に収まる場合、

検証者は `timeOfSigning.insideValidity` 情報コードを返す。しかし有効期間外の場合、検証者は `timeOfSigning.outsideValidity` 情報コードを返す。

15.9. 認証情報の失効情報を検証する

検証者は、署名者の証明書および信頼チェーンを構成するすべてのCA証明書の失効状態を確認しようとする。

CA証明書については、RFC 5280セクション4.2.2.1で規定される権限情報アクセス(AIA)拡張に示される失効状態を判定する。AIA拡張がOCSPの利用可能性を示す場合、検証者はC2PAマニフェストに含まれる関連OCSP応答を活用すべきである。

検証者が、信頼されたタイムスタンプで示された時点、または信頼されたタイムスタンプが存在しない場合は現在の時点で、CA証明書が失効していると判断した場合、署名クレームは失敗ステータス`signingCredential.untrusted`で拒否される。

署名者の証明書については、バリデータは次のプロセスを使用する。

- 証明書が失効ステータスをサポートしていない場合、または証明書発行者がその失効ステータスを照会する方法を指定していない場合、バリデータは、その認証情報を失効していないものとして扱う。
- クレーム生成者がCOSE_Sign1構造体のrValsヘッダーにOCSP応答を「ステープル」した場合、バリデータはセクション15.9.1「C2PAマニフェストストア内のOCSP応答による失効判定」に記載の方法でステープルされたOCSP応答をデコードし検証しなければならない。
- 後続のクレーム生成者がC2PAマニフェストストア内の他のC2PAマニフェストに証明書ステータスアサーションを追加した場合、バリデータはセクション15.9.1「C2PAマニフェストストア内のOCSP応答による失効判定」で説明される検証プロセスにおいてそれらのOCSP応答を使用しなければならない。証明書に対して複数のOCSP応答が見つかった場合、バリデータは検証に成功するまで各応答を試行し（その後、他の応答は無視すべきである）。

C2PAマニフェストストアに失効情報が存在せず、検証者がオンライン状態で、かつ当該証明書の失効ステータスを検証したい場合、検証者はセクション15.9.2「オンラインOCSP応答による失効判定」に記載の方法でOCSPレスポンダを照会し、証明書の失効ステータスを判定しようとするものとする。

15.9.1. C2PAマニフェストストアにおけるOCSP応答による失効判定

検証者は、RFC 6960 の要件、特にセクション 3.2 の要件 1 から 4 に基づいて OCSP 応答をデコードしなければならない。OCSP 応答が受け入れられ、かつ以下のすべての要件が満たされている場合、関連する証明書は署名時点で失効していなかったことが確立される。

- 署名付きタイムスタンプによる認証済み時刻が署名に付与されている。
- OCSP応答のtbsResponseDataフィールドの応答配列内に、以下のすべての条件を満たすSingleResponseが存在します：

- 現在の時刻が `thisUpdate` より古くないこと。
- タイムスタンプの認証時刻が、
  - `thisUpdate`より早い、または
  - nextUpdateが存在する場合、`(thisUpdate,nextUpdate)`区間内に収まる、または
  - nextUpdateが存在しない場合、`producedAt`が対応する`ResponseData`フィールドの値である場合、`(thisUpdate,producedAt + 24時間)`の間隔内に収まる。
- `SingleResponse`の`certStatus`フィールドが有効であるか、失効しているが失効理由が `removeFromCRL` である場合。

> **注記**
> `removeFromCRL` は `revocationReason` の値の中で唯一、正常な応答と同等である点が特徴である。失効応答の一種であるにもかかわらず、この応答は
> は、証明書の完全性に関する懸念により一時的に「保留」状態（certificateHold理由）に置かれていたが、その懸念が解消され、発行者が証明書が信頼できる状態を維持していると表明していることを示します（RFC 5280参照）。

- 応答の OCSP 署名者は、RFC 6960 セクション 4.2.2.2 で定義される「認可されたレスポンダ」である。

バリデータは、無効化済み応答の `revocationReason` を確認し、無効化済み応答の `revocationReason` と実際の失効の `removedFromCRL` ケースと実際の失効を区別する。

C2PA マニフェストストア内のいずれかの OCSP 応答が上記の条件を満たす場合、証明書は署名時点で失効していないと見なされ、バリデータは `signingCredential.ocsp.notRevoked` 成功コードを発行する。

それ以外の場合、C2PA マニフェストストア内の OCSP 応答は、`certStatus` フィールドが `revoked` である点を除き、上記の条件をすべて満たすため、署名時点では証明書は失効しているとみなされ、クレームは `signingCredential.ocsp.revoked` 失敗コードで拒否される。

15.9.2. オンラインOCSP応答からの失効判定

特定の証明書について、C2PAマニフェストストア内のOCSP応答がセクション15.9.1の条件を満たさない場合、「C2PAマニフェストストア内のOCSP応答による失効判定」の条件を満たさない場合、またはクレーム署名にタイムスタンプがない場合、バリデータはRFC 6960に従い、RFC 6960セクション3.1で特定されたレスポンダの`accessLocation`を用いてOCSPレスポンダに問い合わせを行うことを選択できる。

> **注記**
> 資格証明ステータス照会メソッドは、検証対象の資産の身元を観察者に明らかにする可能性があるため、この照会はオプションである。

もし バリデータ が しない 実行しない 実行 検証 オンライン OCSP チェックを それ しなければならない 発行 a `signingCredential.ocsp.skipped` 情報コードを発行する。

バリデータがOCSPレスポンダへの問い合わせを試みたが応答を受信できなかった場合、バリデータは

`signingCredential.ocsp.inaccessible` 情報コードを発行しなければならない。

RFC 6960 3.2節の要件1〜4に従い応答を受信し受理した場合、以下のいずれかの要件を満たせば署名時点での署名者証明書の失効がなかったことを確立する：

- 以下のいずれかが満たされる場合、署名時点において署名者の証明書が失効していなかったことを確立する。 claim 署名 が a 有効な タイムスタンプが そして 証明された 証明された 時刻 は に 応答の `(thisUpdate,nextUpdate)` 間隔内に収まるか、
- クレーム署名に有効なタイムスタンプがないが、現在の現実世界の時刻が `(thisUpdate,nextUpdate)` 間隔内にあり、かつ以下の両方の要件が満

たされている場合：

- 応答の`certStatus`フィールドは良好、または失効しているが、失効理由が `removeFromCRL`」である場合、
- 応答の OCSP 署名者は、RFC 6960 のセクション 4.2.2.2 で定義されている「認可された応答者」である。

応答の `certStatus` フィールドが失効しているが、`revocationReason` が `removeFromCRL` 以外の場合、以下の両方の要件を満たす場合に限り、署名時点において署名者の証明書が失効していなかったことを確立する：

- マニフェストに有効なタイムスタンプが存在し、かつ証明された時刻が応答の `(thisUpdate,nextUpdate)` 間隔内に収まっていること、かつ
- 応答の`revocationTime`が証明されたタイムスタンプより後であること。

上記の条件が満たされている場合、証明書は署名時点で失効していないものとみなされ、バリデータは署名認証情報`.ocsp.notRevoked`の成功コードを発行する。

それ以外の場合は：

- の場合、 のcertStatusフィールド が の 応答 は 不明、 の クレーム は 拒否され、 `signingCredential.ocsp.unknown` 失敗コードで拒否される。
- それ以外の場合、証明書は署名時点で失効しているとみなされ、クレームは `signingCredential.ocsp.revoked` 失敗コードで拒否される。

15.10. アサーションの検証

15.10.1. マニフェストのタイプに対して正しいアサーションを検証する

15.10.1.1. 一般

マニフェストのタイプに応じて、必須または禁止されるアサーションが存在します。バリデータは必須および許可されないアサーションをチェックしなければなりません。

15.10.1.2. 標準マニフェストのアサーション

標準マニフェストの場合：

1. コンテンツへのハードバインディングが正確に1つ存在することを検証する - `c2pa.hash.data`、`c2pa.hash.boxes`、`c2pa.hash.collection.data`、`c2pa.hash.bmff.v2`（非推奨）、または`c2pa.hash.bmff.v3`のいずれかであること。該当するアサーションが存在しない場合、マニフェストは失敗コード `claim.hardBindings.missing` で拒否される。該当するアサーションが複数存在する場合、マニフェストは失敗コード `assertion.multipleHardBindings` で拒否される。
2. `c2pa.ingredient` アサーションで、その関係が `parentOf` であるものが 0 個または 1 個存在することを検証する。複数存在する場合、マニフェストは `manifest.multipleParents` の失敗コードで拒否される。
3. `c2pa.created` または `c2pa.opened` アクションのいずれかが、正確に 1 つの actions アサーションに含まれていることを検証する。

15.10.1.3. マニフェストのアサーションを更新する

更新マニフェストの場合：

1. 正確に1つの成分アサーションが存在し、その関係性が`parentOf`であることを検証する。これらが満たされない場合（欠落、複数存在、関係性の値が`parentOf`でない場合）、マニフェストは失敗コード`manifest.update.wrongParents`で拒否される。
2. `c2pa.hash.data`、`c2pa.hash.boxes`、`c2pa.hash.collection.data`、`c2pa.hash.bmff.v2`（非推奨）、`c2pa.hash.bmff.v3`、またはサムネイルアサーションが存在しないことを検証する。存在する場合は、マニフェストは失敗コード `manifest.update.invalid` で拒否される。
3. `c2pa.hash.multi-asset` アサーションが存在しないことを検証する。存在する場合、マニフェストは `manifest.update.invalid` の失敗コードで拒否される。
4. `c2pa.actions` または `c2pa.actions.v2` アサーションが 1 つ以上存在する場合、そのようなアサーションのいずれかの `actions` 配列内で見つかった各アクションの `action` フィールドが、更新マニフェストで指定されたサポート対象値のいずれかであることを検証する。そうでない場合、マニフェストは `manifest.update.invalid` の失敗コードで拒否される。

15.10.2. 編集済みアサーションのリストの準備

バリデータは、クレームを処理する際、その `redacted_assertions` フィールドにリストされている各 JUMBF URI に基づいて、各成分のマニフェスト（存在する場合）に対する編集済みアサーションのセットを収集する。クレームの `redacted_assertions` フィールドは、決して自身のいずれかのアサーションへの JUMBF URI を含んではならない。

注記

最終アセットの出所において、主張はいつでも成分アセットから削除される可能性がある

履歴の任意の時点で、必ずしもその成分アセットを最初に成分として使用した主張生成者によって行われるとは限りません。

詳細については、セクション 15.11.3.2「明示的な検証の実行」を参照してください。

15.10.3. アサーションの検証

15.10.3.1. 一般

クレームの created_assertions および gathered_assertions フィールド（および v1 クレームの assertions フィールド）の各アサーションは、hashed_uri 構造体である。各アサーションについて、バリデータはまず、url フィールドの URI 参照が編集済みアサーションのリストに含まれているかどうかを判断しなければならない。

NOTE gather_assertionsフィールドにリストされたアサーションは、クレーム生成者によって作成されたものではありませんが、それらは依然としてクレームの一部であり、したがってこの検証アルゴリズムに従って検証されます。

編集対象アサーションのリストに含まれている場合、アサーションのラベルが c2pa.actions または c2pa.actions.v2 である場合は、c2pa.actions および c2pa.actions.v2 アサーションは編集対象とならないため、assertion.action.redacted の失敗コードでクレームは拒否される。編集対象アサーションリストに含まれる場合、アサーションのラベルがコンテンツへのハードバインディングである場合（c2pa.hash.data、c2pa.hash.boxes、c2pa.hash.collection.data、c2pa.hash.bmff.v2（非推奨）、c2pa.hash.bmff.v3）である場合、当該クレームはassertion.dataHash.redactedの失敗コードで拒否される。これらのタイプの断言は編集対象外であるため。それ以外の場合は、編集された断言は有効と見なされ、断言のタイプに基づいて検証が継続される。

その他のすべてのアサーション（編集済みアサーションのリストに含まれないもの）については、urlフィールド内のURI参照を解決してそのデータを取得する。URIが同一C2PAマニフェスト内の位置（self#jumbf位置）を指さない場合、当該クレームはassertion.outsideManifestの失敗コードで拒否される。URIを解決できずデータを取得できない場合、assertion.missingの失敗コードでクレームを拒否する。

ハッシュアルゴリズムおよび可能性のある失敗コードを決定するには、セクション15.4「ハッシュアルゴリズムの決定」の手順に従う。そのアルゴリズムとセクション8.4.2.3「JUMBFボックスのハッシュ化」に記載の手順を用いてアサーションのハッシュを計算し、計算されたハッシュ値をハッシュフィールドの値と比較する。一致しない場合、assertion.hashedURI.mismatchの失敗コードでクレームを拒否する。一致する場合、assertion.hashedURI.matchの成功コードを記録する。

標準アサーションの内容が適切に構成されたCBORでない場合、または非準拠のJSONである場合、当該クレームはassertion.cbor.invalidまたはassertion.json.invalidの失敗コードで拒否されるものとする。

注記 妥当なCBORはRFC 8949の付録Cで定義されている。

注記 RFC 8259、Clause 2 は、JSON データが準拠する文法を定義しています。

アサーションストアに存在するアサーションが、クレーム内の created_assertions 配列または gathered_assertions 配列（v1 クレームの場合は assertions 配列）のいずれの要素からも参照されていない場合、そのクレームは assertion.undeclared の失敗コードで拒否される。

クレームの redacted_assertions 配列内の各 URI について、その URI がクレーム自身のマニフェストを指している場合、クレームは assertion.selfRedacted の失敗コードで拒否される。クレームは自身の

assertions を編集することは許可されない。

15.10.3.2. 特定の断言の検証

各アサーションについて、バリデータは当該アサーションのラベルを確認し、以下に列挙されている場合は、そのアサーションタイプに対する特定の検証手順を実行しなければならない。アサーションのラベルが以下に列挙されていない場合、そのタイプのアサーションは既に記述された検証手順以外に追加の検証手順を必要としない。

- `c2pa.cloud-data`、セクション15.10.3.2.1「c2pa.cloud-data検証」
- `c2pa.actions` または `c2pa.actions.v2`、セクション 15.10.3.2.2、「`c2pa.actions` 検証」
- `c2pa.metadata`、セクション 15.10.3.2.3、「`c2pa.metadata` 検証」

> **注記** Ingredient assertions (`c2pa.ingredient` または `c2pa.ingredient.v2` または `c2pa.ingredient.v3`) は、検証プロセスの別のポイントで追加の検証の対象となります（セクション15.11「成分の検証」を参照）。

標準アサーションの任意のフィールドの値が `hashed_uri` または `hashed_ext_uri` である場合、バリデータはセクション 15.10.3.3「参照の検証」に記載の手順を実行するものとする。ただし、`c2pa.ingredient.v3` の `activeManifest` フィールドについては例外とする。このフィールドについては、セクション 15.11.3「成分アサーションの検証」で特別な検証動作が規定されている。「成分アサーションの検証」で特別な検証動作が規定されているc2pa.ingredient.v3のactiveManifestフィールドを除く。

15.10.3.2.1. `c2pa.cloud-data`の検証

アサーションのラベルが `c2pa.cloud-data` の場合：

1. アサーションが以下のフィールドを含むことを確認する：`label`、`size`、`location`、`content_type`。これらのフィールドのいずれかが欠落している場合、クレームは失敗コード `assertion.cloud-data.malformed` で拒否される。
2. 外部アサーションのラベルフィールドが `c2pa.hash.data`、`c2pa.hash.boxes`、`c2pa.hash.collection.data`、`c2pa.hash.bmff.v2`（非推奨）、`c2pa.hash.bmff.v3` の場合、`assertion.cloud-data.hardBinding` の失敗コードでクレームを拒否する。
3. マニフェストが更新マニフェストであり、外部アサーションのラベルフィールドが `c2pa.actions` または `c2pa.actions.v2` の場合、`assertion.cloud-data.actions` の失敗コードでクレームを拒否する。
4. locationフィールドは、セクション15.10.4.2「外部参照の検証」に従って検証されるものとする。

15.10.3.2.2. `c2pa.actions`の検証

アサーションのラベルが `c2pa.actions` または `c2pa.actions.v2` の場合：

1. アクションフィールドがあることを確認する。ない場合、クレームは拒否され失敗コードが `assertion.action.malformed`.
2. アクションリストの各アクションについて：

a. アクションフィールドが `c2pa.created` または `c2pa.opened` の場合、クレームは次の理由で拒否される:

アサーションの失敗コードは、以下のすべてが真でない限り、assertion.action.malformed となる：

i. assertion が created_assertions または gathered_assertions の最初の actions assertion である場合

配列（v2クレームの場合）の最初のアクションアサーション、またはv1クレームのassertions配列内の最初のアクションアサーションであること、かつ

ii. かつ、アクションがこのアサーション内の actions 配列の最初の要素である場合。

b. アクションフィールドが c2pa.opened、c2pa.placed、または c2pa.removed の場合：

i. アクションにパラメータフィールドが存在しない場合、またはそのフィールドの値が空の場合、クレームは assertion.action.ingredientMismatch の失敗コードで拒否される。

ii. アクションのパラメータフィールドにingredientsフィールド（またはc2pa.actionsの場合はingredientフィールド）が含まれていない場合、assertion.action.ingredientMismatchの失敗コードでクレームを拒否する。

iii. ingredientsフィールドの値が少なくとも1つの要素を含む配列でない場合、assertion.action.ingredientMismatchの失敗コードでクレームを拒否する。

iv. 成分アサーションへの参照を確認する：

A. c2pa.openedの場合：ingredientsフィールド（またはc2pa.actionsの場合はingredientフィールド）に、現在のマニフェスト内のingredientアサーションに解決可能で、relationshipフィールドがparentOfである有効なハッシュ付きURIが正確に1つ含まれていることを確認する。含まれていない場合、assertion.action.ingredientMismatchの失敗コードでクレームを拒否する。

B. c2pa.placedの場合：ingredientsフィールド（c2pa.actionsの場合はingredientフィールド）に、有効なハッシュ付きURIが1つ以上含まれていることを確認してください。各URIは、現在のマニフェスト内のingredientアサーションに解決可能であり、そのrelationshipフィールドがcomponentOfである必要があります。条件を満たさない場合、assertion.action.ingredientMismatchの失敗コードでクレームを拒否します。

C. c2pa.removedの場合：ingredientsフィールド（c2pa.actionsの場合はingredientフィールド）に、有効なハッシュ付きURIが1つ以上含まれていることを確認する。各URIは、別のマニフェスト内のingredientアサーションに解決可能であり、そのrelationshipフィールドがcomponentOfでなければならない。条件を満たさない場合、assertion.action.ingredientMismatchの失敗コードでクレームを拒否する。

c. actionフィールドがc2pa.transcodedまたはc2pa.repackagedの場合：

i. ingredientsフィールド（c2pa.actionsの場合はingredientフィールド）が存在する場合、そのフィールドの各要素が、現在のマニフェスト内のingredientアサーションにresolution可能な有効なハッシュ付きURIであり、かつ関係性がparentOfであることを確認する。そうでない場合、assertion.action.ingredientMismatchの失敗コードでクレームを拒否する。

d. アクションフィールドが c2pa.redacted の場合：

i. パラメータオブジェクトのメンバーであるredactedフィールドにJUMBF URIが存在するか確認する。JUMBF URIが存在しない場合、またはアサーションに解決できない場合、assertion.action.redactionMismatchの失敗コードでクレームを拒否しなければならない。

e. action-common-map-v2 に softwareAgent フィールドが存在する場合、または actions-map-v2 の softwareAgents フィールドに一

つ以上の softwareAgent がリストされている場合：

が存在する場合は：

i. `generator-info-map` に `icon` フィールドが存在する場合、セクション 15.10.3.3「参照の検証」に記載された方法で検証される。

f. テンプレートリスト内の各テンプレートについて：

i. `action-template-map-v2` に `icon` フィールドが存在する場合、セクション 15.10.3.3「参照の検証」に記載された方法で検証される。

15.10.3.2.3. `c2pa.metadata` 検証

アサーションのラベルが `c2pa.metadata` の場合、バリデータはアサーションが許可リスト外のフィールドを含まないことを保証しなければならない。アサーションに含まれるフィールドのいずれかが許可リストにない場合、そのクレームは `assertion.metadata.disallowed` の失敗コードで拒否される。

注記	この検証要件により、バリデータはアサーションに含まれる JSON-LD データを解析する必要が生じる。

15.10.3.2.4. `c2pa.time-stamp` 検証

アサーションのラベルが `c2pa.time-stamp` の場合、バリデータはアサーションが単一のマップ（メジャータイプ `5`）で構成され、少なくとも 1 組のキー/値ペアを含む、適切に形成された CBOR であることを確認しなければならない。これに該当しない場合、クレームは `assertion.timestamp.malformed` の失敗コードで拒否される。

タイムスタンプトークンの検証はセクション15.8.2「TimeStampTokenの検証」で説明されている通りに行われるため、バリデータは後続の処理のためにタイムスタンプトークン（および関連するC2PAマニフェスト識別子）を保存する必要がある。

15.10.3.3. 参照の検証

一部のC2PA標準アサーションでは、`hashed_uri`および`hashed_ext_uri`を使用してC2PAマニフェスト内の他のボックスを参照できます。例えば、actions、ingredient、thumbnailアサーション内で様々な参照が可能となります。

標準アサーション内のすべてのhashed_uriおよびhashed_ext_uriフィールド（`c2pa.ingredient.v3`の`active`Manifestフィールドを除く。これについてはセクション15.11.3「成分アサーションの検証」で特別な検証動作が規定されている）について、バリデータは以下の検証を実行しなければならない：バリデータが取得を選択したリソースを持つ `hashed_ext_uri` については、バリデータはセクション 15.10.4.2「外部参照の検証」に記載の手順を実行しなければならない。`hashed_uri` については、バリデータは以下に記載の手順を実行しなければならない。

`hashed_uri` の宛先は、その url フィールドで特定される。このフィールドが存在しない場合、または宛先が見つからない場合（つまり、データが想定される場所に存在しない場合）、検証失敗としてコード `hashedURI.missing` で処理される。

宛先が特定できる場合、以下の手順を実行する：・セクション15.4「ハッシュアルゴリズムの決定」の手順に従い、ハッシュアルゴリズムおよび可能な失敗コードを決定する。・hashed_uri構造体にハッシュフィールドが存在することを確認する。存在しない場合、クレームは失敗コード`hashedURI.mismatch`で拒否される。・決定したハッシュアルゴリズムとセクション8.4.2.3「JUMBFボックスのハッシュ化」で記述された手順を用いて、アサーションのハッシュを計算する。・計算されたハッシュ値をハッシュフィールドの値と比較する。一致しない場合、クレームは失敗コード`hashedURI.mismatch`で拒否される。

セクション8.4.2.3「JUMBFボックスのハッシュ化」に記載の手順に従う。. 計算されたハッシュ値とハッシュフィールドの値を比較する。一致しない場合、クレームは`hashedURI.mismatch`の失敗コードで拒否される。

15.10.4. 外部データ検証

15.10.4.1. 一般

クラウドデータアサーションの内容には、外部データへのURI参照とハッシュが含まれます。これらは他のアサーションと同様に検証されますが、標準的な検証プロセスではそれらの参照は取得・検証されません。バリデータは、参照された外部データを取得しようとする前に、まずクレームの検証を成功させなければなりません。バリデータは、拒否されたクレームから外部データを取得しようと試みてはならない。外部データの取得はオプションであるため、外部データの取得または検証が不可能な場合でも、クレームが拒否されることはない。

バリデータがクラウドデータアサーション内の外部データを取得することを選択した場合、バリデータはセクション15.10.4.2「外部参照の検証」に記載された手順を実行しなければならない。

15.10.4.2. 外部参照の検証

クラウドデータアサーションで参照される外部データを検証するには、以下の手順を使用する：

1. urlフィールド内のURI参照を解決し、そのデータを取得する。`url`フィールドが存在しない場合、またはURIが解決できずデータを取得できない場合、バリデータは外部データの取得試行を中止する。
2. 取得したデータのサイズが `size` フィールドの値と一致しない場合、バリデータは `assertion.hashedURI.mismatch` の失敗コードをアプリケーションに返し、取得したデータを提供してはならない。
3. HTTPレスポンスのContent-Typeヘッダーで返されるコンテンツタイプが宣言されたコンテンツタイプと等しいことを検証する。一致しない場合、バリデータはアプリケーションに`assertion.hashedURI.mismatch`の失敗コードを返し、取得したデータを提供しない。宣言されたコンテンツタイプは以下によって決定される：
    a. 外部データの場合、コンテンツタイプは`hashed_ext_uri`構造の`dc:format`フィールドによって決定される `dc:format`フィールドが存在しない場合、コンテンツタイプの検証は常に成功する。
    b. クラウドデータアサーションにおいて、そのlocationフィールドに`dc:`formatフィールドが存在する場合、それがコンテンツタイプを決定し、クラウドデータアサーションのcontent_typeフィールドの値は無視される。`location`に`dc:format`フィールドが存在しない場合、アサーションのcontent_typeフィールドがコンテンツタイプを決定する。
4. セクション15.4.2「ハッシュ付きExt URIの場合」で規定されるハッシュアルゴリズム、または発生し得る失敗コードを決定する。
5. 決定したハッシュアルゴリズムとセクション8.4.2.3「JUMBFボックスのハッシュ化」に記載の手順を用いて、取得したコンテンツのハッシュを計算する。外部データの場合、ハッシュ関数の入力としてハッシュアルゴリズムと正確に取得したコンテンツを使用する。
    a. 計算されたハッシュ値をハッシュフィールドの値と比較する。ハッシュフィールドが存在しない場合、または一致しない場合、バリデータはアプリケーションに`assertion.hashedURI.mismatch`の失敗コードを返し、取得したデータを提供してはならない。

b. それ以外の場合は、バリデータは `assertion.hashedURI.match` の成功コードを記録し、取得したデータをアプリケーションに提供しなければならない。

15.11. 原材料の確認

15.11.1. 説明

バリデータは、提示されたアセットとそのアクティブなマニフェストに対して検証ステップを実行しなければならない。いずれかのステップでアクティブなマニフェストが無効であると結論付けられた場合、そのマニフェストは指定された失敗コードと共に拒否される。

資産のアクティブマニフェストは、成分アサーションを通じて一つ以上の成分を列挙する場合があります。それらの成分の一部は、それ自体に関連付けられたマニフェストを持つ可能性があり、またそれらのマニフェストの一部は、それ自体成分や成分マニフェストを持つ可能性があります。

15.11.2. 成分マニフェストの処理

15.11.2.1. 成分内の標準マニフェスト

標準マニフェストを処理する際、バリデータは各成分を検証するものとします（その関係フィールドの値にかかわらず）を以下に記述する通りに検証しなければならない。

15.11.2.2. 成分内のマニフェストの更新

更新マニフェストについては、更新マニフェストのparentOf成分は以下に記述する通り検証されるものとする。

15.11.2.3. 成分内のタイムスタンプマニフェスト

> **重要**
>
> この機能はタイムスタンプアサーションに取って代わられ、非推奨となりました。以下の情報は歴史的記録として保持されています。

成分内で見つかったタイムスタンプマニフェストはすべて無視される。

15.11.3. 成分アサーションの検証

15.11.3.1. 検証の概要

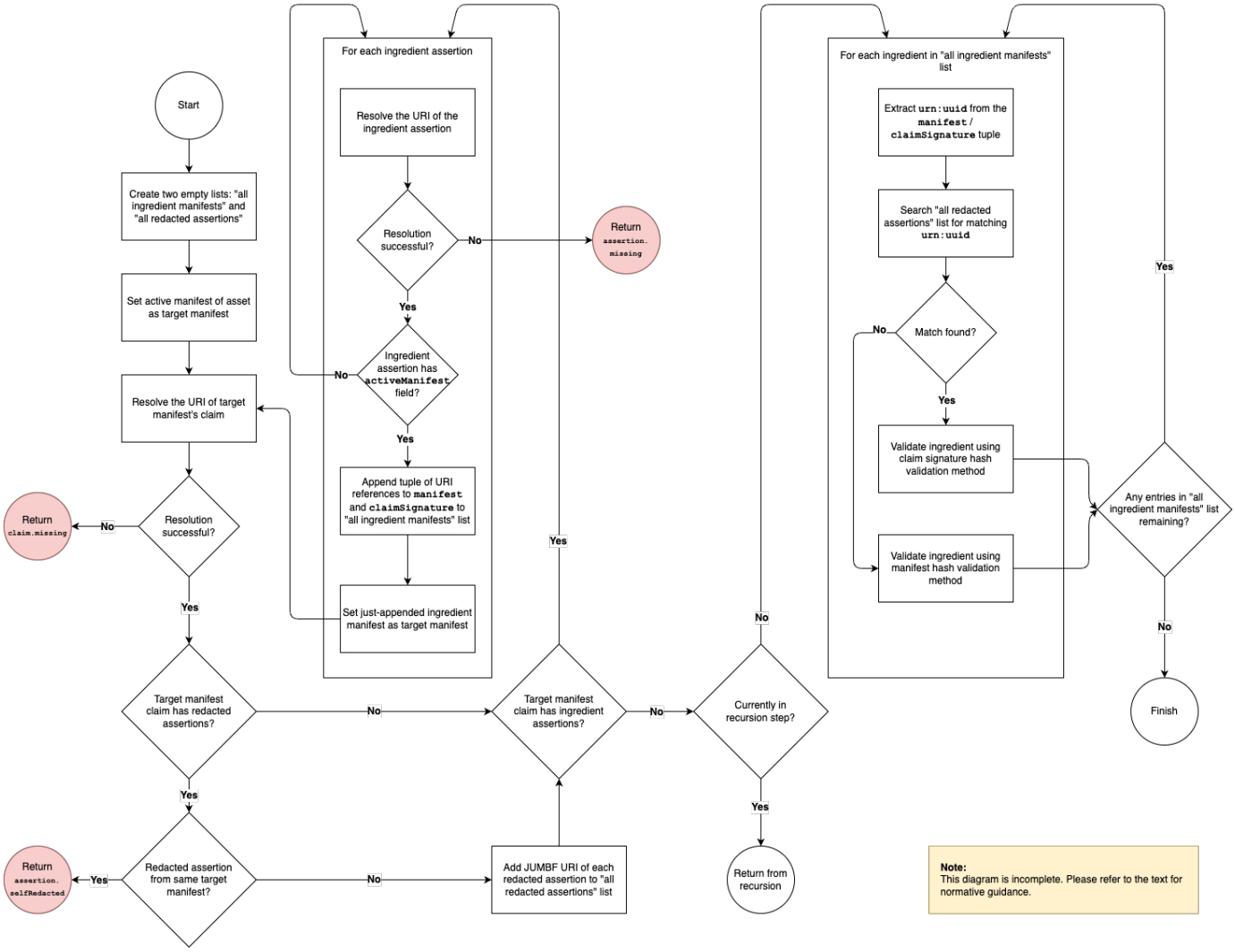

図14. 成分検証

図14「成分アサーション検証」のフローチャートは、特定のC2PAマニフェストに含まれる成分アサーションを検証するプロセスを示しています。

注記	視覚的表現とテキストに相違がある場合、テキストが優先されます。

15.11.3.2. 明示的な検証の実行

成分アサーションに `relationship` フィールドが存在しない場合、`assertion.ingredient.malformed` の失敗コードでアサーションを拒否する。

関係フィールドの値は、`parentOf`、`inputTo`、または `componentOf` のいずれかでなければならない。関係フィールドの値がこれらに該当しない場合、アサーションは `assertion.ingredient.malformed` の失敗コードで拒否される。

15.11.3.3. 再帰的検証の実行

バリデータは、深度優先探索などを使用して、アセット内のすべての成分マニフェストを再帰的に検証しなければならない。

以下に記述する。バリデータはアルゴリズムを記述どおりに実装する必要はないが、検証結果は本アルゴリズムの結果と同等であること。

1. 空のリストを2つ作成する：
    a. アセットの系譜内のどこで使用されているかに関わらず、アセットで使用されている全ての成分マニフェストのハッシュ済みURI値を保持するリスト。
    b. アセットの系譜内のどこにあっても、アセット内のすべての編集済みアサーションのJUMBF URIを保持するリスト。
2. 検証対象アセットのアクティブマニフェストをターゲットマニフェストとして設定する
3. 再帰を開始します。
4. セクション15.6「クレームの特定と検証」に記載されている手順に従い、クレームを特定する。特定できない場合、失敗コード claim.missingでクレームを拒否する。
5. 対象マニフェストのクレームに `redacted_assertions` フィールドが含まれる場合、各編集済みアサーションの JUMBF URI を確認する。
    a. 編集済みアサーションが対象マニフェストからのものである場合、`assertion.selfRedacted` 失敗コードでクレームを拒否する。
    b. そうでない場合、編集済みアサーションのリストに、編集済みアサーションのJUMBF URIを追加する。
6. 対象マニフェストのクレームに成分アサーションが含まれる場合：
    a. 各成分アサーションについて：
        i. 成分アサーションのハッシュ付きURIを解決しようとする。URIが解決されない場合、ハッシュが一致しない場合、またはアサーションのJUMBFコンテンツボックスがすべてゼロのみを含む場合、次の成分アサーションへスキップする。
        ii. 成分アサーションに `activeManifest` フィールド（v1 または v2 成分アサーションの場合は `c2pa_manifest` フィールド）がある場合：
            A. 以下の値を含むタプルを全成分マニフェストのリストに追加する：
                - ingredientアサーション内のactiveManifest（または`c2pa_manifest`）フィールドのhashed_uri値
                - ingredientアサーション内の`claimSignature`フィールドの`hashed_uri`値
            B. 追加したばかりの成分マニフェストをターゲットマニフェストとして設定し、「再帰開始」ステップから上記のプロセスを繰り返す。
        iii. 成分アサーションに `activeManifest` (または `c2pa_manifest`) フィールドが存在しない場合、relationship フィールドの値が `inputTo` でない限り `ingredient.unknownProvenance` 情報コードを記録し、すべての成分アサーションが尽きるまで次の成分アサーションへスキップする。その時点で現在の再帰レベルから戻る。
7. ターゲットマニフェストのクレームに成分アサーションが含まれていない場合、現在の再帰レベルから戻る。
8. 再帰を終了する。

すべての成分マニフェストのリストとすべての編集済みアサーションのリストをまとめた後、バリデータは以下の検証アルゴリズムを実行する。

以下の検証アルゴリズムを実行する：

1. 全成分マニフェストリスト内の各成分マニフェストについて：
    a. 各タプルから成分マニフェストJUMBF URIを抽出し、マニフェストラベルを取得する
    b. すべての編集済み断言リストから、一致するマニフェストラベルを持つ断言を検索する
    c. 一致する編集済み断言が一つ以上見つかった場合：
        i. セクション15.11.3.3.1「クレーム署名ハッシュ検証方法」に記載されているクレーム署名ハッシュ検証方法を使用して、成分を検証する。
    d. 一致する編集済みアサーションが見つからない場合：
        i. セクション15.11.3.3.2「マニフェストハッシュ検証方法」に記載されているマニフェストハッシュ検証方法、またはセクション15.11.3.3.1「クレーム署名ハッシュ検証方法」に記載されているクレーム署名ハッシュ検証方法のいずれかを使用して、成分を検証する。
    e. 成分アサーションに `validationResults` フィールドが含まれる場合：
        i. validationResultsフィールドの値の各エントリについて、検証プロセスで同等のエントリが返されなかった場合、検証結果の一部としてそれを返す。
        ii. 検証プロセスの一部として返されたエントリのうち、validationResultsフィールドに存在しないエントリがある場合、それを検証結果の一部として返す。
    f. validationResultsフィールドが存在せず、かつingredientアサーションがv3形式のingredientアサーションで `activeManifest` フィールドが存在する場合、失敗コード `assertion.ingredient.malformed` を返す。

バリデータは、C2PA マニフェストストアに存在するが、構成要素マニフェストのリストに含まれていない追加の C2PA マニフェストをすべて無視する必要があります。

> **注** 追加の C2PA マニフェストを無視することで、カスタムアサーションや、バリデータが認識しない方法で C2PA マニフェストを参照する可能性のある将来の構造との互換性がサポートされます。

15.11.3.3.1. クレーム署名ハッシュ検証メソッド

このメソッドは、アクティブマニフェストに対して行われるものと同様に、成分のクレームの完全な検証を含みます。ただし、コンテンツバインディングは評価されません：

1. claimSignatureフィールドのurl値内のURI参照を解決し、成分のクレーム署名ボックスを取得します。URI参照が解決できない場合、または `claimSignature` フィールドが存在しない場合、成分クレームはfailure code `ingredient.claimSignature.missing` で拒否されます。
2. セクション15.4「ハッシュアルゴリズムの決定」の手順に従い、ハッシュアルゴリズム識別子（または失敗コードの可能性）を決定する。
3. セクション8.4.2.3「JUMBFボックスのハッシュ化」に記載の手順とアルゴリズムを用いて、成分表示署名ボックスのハッシュ値を計

算する。

4. 計算されたハッシュ値をハッシュフィールドの値と比較する。

a. ハッシュが一致しない場合、またはハッシュフィールドが存在しない場合：

   i. 失敗コード `ingredient.claimSignature.mismatch` を付してクレームを拒否する。

b. ハッシュが一致する場合：

   i. セクション15.7「署名の検証」、セクション15.8「タイムスタンプの検証」、およびセクション15.9「認証情報の失効情報の検証」に従い、クレーム署名、タイムスタンプ、および認証情報の失効情報を検証する。

   ii. リスト内の編集済みアサーションのうち、マニフェストラベルが一致するURIごとに、参照されたアサーションが存在し、かつその内部のJUMBFコンテンツボックスまたはパディングボックスのいずれかがゼロまたは複数の`0x00`バイト以外の内容を含む場合、当該クレームは`assertion.notRedacted`の失敗コードで拒否されるものとする。

   iii. ハードバインディングアサーション（原材料については検証不可）を除き、各非編集済みアサーションをセクション15.10「アサーションの検証」に従って検証する。

クレーム署名ハッシュ検証方式を使用する場合、検証者は `activeManifest` フィールドのハッシュ不一致失敗コードを記録してはならない。

> **注記** この理由は、編集が参照されるマニフェストに影響を与える場合、このフィールドのハッシュが一致しなくなる可能性があるためです。

15.11.3.3.2. マニフェストハッシュ検証方法

編集により変更されていない成分マニフェストは、現在の検証者が以前のクレーム生成者の検証結果を信頼している場合、より迅速に検証できます：

1. activeManifestフィールドのurl値にあるURI参照を解決し、成分のマニフェストボックスを取得します。`url`フィールドが存在しない場合、またはURI参照が解決できない場合、成分クレームはfailure code `ingredient.manifest.missing`で拒否されます。
2. セクション15.4「ハッシュアルゴリズムの決定」の手順に従い、ハッシュアルゴリズム識別子（または失敗コードの可能性）を決定する。
3. セクション8.4.2.3「JUMBFボックスのハッシュ処理」に記載の手順に従い、そのアルゴリズムを用いて成分マニフェストボックスのハッシュを計算する。
4. 計算されたハッシュ値をハッシュフィールドの値と比較する。
   a. ハッシュが一致しない場合、またはハッシュフィールドが存在しない場合：
      i. 失敗コード `ingredient.manifest.mismatch` でクレームを拒否する。
   b. ハッシュが等しい場合、成分は完全に検証され、success code `ingredient.manifest.validated` 成功コードを発行する。

15.12. アセットのコンテンツを検証する

アセットの内容は、アクティブなマニフェストが標準マニフェストである場合、そのハードバインディングを用いて検証される。アクティブなマニフェストが更新マニフェストである場合、ハードバインディングは親の成分（`parentOf` ingredient）のマニフェスト内で見出され

る。

マニフェストでハードバインディングを検索する。そのマニフェストも更新マニフェストである場合、parentOfイングリディエントの連鎖をたどり最初の標準マニフェストで検索する。標準マニフェストが見つからない場合、または標準マニフェストにハードバインディングが存在しない場合、アクティブマニフェストのクレームは失敗コード`claim.hardBindings.missing`で拒否される。

アセットは複数のパーツで構成される場合があり、各パーツには固有の関連ハッシュ（セクション18.9「マルチアセットハッシュ」参照）が割り当てられ、個別に検証可能である。例えば、アセットが静止画像と動画の別々のパーツで構成される場合、各パーツを個別に検証できる。

15.12.1. データハッシュの検証

15.12.1.1. 一般事項

標準マニフェスト（およびそのバインディング）が特定された後、除外範囲は
`c2pa.hash.`dataアサーションから除外範囲を抽出する。

1 つの除外範囲の終了バイトオフセット（開始位置 + 長さ）が、配列内の次の除外範囲の開始バイトオフセットよりも大きい場合、または開始位置または長さの値が負の場合、マニフェストは `assertion.dataHash.malformed` の失敗コードで拒否される。

更新マニフェストが検出された場合、C2PAマニフェストストア全体の開始位置をオフセットとする除外範囲の長さ値は、C2PAマニフェストストア全体の長さにファイル形式固有の追加要素を加えた値として扱われる。

`c2pa.hash.data` で指定されたハッシュアルゴリズム (`alg`) は、除外範囲で指定されたバイトを除き、アセットの全バイトに対して計算される。除外範囲の終了位置がアセットの終了位置を超える場合、マニフェストは `assertion.dataHash.mismatch` の失敗コードで拒否される。

algフィールドで指定されたハッシュアルゴリズムが、セクション13.1「ハッシュ処理」の許可リストまたは非推奨リストに記載されていない場合、マニフェストは`algorithm.unsupported`の失敗コードで拒否される。hashフィールドが存在しない場合、マニフェストは`assertion.dataHash.mismatch`の失敗コードで拒否される。

除外範囲とパディング値の組み合わせ、特にマルチパス処理ワークフローをサポートするために必要なパディングは、攻撃者がハッシュを無効化せずにアセットの消費に影響を与える可能性のある任意のデータでそのパディングの一部を置き換えることを可能にするかもしれない。このため、バリデータは除外範囲（C2PAマニフェストストアを含む）内に含まれるデータが、C2PAマニフェストストアと適切なパディング（例：ゼロ埋めデータ）のみで構成され、明確にマークされたパディングフィールドまたはフリー/スキップボックスに配置されていることを保証しなければならない。上記C2PAマニフェストストア以外の除外範囲内では、セクション9.2.5「アセットメタデータバインディング」に記載の通り、アセットメタデータの一部または全部が含まれる場合がある。バリデータが上記で許可されたデータ以外のデータに遭遇した場合、マニフェストは`assertion.dataHash.mismatch`の失敗コードで拒否される。検証者がC2PAマニフェストストア以外の除外範囲、または明示的にマークされたパディングフィールドもしくはフリー/スキップボックス内の適切なパディング（例：ゼロ埋めデータ）を検出した場合、情報コード`assertion.dataHash.additionalExclusions`Presentを設定する。

エラー条件が発生しなかった場合、バリデータは成功コード `assertion.dataHash.match` を追加する。
最終的に返されるリストに対して。

アセットの全データ（除外範囲を除く）に対して計算されたハッシュが、`c2pa.hash.data` のハッシュフィールドの値と一致しない場合、バリデータはマルチアセットハッシュアサーションの存在を確認する。存在する場合、セクション15.12.4「マルチアセットハッシュの検証」に記載された方法で検証を行う。存在しない場合、マニフェストは`assertion.dataHash.mismatch`の失敗コードで拒否される。

15.12.1.2. JPEG 1ファイルのハッシュ処理

JPEG 1ファイルにおいて、前述のファイル形式の追加情報は、APP11セグメント用の`APP11`マーカーおよびそれに対応するセグメント長バイトを含む。セグメント長は除外範囲内に含まれるため、バリデータは除外範囲の合計長と、C2PAマニフェストを表す全APP11セグメントの合計長とを照合し、長さが改ざんされていないことを確認しなければならない。

> **注記**
>
> JPEG 1ファイルには、C2PA以外の目的（例：JPEG 360やJPEGプライバシーとセキュリティ）でAPP11セグメントが含まれる場合があり、それらは本計算の対象外である。

15.12.2. BMFFハッシュの検証

ユーザーへの表示のためにレンダリングされるアセットの一部（音声、動画、テキストを含むがこれらに限定されない）については、レンダリングされたコンテンツに対応するハードバインディングを第9.2節「ハードバインディング」に従って検証しなければならない。標準のハードバインディングが検証に失敗し、かつマルチアセットハッシュアサーションが存在する場合、[validating_a_multi_asset_hash]に記載の方法で検証を行う。コンテンツの検証が失敗した場合は、検証ツールはユーザーに対し、コンテンツの一部が主張と一致しないことを明確に通知し、可能であれば検証に失敗したコンテンツの箇所を示すべきである。コンテンツバインディングが存在するコンテンツが欠落している場合、この欠落の発見も検証失敗とみなす。コンテンツの後半部分が正しく検証された場合でも、検証ツールは検証失敗を継続して報告しなければならない。

レンダリング開始前に完全に利用可能でないコンテンツ（適応ビットレートストリーミング（ABR）やプログレッシブダウンロード時など）については、未利用部分の存在は検証失敗とはみなされない。コンテンツが利用可能になるにつれ、検証ツールは前述の手順に従い、各コンテンツ部分をレンダリング前に検証しなければならない。さらに、バリデータは当該コンテンツのシーケンスがマニフェスト生成時と同一であることを検証する。プレイヤーが断続性を明示的に通知した場合（例：ユーザーがUI経由で手動シーク操作を実行時）を除き、シーケンスが一致しない場合は常に、予期せぬ断続が発生したことをユーザーに明確に通知する。これには、特定のマークルツリーの位置値がゼロから始まり、続く各チャンクごとに1ずつ増加していることの検証が含まれる。等価的に、位置値は常にどのチャンクがレンダリングされているかを示す。

プログレッシブダウンロードによる再生中に検証されるコンテンツの場合、マークルツリーのリーフノードは『`mdat`』内のビデオトラックの同期ポイント（例：RAPポイント（ランダムアクセスポイント））に整合させることが可能である。『`variableBlockSizes`』をこの整合を達成するよう設定すれば、直線的な再生中または指定再生時間へのシーク時の検証を同一シーケンスで実現できる。目的のブロックを取得し、検証し、その中のトラックをレンダリング用に選択する。

早送り、巻き戻し、または異なる速度での再生など、クレーム生成者が当初意図した方法とは異なる方法で意図的にレンダリングされないコンテンツについては、バリデータがコンテンツを検証できない場合があります。この場合、

検証ツールは、対応する操作中にコンテンツを検証できないことをユーザーに明確に通知しなければならない。

box_purposeがupdateに設定されたC2PA ContentProvenanceBoxを持つコンテンツについては、アクティブなマニフェストをまずbox_purposeがupdateに設定されたC2PA ContentProvenanceBox内で検索し、次にbox_purposeがoriginalに設定されたC2PA ContentProvenanceBox内で検索します。アクティブなマニフェストが box_purpose が update に設定された C2PA ContentProvenanceBox にある場合、最初の非更新マニフェストが見つかるまで、成分親チェーンを追跡します（必要に応じて box_purpose が update または original に設定された C2PA ContentProvenanceBox を参照）。このマニフェストコンテンツの BMFF ハッシュは、セクション 9.2「ハードバインディング」に従って検証されるものとします。box_purpose が update に設定された C2PA ContentProvenanceBox の追加は、ファイル末尾に追加されオフセットを変更しないため、ハッシュ計算に影響を与えない。

bmff-hash-mapにexclusionsフィールドが含まれていない場合、またはそのフィールドの値が少なくとも1つのエントリを持つ配列型でない場合、マニフェストはassertion.bmffHash.malformedの失敗コードで拒否される。

ハッシュアルゴリズム識別子（または失敗コードの可能性）は、セクション15.4「ハッシュアルゴリズムの決定」の手順に従って決定する。

1 つのサブセット範囲の終了バイトオフセット（オフセット + 長さ）が、配列内の次の範囲のオフセット値よりも大きい場合、またはオフセット値もしくは長さ値が負の場合、マニフェストは assertion.bmffHash.malformed の失敗コードで拒否される。assertion.bmffHash.mismatch 失敗コードは、このセクションで説明するその他のすべての失敗に使用される。それ以外の場合は、バリデータは最終的に返すリストに成功コード assertion.bmffHash.match を追加しなければならない。

BMFF ハッシュ処理によって assertion.bmffHash.mismatch 失敗コードが生成された場合、バリデータはマルチアセットハッシュアサーションの存在を確認する。存在する場合、assertion.bmffHash.mismatch 失敗コードは発行されず、代わりにマルチアセットハッシュアサーションはセクション 15.12.4「マルチアセットハッシュの検証」に記載されているように検証される。そうでない場合、マニフェストは assertion.bmffHash.mismatch 失敗コードで拒否される。

15.12.2.1. マークルツリーを使用した非断片化資産

bmff-hash-map内のmerkleフィールドが存在する場合、バリデータはMerkleツリーを検証しなければならない。bmff-merkle-map内のfixedBlockSizeとvariableBlockSizesが存在しない場合、mdatのペイロード全体はハッシュ計算において単一のリーフノードとして扱われる。fixedBlockSizeが存在しvariableBlockSizesが存在しない場合、mdatのペイロードは固定長ブロックに分割され、各ブロックはリーフノードとして扱われる。最終ブロックがmdatペイロードの末尾を超える場合、最後のブロックのサイズはmdatペイロードの末尾までしか延長しないように設定されるべきである。variableBlockSize が存在し、fixedBlockSizes が存在しない場合、mdat のペイロードは variableBlockSizes の配列で定義されたサイズに分割される。要素数が count と等しくない場合、または値の合計が mdat のペイロードサイズと等しくない場合、マニフェストは assertion.bmffHash.malformed の失敗コードで拒否される。bmff-merkle-map内のfixedBlockSizeとvariableBlockSizesが存在する場合、マニフェストはassertion.bmffHash.malformedの失敗コードで拒否される。

bmff-merkle-map内のcountが、bmff-merkle-map内のハッシュ要素数と等しい場合
そして、リーフノードのハッシュがbmff-merkle-map内のハッシュ要素と一致しない場合、マニフェストは

assertion.bmffHash.mismatch の失敗コードで拒否される。bmff-merkle-map のカウントが bmff-merkle-map 内のハッシュ要素数より少なく、かつ補助 uuid C2PA ボックスがセクション A.5.4 に記述されているように存在しない場合、「大規模かつ断片化されたファイルのための補助 'c2pa' ボックス」で説明されているように補助 UUID C2PA ボックスが存在しない場合、マニフェストは assertion.bmffHash.malformedの失敗コードで拒否される。補助UUID C2PAボックスとリーフノードから計算されたハッシュがbmff-merkle-map内のハッシュ要素と一致しない場合、マニフェストはassertion.bmffHash.mismatchの失敗コードで拒否される。bmff-merkle-map内のカウントがbmff-merkle-map内のハッシュ要素数より大きい場合、マニフェストはassertion.bmffHash.malformedの失敗コードで拒否される。

15.12.2.2. Merkleツリーを使用した断片化された資産

bmff-hash-map内のマークルフィールドが存在する場合、バリデータはマークルツリーを検証しなければならない。セクションA.5.4「大規模かつ断片化されたファイルのための補助'c2pa'ボックス」に記載されている補助UUID C2PAボックスが存在しない場合、マニフェストは assertion.bmffHash.malformedの失敗コードで拒否される。補助UUID C2PAボックスとリーフノードから計算されたハッシュが、bmff-merkle-map内のハッシュ要素と一致しない場合（ ではなく である場合）、 すると マニフェスト は 拒否される 失敗コード
assertion.bmffHash.mismatch

15.12.3. 汎用ボックスハッシュの検証

標準マニフェスト（およびそのバインディング）が特定された後、検証対象のボックスリストは、c2pa.hash.boxes アサーションに格納された box-map 構造体の boxes フィールドから抽出される。当該フィールドが存在しない場合、マニフェストは assertion.boxesHash.malformed の失敗コードで拒否される。

ボックスは、C2PAマニフェストを含むボックスを含め、boxes配列に現れるのと同じ順序でアセットに現れるものとする。アセットに他のボックスが存在する場合、マニフェストはassertion.boxesHash.unknownBoxの失敗コードで拒否される。ボックスが順序通りでない場合、マニフェストはassertion.boxesHash.mismatchの失敗コードで拒否される。

いずれかのボックスのハッシュ値が一致せず、かつそのボックスに値が true の除外フィールドが存在しない場合、マニフェストは assertion.boxesHash.mismatch の失敗コードで拒否される。それ以外の場合は、バリデータは最終的に返すリストに assertion.boxesHash.match の成功コードを追加する。

algフィールドで指定されたハッシュアルゴリズムが、セクション13.1「ハッシュ処理」の許可リストまたは非推奨リストに記載されていない場合、またはalgフィールドがbox-mapまたは特定のbox-hash-mapのいずれにも記載されていない場合、マニフェストは失敗コードalgorithm.unsupportedで拒否される。

boxes配列内のいずれかのボックスハッシュマップがnamesフィールドを含まない場合、マニフェストはassertion.boxesHash.malformedの失敗コードで拒否される。

names配列およびboxes配列にリストされた各ボックスについて、指定されたハッシュアルゴリズムはボックスのバイトデータ（関連するヘッダーを含む）に対して計算される。names配列に複数のエントリが存在する場合、その範囲内のボックス群に対するハッシュ値は、範囲内の最初のボックスの先頭から最後のボックスの末尾までを 基 に計算される。これにはボックス間に存在する任意のバイトデータも含まれる。

ハッシュフィールドが存在しない場合、または結果のハッシュがそれらのボックスのハッシュフィールドの値と一致しない場合、マニフェストは`assertion.boxesHash.mismatch`の失敗コードで拒否される。ボックスハッシュ処理が`assertion.boxesHash.mismatch`の失敗コードを生成した場合、バリデータはマルチアセットハッシュアサーションの存在を確認する。存在する場合、セクション15.12.4「マルチアセットハッシュの検証」に記載の方法で検証を行う。存在しない場合、マニフェストは`assertion.boxesHash.mismatch`失敗コードで拒否される。

15.12.3.1. JPEGの特別な取り扱い

JPEGを検証する際、バリデータは、特別な`C2PA`ボックス識別子で識別される各ボックスが、実際にC2PAマニフェストストアの一部または全部を含むAPP11であることを確認しなければならない。C2PAマニフェストストアは、セクション11.1.4.2「マニフェストストア」で説明されているように、ラベル`c2pa`を持つJUMBFスーパーボックスであり、JUMBFタイプ`UUID`が`63327061-0011-0010-8000-00AA00389B71`であることで識別される。

C2PAマニフェストストアの一部ではない`APP11`が存在し、ハッシュ付きボックスのリストに含まれていない場合、マニフェストは`assertion.boxesHash.unknownBox`の失敗コードで拒否される。

15.12.3.2. フォントの特別な取り扱い

フォントを検証する際、バリデータはフォントの`C2PA`テーブルに対応するボックスが存在することを確認し、そのボックスに埋め込みマニフェスト、リモートマニフェストURI、またはその両方が含まれているかを判定しなければならない。

いずれのボックスにも含まれないフォントテーブルが存在する場合、マニフェストは `assertion.boxesHash.unknownBox` の失敗コードで拒否される。

15.12.4. マルチアセットハッシュの検証

アセットのハードバインディングの標準検証が失敗し、かつアセットにマルチアセットハッシュアサーションが含まれている場合、バリデータはマルチアセットハッシュアサーションの検証を続行する。複数のマルチアセットハッシュアサーションが存在する場合、マニフェストは失敗コード `assertion.multiAssetHash.malformed` で拒否される。

マルチアセットハッシュアサーション（`c2pa.hash.multi-asset`）の検証は、マルチアセットハッシュマップ内のパーツ配列を反復処理して行う。`parts`フィールドが存在しない場合、または空の配列値で存在する場合、マニフェストは`assertion.multiAssetHash.malformed`の失敗コードで拒否される。

各パートについて、バリデータは有効なロケーターと有効なハッシュアサーションフィールドの両方が含まれていることを確認しなければならない。いずれかが欠落している場合、マニフェストは`assertion.multiAssetHash.malformed`の失敗コードで拒否される。

ロケーターがバイトオフセットロケーターの場合、バリデータはbyteOffsetフィールドとlengthフィールドが存在し、非負であり、アセットの総長を超えないことを保証しなければならない。いずれかが欠落しているか、負であるか、大きすぎる場合、マニフェストは`assertion.multiAssetHash.malformed`の失敗コードで拒否される。

ロケータが `bmffBox` で表現される場合、バリデータは指定されたボックスが

asset. box is not present, then the manifest shall be rejected with a failure code of assertion.multiAssetHash.malformed.

有効なロケーターとハッシュが与えられた場合、バリデータはロケーター情報を使用してパーツの検索を試みる。該当パーツが存在せず、かつオプションフィールドが存在しないか、または値がfalseである場合、マニフェストはassertion.multiAssetHash.missingPartの失敗コードで拒否される。オプションフィールドが存在し値がtrueである場合、バリデータはこのパーツをスキップし次のパーツへ進む。

> **注記** 特定パーツの破棄は、バリデータが残りのパーツを明確に識別できなくなる可能性がある。ほとんどの場合、ファイルの中間にあるパーツではなく、末尾にある1つ以上のパーツのみを効果的に破棄できる。

検出された部分が重複している場合、または全体としてアセットの全バイトをカバーしていない場合、マニフェストは assertion.multiAssetHash.malformed の失敗コードで拒否される。

検出された各部分について、バリデータは指定されたアルゴリズムと方法論（データハッシュ、一般ボックスハッシュ、またはBMFFハッシュ）を用いて、その部分のバイト列に対するハッシュを計算する。結果のハッシュがhashAssertionフィールドから参照されるハードバインディングアサーション内の値と一致しない場合、マニフェストはassertion.multiAssetHash.mismatchの失敗コードで拒否される。

各特定されたパーツのハッシュアサーションが正常に検証された場合、バリデータは成功コード assertion.multiAssetHash.match を記録し、アセットのハードバインディングに関連する失敗コードは一切記録しない。

15.12.5. コレクションデータハッシュの検証

15.12.5.1. 一般事項

標準マニフェスト内で特定されたコレクションデータハッシュアサーション（c2pa.hash.collection.data）の検証は、collection-data-hash-map 内の URI 配列を反復処理して実施する。uri フィールドが存在しない場合、マニフェストは assertion.collectionHash.malformedの失敗コードで拒否される。

使用する具体的なハッシュアルゴリズムは alg フィールドの値から決定され、セクション 15.4.3「アルゴリズム検証」で規定される通りに処理される。alg フィールドが存在しない場合、マニフェストは assertion.collectionHash.malformed の失敗コードで拒否される。

uri配列内の各uri-hashed-data-mapについて、バリデータはuriフィールドとhashフィールドの両方が含まれていることを確認しなければならない。いずれかのフィールドが欠落している場合、マニフェストはassertion.collectionHash.malformedの失敗コードで拒否される。

潜在的なセキュリティ上の懸念を回避するため、バリデータは使用前にURI（すなわちuriフィールドの値）を検証し、URIの一部として「.」または「..」が含まれていないことを保証しなければならない。いずれかがURI内で検出された場合、マニフェストは assertion.collectionHash.invalidURIの失敗コードで拒否される。

URIから取得したアセットについては、そのデータの全バイトに対して指定されたアルゴリズムを用いてハッシュを計算しなければならない。結果のハッシュがハッシュフィールドの値と一致しない場合、マニフェストは`assertion.collectionHash.mismatch`の失敗コードで拒否される。それ以外の場合、バリデータは最終的に返すリストに`assertion.collectionHash.match`の成功コードを追加しなければならない。

コレクションデータハッシュアサーションにリストされているファイルのうち、バリデータによって見つからないファイルがある場合、マニフェストは失敗コード `assertion.collectionHash.incorrectFileCount` で拒否される。

15.12.5.2. ZIP用エクストラ

C2PAマニフェストが関連付けられたZIPファイルでは、コレクションデータハッシュに追加の`zip_central_`directory_hashフィールドが含まれる。前述の通り、このフィールドにはZIP中央ディレクトリ内の各「中央ディレクトリヘッダー」および「中央ディレクトリ終了レコード」（ZIPファイルの最終部分）のハッシュが含まれる。このフィールドに使用されるハッシュアルゴリズムは、`c2pa.hash.collection.`dataアサーションの`hash`フィールドで使用されるものと同一である。

ZIPファイルを検証する際、バリデータはzip_central_directory_hashフィールドが存在し、ZIP中央ディレクトリと「中央ディレクトリ終了レコード」のハッシュがその値と一致することを確認しなければならない。ハッシュが一致しない場合、マニフェストは`assertion.collectionHash.mismatch`の失敗コードで拒否される。

第16章 ユーザーエクスペリエンス

16.1. アプローチ

C2PAは、プロバンス対応ユーザー体験（UX）の実装者向けに明確な推奨事項とガイダンスを提供することを目的としています。これらの推奨事項の開発は、多様なステークホルダーが関与する継続的なプロセスであり、その結果は、統一性と親しみやすさと、コンテキスト、プラットフォーム、デバイスを横断したユーザーにとっての実用性と柔軟性のバランスを取ったものとなります。これらの推奨事項は、ユーザー体験ガイダンス文書に記載されています。

16.2. 原則

UX推奨事項は、C2PAプロビネンスを消費者に提示するためのベストプラクティスを定義することを目的としています。これらの推奨事項は、以下の標準的で容易に認識可能な体験を記述するよう努めています：

- * コンテンツ作成者が自身の制作物に関する情報と履歴を捕捉する手段を提供し、
- 資産の利用者に、彼らが体験しているコンテンツに関する情報と履歴を提供し、それによってコンテンツの出所を理解し、どの程度信頼すべきかを判断できるようにする。

C2PAプロバンスの消費向けに設計されたユーザーインターフェースは、アセットのコンテキストに基づいて構築されるべきである。我々は、C2PA資産が遭遇する4つの主要なユーザーグループと一連のコンテキストを研究した。これらのユーザーグループは、C2PAガイディング・プリンシプルにおいて、消費者、作成者、発行者、検証者（または調査者）として定義されている。共通のコンテキストにおいてこれらの各グループのニーズに応えるため、多くの一般的なケースに対する模範的なユーザーインターフェースが提示される。これらは推奨事項であり、義務ではない。ベストプラクティスは進化していくものと期待している。

16.3. 開示レベル

特定の資産に関する完全なC2PAデータセットはユーザーにとって圧倒的となる可能性があるため、設計の指針となる段階的開示の4レベルを定義します：

- レベル1：C2PAデータが存在すること、およびその暗号学的検証ステータスの表示。
- レベル2：特定の資産について利用可能なC2PAデータの概要。このレベルでは、特定のコンテンツ、ユーザー、コンテキストに対して十分な情報を提供し、消費者が資産が現在の状態に至った経緯を十分に理解できるようにすべきである。
- レベル3：関連するすべての来歴データの詳細な表示。特定の項目の関連性は文脈に依存し、UX実装者によって決定されることに留意。
- レベル4：高度なフォレンジック調査用途には、署名と信頼シグナルの細部までを可視化するツールの使用が推奨される。

16.4. 公開レビュー、フィードバック、および進化

UX推奨事項を作成するチームは、その限界と潜在的な偏りを認識しており、フィードバック、レビュー、ユーザーテスト、継続的な進化が成功の鍵であることを理解しています。したがって、これらの推奨事項は、多様なアプリケーションやシナリオにおけるC2PA UXの実世界での展開経験に基づいて進化する文書となります。

第17章 情報セキュリティ

17.1. 脅威とセキュリティ上の考慮事項

このセクションでは、C2PAコア仕様書に記載されている技術に関する情報セキュリティ上の考慮事項とプロセスの概要を説明します。より詳細な内容は、ガイダンス文書を含む今後のC2PA資料のリリースで提供されます。

17.1.1. 背景

情報セキュリティはC2PAの主要な関心事項です。C2PAは仕様に対する脅威モデルとセキュリティ上の考慮事項を維持しています。この取り組みはC2PA内の他のセキュリティ関連作業を補完するものです。関連文書は現在開発中であり、「セキュリティ上の考慮事項」で確認できます。

C2PAは、以下の内容を含むセキュリティ考慮事項文書を開発中です：

- C2PA技術の関連セキュリティ機能の概要
- C2PA技術の実用化におけるセキュリティ上の考慮事項
- C2PA技術に対する脅威及びそれらの脅威に対する対応策（対策を含む）

17.1.2. 脅威モデリングプロセスの概要

C2PAは設計開発段階からセキュリティを組み込む一方で、システム・エコシステム・脅威環境の進化に伴い、セキュリティ設計と脅威モデリングを継続的に実施することを前提としています。

この目的のため、C2PAは強力なセキュリティおよびプライバシー設計の開発を支援する、焦点を絞った脅威モデリングプロセスを採用しています。この取り組みの成果は、明示的な脅威とセキュリティ考慮事項の文書化を直接支援するだけでなく、設計プロセス全体を通じたセキュリティ思考を促進します。

脅威モデリングプロセスは、特定分野の専門家（SME）による集中的なグループで構成される同期型（ライブ）脅威モデリングセッションと、非同期型のコンテンツ開発を組み合わせたものです。各同期セッションの参加者は効率的な議論を促進するため少人数に抑えられますが、C2PAの全メンバーはいずれかの形式で参加する機会があります。

他のセキュリティ活動と同様に、脅威モデリングプロセスもC2PAエコシステムと共に進化していくものと想定しています。プロセス文書は、C2PA内における脅威モデリングの厳格な指針ではなく、ガイドとして位置付けられています。

17.1.2.1. 参考文献

C2PAの脅威モデリングおよび関連するセキュリティ活動には、様々な参照資料や経験が活用されています。本セクションでは、参照用の公開文書のサブセットを提供します。

- セキュリティに関するIETFの検討事項

- IETFのプライバシーに関する考慮事項（ガイドライン）
- W3Cセキュリティおよびプライバシー自己評価質問票
- OAuth2脅威モデル（例）
- 脅威モデリング：セキュリティを考慮した設計
- OWASP 脅威モデリング
- Microsoft 脅威モデリング

17.2. 危害、誤用、悪用

17.2.1. はじめに

C2PAの指針原則では、C2PA仕様は、意図しない危害、人権への脅威、または世界中の脆弱なグループに対する不均衡なリスクを引き起こす可能性のあるフレームワークの悪用や誤用に対して批判的な視点で検討されるべきであると定めています。

C2PAがこの原則の側面を満たしていることを保証するため、危害・誤用・悪用評価は、仕様開発段階およびその後の実装過程で遭遇する可能性のある懸念事項を特定し対処することを目的としています。

さらに、仕様書は以下を目的として見直されています：

- 潜在的な悪用や誤用を予測し軽減すること；
- ユーザーの一般的なプライバシー懸念に対応すること；および
- 世界中のユーザーと関係者のニーズを考慮すること。

17.2.2. 考慮事項

危害・悪用・乱用の評価は継続的なプロセスである。危害モデリング文書に記載された情報は、包括的評価の最終結果ではなく、影響を受けるコミュニティを中心に、潜在的な悪用・乱用の軽減と人権保護を目的とした継続的議論の基盤として捉えるべきである。

このアプローチには二つの重要な側面がある：

継続的

C2PAの設計・開発段階から実装・利用段階に至るまで、危害・誤用・悪用評価は常に付随する。これは仕様策定プロセス、実装・ユーザー体験ガイドライン、啓発活動、連合のガバナンス、そして多様なC2PAエコシステムを促進するための多国間協力の可能性に対し、継続的に情報を提供することで実現される。このエコシステムは幅広いグローバルな文脈に対応するものとなる。

学際的かつ多様性のある

危害・悪用・乱用評価は、多様な地理的場所、文化的背景、個人のアイデンティティから問題に関する実践的・技術的経験を持つ多分野の専門家と幅広いステークホルダーを包含する協働的取り組みであるべきである。

17.2.3. 評価

ハームズ・モデリングは、社会技術システムがユーザー、その他の利害関係者、あるいはより広範な社会に悪影響を及ぼす可能性、あるいは不公正な構造の創出・強化、人権への脅威、世界中の脆弱なグループに対する不均衡なリスクを生じさせる可能性を分析することに焦点を当てる。危害モデリングのプロセスでは、システムアーキテクチャとそのユーザーアフォーダンスに関する知見を、類似の既存システムが様々な社会集団に与えた影響に関する歴史的・文脈的証拠と体系的に統合するとともに、当該システムの影響を受ける可能性のある多様なコミュニティとの参加型協議を組み合わせることが求められる。この統合された情報が、危害を予測し、対応策を積極的に特定する能力の枠組みを構築する。

ハームズ・モデリングの文書では、フレームワークとこれまでに実施されたプロセスを説明した後、方法論、評価の概要、公開レビューとフィードバックのための概要、およびこれらの仕様のバージョン1.0、その実装、進化に付随して開発中のデューデリジェンス行動について述べています。

17.2.4. デューデリジェンス措置

危害・誤用・悪用評価は、C2PA技術仕様書および関連文書の開発に情報を提供しており、今後も提供し続けるべきである：

- 実装者向けガイダンス
- ユーザー体験ガイダンス
- セキュリティ上の考慮事項
- 解説書

さらに、危害・誤用・悪用評価は連合のガバナンスに反映され、C2PA仕様策定の原動力となったメディアへの信頼性、ユーザー制御、透明性における利益の最適化を推進する多様なC2PAエコシステムの促進に向けた潜在的な多国間協力の指針となるべきである。

第18章 C2PA標準アサーション

18.1. はじめに

本ドキュメントのこのセクションでは、C2PA実装で使用される標準アサーションのセットを列挙し、その構文や使用方法などを説明します。簡潔さを保つため、説明用の例としてすべてのJUMBF URIは短縮されていますが、実際のデータでは完全なURIが必要です。

すべてのアサーションは、第6.2節「ラベル」で説明されているラベルを持ち、第5章「バージョン管理」で説明されているようにバージョン管理されなければならない。
バージョン管理」で説明されているとおりにバージョン管理される。

すべてのC2PA標準化アサーションは、ISO 19566-5:2023に規定されるJSON JUMBFコンテンツタイプ、CBOR JUMBFコンテンツタイプ、または埋め込みファイルコンテンツタイプを使用します。エンティティ固有のアサーションは、これらいずれか、ISO 19566-5:2023 附属書 B に規定されるその他の JUMBF コンテンツタイプ（XML など）、または独自に作成したコンテンツタイプ（ISO 19566-5:2023 表 B.1 の指示に従う）を使用できます。Codestream コンテンツタイプは C2PA アサーションに使用してはなりません。

特に断りがない限り、この標準アサーションセットに文書化されたすべてのアサーションはCBORとしてシリアライズされるものとする。CBORとしてシリアライズされるすべてのアサーションは、CBORのコア決定的符号化要件（RFC 8949、4.2.1項参照）に準拠し、そのスキーマはCDDL定義を用いて定義されるものとする。

注記	すべてのCDDLは非規範的と見なされる。

JSONを使用して定義されるものについては、そのスキーマは最新バージョンのJSON Schemaを使用して定義されなければならない。

18.2. 関心領域

18.2.1. 説明

一部のユースケースでは、アクションアサーションなどの特定のアサーションが、アセット全体ではなく特定の部分のみに関連する場合があります。そのような場合、時間的、空間的、テキスト的、またはそれらの組み合わせである領域を記述する方法が必要です。領域定義はその目的を果たします。

18.2.2. 共通

領域定義において最も重要な部分は範囲フィールドであり、領域の時間範囲、空間範囲、フレーム範囲、テキスト範囲、またはそれらの組み合わせを記述するために使用される。

注記	仕様では範囲の組み合わせを指定できるが、マニフェストが 消費者がそれらを利用します。C2PAのユーザーエクスペリエンスタスクフォースが今後この件を扱うことが期待されています。

地域には以下の共通フィールドのいずれかを含む場合がある：

name

ユーザーインターフェースで使用される可能性のある、人間が読み取れる名称を表す自由形式の文字列。

identifier

このアサーションに固有の、機械可読な領域識別子を表す自由形式の文字列。

type

制御語彙（例：https://cv.iptc.org/newscodes/imageregiontype/）からの値、またはエンティティ固有の値（例：`com.litware.newType`）であり、領域が表す事象の種類を表す。

description

自由記述の文字列。

この仕様の旧バージョンには役割フィールドが含まれていました。このフィールドは非推奨となり、関心領域を生成する際には含めないものとします。

18.2.2.1. 範囲

すべての範囲は、値が「空間的」「時間的」「フレーム」「テキスト的」「特定済み」のいずれかであるタイプフィールドで構成される。さらに、その範囲の特定データを構成するオブジェクトである以下のいずれかのフィールドを含まなければならない：

- `shape`（空間用）；
- `time`（時間範囲用）；
- `frame`（時間的またはテキスト的範囲用）；
- `text`（テキスト用）；
- `item`（特定識別項目の場合）。

18.2.2.2. 空間

空間範囲は形状オブジェクトを用いて記述される。形状は矩形、円、多角形を表すために使用できる。これはIPTCの領域境界構造をモデルとしている。

18.2.2.3. 時間的

時間範囲は、開始時刻から終了時刻までの範囲を表す時間オブジェクトを用いて記述される。時刻は、RFC 2326（W3C Media Fragments仕様で推奨）で規定される標準再生時間（`npt`）を使用するか、RFC 3339で規定されるISO 8601のインターネットプロファイルに基づく「実時刻」を用いて記述される。

注記	「ウォールクロックタイム」は、ニュース放送やライブイベントなど、メディア資産が特定の日時に行われた活動を表す場合に有用です。

タイプフィールドが指定されていない場合、範囲はnpt形式であるとみなされる。開始フィールドが指定されていない場合、範囲はアセット

の先頭から開始する。終了フィールドが指定されていない場合、範囲はアセットの末尾で終了する。いずれのフィールドも指定されていない場合、範囲はアセット全体を表す。

指定されていない場合、範囲はアセット全体を表すものとみなされる。

18.2.2.4. フレーム

フレームオブジェクトは、開始フレームまたは開始ページと終了フレームまたは終了ページ（両端を含む）を用いて範囲を定義します。開始が指定されない場合、範囲はアセットの先頭から始まります。終了が指定されない場合、範囲はアセットの末尾で終了します。いずれも指定されない場合、範囲はアセット全体を表します。

フレームは単一の順序番号で表され、`0`が最初のフレームとなります。

フレームは通常、PDFなどの文書のページ番号を表すために使用されますが、アニメーション、動画、音声などの他のメディアタイプでも使用される場合があります。時間経過に伴う関心領域を扱うメディアタイプでは、可能な限り時間範囲を使用することが推奨されます。

18.2.2.5. テキスト

テキストオブジェクトは、W3C Web Annotationフラグメントセレクタで定義される1つ以上のURLフラグメント識別子を用いて範囲を定義します。開始文字と終了文字（両端を含む）へのオフセットを用いて範囲を精緻化することも可能です。開始が指定されない場合、範囲はフラグメントの先頭から始まります。終了が指定されない場合、範囲はフラグメントの末尾で終了します。いずれも指定されない場合、範囲はフラグメント全体を表します。

テキスト選択マップのフラグメントエントリは、単独で使用される場合、指定されたテキスト範囲全体を表します。ただし、テキスト選択範囲マップは、テキスト選択マップオブジェクトのペアをサポートします。`selector`の値は範囲の開始位置（終了エントリが存在しない場合は範囲全体）であり、`end`の値（存在する場合）は連続した範囲の終了位置を表します。さらに、複数のペアを使用して非連続な範囲を表現することも可能です。

18.2.2.6. 特定済み

アイテムオブジェクトは、アセット内のメディアトラック、メディアアイテム、またはその他の個別のコンテンツアイテムを定義し、クレーム生成者がアセットのファイルコンテナに格納されたコンテンツのサブセットのみに適用されるアサーションを示すことを可能にします。例えば、ビデオファイルのオーディオトラックのみが関連することを示すために使用できます。

メディアまたはコンテンツ項目は、識別子文字列によって特定されます。この値は、その特定のコンテナ形式における標準的な項目識別命名規則に一致する必要があります。例えば、MP4ファイルでは識別子の値は`track_id`、HEIFファイルでは`item_ID`であるべきです。識別子の値はvalueフィールドに提供されます。例えば、MP4動画ファイルコンテナにおいて識別子`track_id`で値2が指定された場合、ファイル内の2番目のメディアトラック（音声トラックなど）に関連するアサーションを示す。

特定範囲の識別子には、既知の意味的値による特定領域の指定用途もあります。例えば、Foundational Model of Anatomy（FMA）を用いて人体の特定部位を識別できます。この場合、識別子はスキーマの所在を示すURLまたはURIとします（機械可読形式への直接リンクである必要はありません）。

18.2.3. スキーマ

このタイプのスキーマは、以下のCDDL定義における`region-map`ルールによって定義されます：

```
region-map = {
      "region":            [1* $range-map],              ; 範囲の定義、1つ以上の範囲
      ? "name": tstr .size (1..max-tstr-length), ; ユーザーインターフェースで使用可能な、人間が読める地域名を表す自由形式の文字列。
      ? "identifier": tstr .size (1..max-tstr-length), ; このアサーションに固有の、機械可読な地域識別子を表す自由形式の文字列
。
      ? "type":       tstr .size (1..max-tstr-length),              ; 制御リストから
      ? "role":       $role-choice,        ; 非推奨
      ? "description": tstr .size (1..max-tstr-length), ; 領域の人間が読める説明
      ? "metadata": $assertion-metadata-map, ; アサーションに関する追加情報
}

$range-choice /= "spatial"               ; 物理領域で識別される範囲
$range-choice /= "temporal"              ; 期間によって特定される範囲
$range-choice /= "frame"                 ; 一連のフレームまたはページによって特定される範囲
$range-choice /= "textual"               ; テキストの範囲によって特定される範囲
$range-choice /= "特定された"              ; 特定の識別子と値によって識別される範囲

range-map = {
      "type":         $range-choice,            ; 「空間的」、「時間的」、「フレーム」、「テキスト的」または「特定済み」のいずれ
か
        ? "shape":    $shape-map,               ; 空間範囲の形状の説明
        ? "時間":      $time-map,                ; 時間範囲の時間境界の説明
        ? "frame":    $frame-map,               ; 時間範囲のフレーム境界の説明
範囲
      ? "text":       $text-map,                ; テキスト範囲の境界の説明
      ? "item":       $item-map,                ; 特定された範囲の境界の説明
}

座標マップ = {
      "x": float,          ; x軸に沿った座標 "y": float,   ; y軸に沿
      った座標
}

$shape-choice /= "rectangle"             ; 長方形の形状
$shape-choice /= "circle"                ; 円形
$shape-choice /= "polygon"               ; 多角形の形状

$unit-choice /= "pixel"             ; 単位はピクセル
$unit-choice /= "percent"           ; 単位は全体のサイズのパーセント


shape-map = {
      "type":         $shape-choice,                ; 「rectangle」、「circle」、または「polygon」のいずれか「
      unit」:         $unit-choice,                 ; "pixel" または "percent" のいずれか
      "origin":       $coordinate-map,              ; 図形の開始/原点座標。
      ? "width": float,                             ; 長方形の場合は幅、円の場合は直径（多角形では無視される）
      ? "height": float                             ; 長方形の高さ
      ? "inside" : bool,                            ; 形状の内側または外側、デフォルトは `true`
      ? "頂点数":           [1* $coordinate-map]          ; 多角形の残りの点/頂点
}
```

; npt と utc の開始時刻・終了時刻は異なる正規表現形式 time-map = npt-time-map / wall-clock-time-map

```
npt-time-map = {
      ? "type":       "npt"; 存在しない場合、"npt"タイムマップを想定
      ? "start": tstr .regexp "^(?:(?:([01]?\d|2[0-3]):)?([0-5]?\d):)?([0-5]?\d)(\.(\d{1,9}))?$",  ; 開
始時刻（存在しない場合はアセットの開始位置）。
      ? "end":                        ; 終了時刻（存在しない場合はアセットの終了位置）。
}

wall-clock-time-map = {
      "type":         "wallClock"; "wall-clock" タイムマップに必須
      ? "start": tstr .regexp "^(\d{4})-(\d{2})-(\d{2})T(\d{2}):(\d{2}):(\d{2})(\.\d+)?(([+-
]\d{2}:\d{2})|Z)$", ; 開始時刻（存在しない場合はアセットの開始時点）。
      ? "end":
]\d{2}:\d{2})|Z)$", ; 終了時刻（または存在しない場合はアセット/ライブエッジの終了位置）。
}

; これは動画のフレームまたは文書のページの両方に使用可能 frame-map = {
      ? "開始": int,             ; 開始フレーム（存在しない場合はアセットの先頭）。
      ? "end":        int        ; 終了フレーム（存在しない場合はアセットの終了位置）。
}

; これはW3C Web Annotationセレクタモデルを模倣したものですtext-selector-map = {
      "フラグメント":        tstr,      ; RFC3023 または ISO 32000-2 附属書 O に準拠したフラグメント識別子
      ? "start":           整数,       ; 開始文字オフセット（存在しない場合はフラグメントの先頭）
      ? "end":             整数        ; 終了文字オフセット（存在しない場合はフラグメントの終わり）。
}

; 範囲を特定するために使用される1つまたは2つのテキスト選択マップtext-selector-range-
map = {
      "selector":          $text-selector-map,             ; 開始（または唯一の）テキストセレクタ
      ? "end":             $text-selector-map              ; 存在する場合、テキスト範囲の終了を表す
}

text-map = {
      "selectors": [1* $text-selector-range-map] ; テキストの範囲の配列（連続しない場合あり）
}

item-map = {
      "identifier":        tstr .size (1..max-tstr-length),              ; アイテムを識別するために使用されるコンテナ固有の用
語。例: MP4 の「track_id」や HEIF の「item_ID」
      "value":             tstr .size (1..max-tstr-length),              ; 識別子の値。例: 識別子「track_id」の値「2」は、ア
セットのトラック2を意味する
}

; これらの値は非推奨です
$role-choice /= "c2pa.areaOfInterest" ; 特定すべき任意の領域
$role-choice /= "c2pa.cropped"                    ; クロップ処理後に残った領域全体
$role-choice /= "c2pa.edited"                     ; この領域には編集が適用されている
$role-choice /= "c2pa.placed"                     ; 材料が配置/追加された領域
$role-choice /= "c2pa.redacted"                   ; この領域内の何かが編集済み
$role-choice /= "c2pa.subjectArea"                ; 対象（人間か否か）固有の領域
$role-choice /= "c2pa.deleted"                    ; 情報の範囲が削除された
$role-choice /= "c2pa.styled"                     ; この領域にスタイルが適用された
$role-choice /= "c2pa.watermarked"                ; ソフトバインディングの目的でこの領域に透かしが適用されました
```

18.2.4. 例

CBOR診断表記法（RFC 8949、第8条）による一連の例を以下に示す：

```
// ビデオにおける時間的範囲と空間的範囲の組み合わせ例 //
{
   "region": [
      {
         "type": "temporal", "time":
         {
            "type": "npt",
            "start": "0",
            "end": "5.2"
         }
      },
      {
         "type": "spatial","shape":
         {
            "type": "rectangle",
            "unit":
            "pixel","origin": {
               "x": 10.0,
               "y": 10.0
            },
            "width": 200.0,
            "height": 112.0
         }
      }
   ],
   "name": "Animated Logo","identifier": "logo-
   clip",
   "description": "オープニングアニメーションロゴ（5.2秒間）。画面上部・左端から10ピクセル下、200px×112pxの矩形内に配置"
}

// Word/DOCXファイル内のテキスト範囲の例 //
{
   "region": [
      {
         "type": "textual", "text" :
         {
            "selectors" : [ [
                  {
                     "fragment" : "xpointer(/w:document/w:body/w:p/)"
                  }
               ]
            ]
         }
      },
   ],
   "description": "AIアシスタント編集済みコンテンツ"
}

// タグ付きPDFファイル内のテキスト範囲の例 //
{
   "region": [
      {
         "type": "textual", "text" :
         {
            "selectors" : [
```

```
            [
               {
                  "selector" : {
                     "fragment" : "path=/Document/Sect[0]/P[3]","start" : 10,
                     "end" : 20
                  }
               }
            ]
         ]
      }
   },
],
"description": "情報公開法（FOIA）に基づく開示請求に応じた黒塗り処理を実施"
}

// タグ付けされていないPDFファイルにおけるテキスト範囲の例 //
// この場合、ページと矩形領域のみ指定可能 //
{
   "region": [
      {
         "type": "textual", "text" :
         {
            "selectors" : [ [
                  {
                     "selector" : {
                        "fragment" : "page=1,rect=10,10,450,500","start" : 10,
                        "end" : 20
                     }
                  }
               ]
            ]
         }
      },
   ],
   "description": "情報公開法（FOIA）に基づく開示請求に応じた黒塗り処理を実施"
}

// PDFから一部のページを削除した例 //
{
   "region": [
      {
         "type": "frame", "frame" :
         {
            "start" : 27,
            "end" : 30
         }
      },
   ],
   "description": "配布前に不要なページを削除しました"
}

// Excelのセル範囲の例 //
{
   "region": [
      {
         "type": "textual", "text" :
         {
            "selectors" : [ [
                  {
```

```
                  "selector" : {
                     "fragment" : "xpointer(Sheet1!A5:A10)",
                  }
               }
            ],
            [
               {
                  "selector" : {
                     "fragment" : "xpointer(Sheet1!B5:B10)",
                  }
               }
            ]
         ]
      }
   },
],
"description": "Excelのセル範囲にスタイルを適用しました"
}

// テーブルセルの連続した範囲の例 //
{
   "region": [
      {
         "type": "textual","text" :
         {
            "selectors" : [ [
                  {
                     "selector" : {
                        "fragment" : "xpointer(//table[1]/tr[1]/td[2])",
                     },
                     "end" : {
                        "fragment" : "xpointer(//table[1]/tr[1]/td[4])",
                     }
                  }
               ]
                  ]
         }
      },
   ],
   "description": "テーブルのセルをいくつかクリアしました"
}

// 動画の特定トラックの範囲の例 //
{
   "region": [
      {
         "type": "temporal", "time":
         {
            "type": "npt",
            "start": "0",
            "end": "5.2"
         }
      },
      {
         "type":
         "identified","item": {
            "identifier": "track_id", "value":
            "2"
         }
      }
   ],
  "description": "オーディオトラックの一部を強化しました"
```

```
}


// アセット全体で目が変更されたことを指定する例 //
{
   "region" : [
      {
         "type" : "temporal",
         "time" : {},
      },
      {
         "type" : "identified","item" : {
            "識別子" : "https://bioportal.bioontology.org/ontologies/FMA","値" : "眼球の集合"
         },
      }
   ]
   "description": "会議中ずっと眠っているように見えるようにした"
}
```

18.3. アサーションに関するメタデータ

18.3.1. 説明

多くの場合、アサーションに関する追加情報を提供することが有用、あるいは必要となる。これには、アサーションが生成された日時や、アサーションデータの由来や真実性についてマニフェストコンシューマーが情報に基づいた判断を下すのに役立つその他のデータが含まれる。

	マニフェストコンシューマーはアサーションメタデータのどの部分も読み取る必要はありません。どの部分を読み取るかを選択できます。
注記	適用されるアサーションの種類に応じて、消費するフィールドを自由に選択できます。

以下に、他のアサーション内で使用されるコアスキーマを示します。

アサーションメタデータマップ規則に関するCDDLの定義は、アサーションメタデータ用CDDLに記載されています：

アサーションメタデータ用CDDL

```
;アサーションに関する追加情報を記述し、それへのハッシュ付きURI参照を含みます。ここではソケット/プラグを使用し、マップを同じファイル内で定義せずに個々のファイルでハッシュ付きURIマップが使用できるようにします
$assertion-metadata-map /= {
   ? "dateTime": tdate, ; アサーションが作成/生成された時点のRFC 3339形式の日時文字列
   ? "reviewRatings": [1* rating-map], ; アサーションに付与された評価（空の場合あり）
   ? "reference": $hashed-uri-map, ; このレビューの対象となる別のアサーションへのハッシュ付きURI参照
   ? "dataSource": source-map, ; 事前定義されたリストから選択された、アサーションデータのソースの説明
   ? "localizations" : [1* localization-data-entry] ; アサーション内の文字列のローカライズ
   ? "regionOfInterest" : $region-map ; このアサーションが関連するアセットの領域を記述
```

```
}

$source-type /= "signer"
$source-type /= "claimGenerator.REE"
$source-type /= "claimGenerator.TEE"
$source-type /= "localProvider.REE"
$source-type /= "localProvider.TEE"
$source-type /= "remoteProvider.1stParty"
$source-type /= "remoteProvider.3rdParty"
$source-type /= "humanEntry"
; 以下の2つのsource-typeの値は2.0以降非推奨となります
$source-type /= "humanEntry.anonymous"
$source-type /= "humanEntry.identified"


; 注記: この仕様の以前のバージョンには「actors」フィールドも含まれていましたが、バージョン2.0で削除されました。
source-map = {
   "type": $source-type, ; 列挙リストから選択される値。アサーションのソースが以下のいずれかを示す：リッチ実行環境（REE）で動作するクレーム生成器、信頼実行環境（TEE）で動作するクレーム生成器、REE内のローカルデータプロバイダー（例：モバイルOSの位置情報API）、TEE内で動作するローカルデータ（例：チップセットベンダー提供の信頼された位置情報アプリ）、サーバーなどのリモートデータプロバイダー（例：Googleの地理位置情報APIサービス）、または人間による入力のいずれかを示す列挙型リストからの値。
   ? "details": tstr .size (1..max-tstr-length), ; アサーションデータのソースに関する詳細を記述した人間が読める文字列。例：データを提供したリモートサーバーのURL
}

int-range = 1..5

$review-code /= "actions.unknownActionsPerformed"
$review-code /= "actions.missing"
$review-code /= "actions.possiblyMissing"
$review-code /= "depthMap.sceneMismatch"
$review-code /= "ingredient.modified"
$review-code /= "材料.変更の可能性あり"
$review-code /= "thumbnail.primaryMismatch"

; 以下の3つのレビューコード値はバージョン2.0以降非推奨となります
$review-code /= "stds.iptc.location.inaccurate"
$review-code /= "stds.schema-org.CreativeWork.misattributed"
$review-code /= "stds.schema-org.CreativeWork.missingAttribution"

rating-map = {
   "value": int-range, ; "アイテムの評価値（1：最低〜5：最高）"
   ? "code": $review-code, ; 評価理由を説明するラベル形式の文字列
   ? "説明": tstr .size (1..max-tstr-length), ; 評価理由を説明する人間が読める文字列
}

; ローカライゼーション辞書を格納するために使用されるデータ構造
$localization-data-entry /= {
   * $$language-string
}

language-string /= tstr .size (1..max-tstr-length)
```

CBOR診断表記法（RFC 8949、第8条）の例：

```
{
   "reference": {
         "url": "self#jumbf=c2pa.assertions/c2pa.metadata", "alg":
         "sha256",

      },
   "dataSource": {
      "type": "localProvider.REE",
      "details": "オペレーティングシステムの位置情報APIによって提供されるEXIF GPSデータ"
   }
}
```

ほとんどの場合、このアサーション固有のメタデータは、他のアサーション（例：成分）のメタデータフィールドの値として直接内部に表示されます。ただし、サムネイルのようにアサーションがJSONやCBORでない場合など、アサーションメタデータを独立した個別のアサーションメタデータアサーションに保存する必要がある、または望ましい場合があります。

アサーションメタデータアサーションのラベルは `c2pa.assertion.metadata` です。

18.3.2. データソース

このdataSourceフィールドはオプションフィールドであり、クレーム生成者が下流のマニフェストコンシューマーに対し、アサーションの内容がどのソースから生成されたかを通知することを可能にします。特定のアサーションに対して`dataSource`が提供されない場合、`dataSource`は署名者とみなされます。

> **注記**
>
> デフォルトでは、すべての作成済みアサーションは署名者に帰属します。これは、信頼モデルが署名者（通常はクレーム生成者でもある）への信頼を基盤としているためです。

このフィールドの値は、`type` と `details` の 2 つのフィールドで構成される `dataSource` オブジェクトです。

dataSource typeフィールドは、データソースのタイプを定義します。これは、セクション6.2「ラベル」で説明されている形式のラベルで構成されます。値は、表5「データソースタイプ」に定義された仕様定義値のいずれか、または拡張メカニズムとしてエンティティ固有の名前空間を使用できます。

表5. データソースの種類

値の型	意味
署名者	アサーションの内容は署名者から提供された
`claimGenerator.REE`	アサーションの内容は、デスクトップやモバイルオペレーティングシステムなどのリッチ実行環境（REE）で動作するクレームジェネレータから提供された
`claimGenerator.TEE`	アサーションの内容は、信頼されたOSなどの信頼された実行環境（TEE）で動作するクレームジェネレータから提供されました

localProvider.REE	アサーションの内容は、クレーム生成器と同じ物理コンピューティングデバイス上で実行されているREE内のデータソースから取得されました
値の型	**意味**
localProvider.TEE	アサーションの内容は、クレーム生成器と同じ物理コンピューティングデバイス上の TEE で実行されているデータソースから取得されました
remoteProvider	アサーションの内容は、署名者またはクレーム生成ベンダーが管理するリモートデータソースから取得された
remoteProvider.external	アサーションの内容は、署名者またはクレーム生成ベンダー以外の、外部リモートデータソースから取得されました
humanEntry	アサーションの内容は人間が入力したものです

詳細フィールドは、dataSource に関する追加情報（アサーション内容の提供に使用された API の名前、内容の提供元サーバーの URL など）を記述する、人間が読める文字列です。たとえば、広域の位置情報アサーションソースの場合、`type` 値は `remoteProvider.3rdParty`、`details` 値は`www.googleapis.com/geolocation/v1/geolocate` と設定されます。

18.3.3. レビュー評価

reviewRatings配列が存在する場合、クレーム生成者はアサーションの品質（またはその欠如）に関する1つ以上の評価オブジェクトを提供する場となります。`dataSource`オブジェクトが存在し、そのtypeフィールドの値が`humanEntry.anonymous`または`humanEntry.credentialed`のいずれかである場合、`reviewRatings`は存在してはなりません。

評価オブジェクトの値フィールドは、*1*（最低）から*5*（最高）までの整数値を必ず含むものとする。説明フィールドが存在する場合、評価の種類に関する人間が理解可能な文字列記述を含むものとする。さらに、アサーション固有の評価結果コードを定義するオプションの機械可読コードフィールドを提供できる。コードフィールドの値は、セクション 6.2「ラベル」で記述された形式を用いて定義される。値は、表6「コードフィールドの値」に示す仕様定義値のいずれか、または拡張メカニズムとしてエンティティ固有の名前空間を使用できる。

表6. コード フィールドの値

コードの値	適用可能なアサーション	意味
actions.unknownActionsPerformed	c2pa.actions	アクションアサーションには、オーサリングツールで実行されたすべてのアクションの完全なリストが含まれていません（例：オーサリングツールがその効果を認識していないサードパーティ製フィルターの使用による）。
actions.placedIngredientNotFound	c2pa.actions	レビュー対象のアクションアサーションには、解決可能な成分 URI を持たない配置済みアクションが含まれています。値は `1` である必要があります。

ingredient.action 欠落	c2pa.ingredient	審査対象の成分アサーションには、そのクレーム内で参照するアクションが少なくとも1つ存在しません。値は1である必要があります。
ingredient.notVisible	c2pa.ingredient	審査対象の成分アサーションは、そのマニフェストに紐付けられたデジタルコンテンツ内で表示されていません。値は 1 である必要があります。
コードの値	**適用可能なアサーション**	**意味**
depthMap.sceneMis match	c2pa.depthmap.GDe pth	深度マップアサーションの内容が、アセットの主要なプレゼンテーションで描かれているシーンに対応していません（例：ピクチャー・オブ・ピクチャー攻撃による）。
thumbnail.primary 不一致	c2pa.thumbnail.cl aim	サムネイルの内容が、アセット内のプライマリプレゼンテーションの内容と一致しません。

18.3.4. 参照

アサーションメタデータアサーションの参照フィールドは標準のハッシュドURIであるため、アクティブなマニフェスト以外のマニフェスト内のアサーションを参照することも可能です。例えば、アクティブなマニフェストには、成分のマニフェスト内に存在するc2pa.metadataアサーションを検証するアサーションメタデータアサーションが含まれる場合があります。

> **注記**
> クレームはアサーションの特殊なタイプであるため、この同じ手法を用いて他のマニフェスト内のクレームを参照することも可能です。

18.3.5. DateTime

dateTime フィールドが存在する場合、その値は CBOR 日付/時刻 (RFC 8949, 3.4.1) に準拠した日付時刻文字列でなければなりません。

18.3.6. 関心領域

このアサーションは、資産の一部（例えば、動画の特定のフレーム範囲や画像上の特定領域）にのみ適用される場合があります。このような部分は、regionOfInterestフィールドを使用して特定できます。このフィールドの値は、領域マップオブジェクト（セクション18.2「関心領域」で定義）です。

18.3.7. ローカライゼーション

18.3.7.1. 一般

C2PAマニフェストの利用者は、可能な限り母国語で情報を理解できることが重要です。この目的のため、アサーションのメタデータに辞書を含めることで、アサーションのローカライズ情報を追加することが可能です。

18.3.7.2. ローカライゼーション辞書

ローカライゼーション辞書は単一のオブジェクトで構成され、その各キーはJSON-LDの言語インデックス技術を用いた翻訳を表す。翻訳が必要な値が最上位キーに関連付けられていない場合、オブジェクト内にネストされたキーを参照するには「ドット表記」(.)を使用する。配列内の特定の要素を参照する必要がある場合は、配列インデックス表記(`[n]`, $n \geq 0$)を使用する。翻訳が必要な値自体が配列である場合、特定の要素を参照できる。例4「ローカライゼーション

辞書」に例を示します：

例4. ローカライゼーション辞書の例

```
{
  "dc:title": {
    "en-US": "ケビンの五匹の猫",
    "en-GB": "Lord Kevin's Five Cats", "es-MX":
    "Los Cinco Gatos de Kevin", "es-ES": "Los Cinco
    Gatos de Kevin", "fr": "Les Cinq Chats de
    Kevin",
    "jp": "ケヴィンの５匹の猫"
  }
}
```

```
{
  "actions[0].softwareAgent": { "en-US":
    "Joe's Photo Editor", "en-GB": "Joe's
    Photo Editor",
    "es": "ジョーの写真編集者", "fr": "ジョーの写真編集
    者",
    "jp": "ジョーの写真編集者"
  }
}
```

このようなサードパーティのキーまたは値は、セクション6.2.1「名前空間」と同様の方法で名前空間を付与する必要があります。

例: `com.litware`。マニフェストコンシューマーがこれらのキーと値に関する人間が読める情報を表示するためには、クレームジェネレータはこのローカライゼーション手法を通じて文字列を提供すべきである。

ローカライズされたアクションは、カスタムアクションのローカライズにおける使用例を示しており、

`c2pa.actions` アサーション。

ローカライズされたアクション

```
{
  "com.litware.blur": { "en-
    US": "Blur",
    "fr-FR": "Brouiller",
  },
  "com.litware.filter": { "en-
    US": "Filter",
    "es-ES": "Filtrar",
    "jp-JP": "フィルター"
  }
}
```

18.4. 標準C2PAアサーションの概要

標準C2PAアサーションは表7「標準C2PAアサーション」に列挙されている：

表7. 標準C2PAアサーション

タイプ	アサーション	スキーマ	シリアライゼーション
アクション	c2pa.actions、c2pa.actions.v2	C2PA	CBOR
アサーションメタデータ	c2pa.assertion.metadata	C2PA	CBOR
資産参照	c2pa.asset-ref	C2PA	CBOR
Asset Type	c2pa.asset-type (非推奨)、c2pa.asset-type.v2	C2PA	CBOR
BMFFベースのハッシュ	c2pa.hash.bmff (削除済み)、c2pa.hash.bmff.v2 (非推奨)、c2pa.hash.bmff.v3	C2PA	CBOR
証明書ステータス	c2pa.certificate-status	C2PA	CBOR
クラウドデータ	c2pa.cloud-data	C2PA	CBOR
コレクションデータハッシュ	c2pa.hash.collection.data	C2PA	CBOR
データハッシュ	c2pa.hash.data	C2PA	CBOR
深度マップ	c2pa.depthmap.GDepth	https://developers.google.com/depthmap-metadata/reference	CBOR
組み込みデータ	c2pa.embedded-data	C2PA	JUMBF 埋め込みファイル
フォント情報	font.info	C2PA	CBOR
General Box Hash	c2pa.hash.boxes	C2PA	CBOR
成分	c2pa.ingredient、c2pa.ingredient.v2、c2pa.ingredient.v3	C2PA	JUMBF 埋め込みファイル
メタデータ	c2pa.metadata	C2PA	JSON-LD
マルチアセットハッシュ	c2pa.hash.multi-asset	C2PA	CBOR
ソフトバインディング	c2pa.soft-binding	C2PA	CBOR
サムネイル	c2pa.thumbnail.claim (クレーム作成時間)、c2pa.thumbnail.ingredient (成分のインポート)	C2PA	埋め込みファイル
タイムスタンプ	c2pa.タイムスタンプ	C2PA	CBOR

18.5. データハッシュ

18.5.1. 説明

非BMFFベースの資産の一部について、その完全性を一意に検証する最も一般的な方法は、データハッシュアサーションに含まれるハードバインディング（すなわち暗号ハッシュ）を介して行われます。ただし、BMFFと互換性のない「ボックスのような」形式については、一般的なボックスハッシュアサーションの使用が推奨されます。

データハッシュアサーションは、セクション13.1「ハッシュ処理」に記載されているハッシュの作成と保存をサポートし、その値はハッシュフィールドに存在しなければならない。

各データハッシュアサーションは、ハッシュが計算されたバイト範囲を定義する。資産の一部のみをハッシュ化する場合、除外される範囲は`exclusions`フィールドの配列値に存在しなければならない。これらの除外範囲は開始位置の昇順で並べられ、重複してはならない。

データハッシュ除外範囲については、範囲は同一の論理単位（例：ボックス、セグメント、オブジェクト）内で開始および終了し、フリーボックスまたはパットデータを除き、当該単位に関連付けられたヘッダーまたは長さのフィールドと重複してはならない。攻撃者がこれらの範囲に配置する可能性のあるいかなるデータも、アセットの解釈に実質的な影響を与えないことを保証する方法で除外範囲を定義することは、クレーム生成者の責任である。さらに、クレーム生成者は除外範囲がC2PAマニフェストストアまたは資産メタデータ（例：EXIF、IPTCメタデータ）からの内容のみを含むことを保証しなければならない。スキップ可能なメタデータの例としては、検証されていないユーザー名や画像回転情報などが挙げられる。

この仕様の以前のバージョンでは、ハッシュ化されたデータの所在を示すポインタを提供する`URL`フィールドが用意されていたが、実際には使用されなかった。このフィールドは現在、アセット参照アサーションに取って代わられ非推奨となっている。クレーム生成者はデータハッシュアサーションにこのフィールドを追加してはならず、消費者は存在する場合このフィールドを無視しなければならない。ただし、セクション15.10.3「アサーション検証」で説明される検証対象コンテンツの一部としてのフィールド包含には影響しない。

データハッシュアサーションのラベルは`c2pa.hash.data`とする。データハ

ッシュアサーションはクラウドデータアサーション内に存在してはならない

。

データハッシュアサーションは圧縮マニフェストと併用してはならない。

> **注記** この制限は、両者の技術的な非互換性に対処するために設けられています。

18.5.2. スキーマと例

このタイプのスキーマは、以下のCDDL定義における`data-hash-map`ルールによって定義されます：

```
; ハッシュマップ内のオプション性も確認すること
; アセットのデータの一部または全部の暗号ハッシュを格納するために使用されるデータ構造

; およびハッシュ計算に必要な追加情報。data-hash-map = {
   ? "除外": [1* EXCLUSION_RANGE-map], ; 範囲は単調増加する `start` 値を持ち、2つの範囲が重複してはならない。
   ? "alg":tstr .size (1..max-tstr-length), ; このアサーションでハッシュを計算するために使用される暗号ハッシュアルゴリズムを識別する文字列。C2PAハッシュアルゴリズム識別子リストから取得される。このフィールドが存在しない場合、ハッシュアルゴリズムは
```

外側の構造体の`alg`値が採用される。両方が存在する場合、この構造体のフィールドが使用される。いずれの場所にも値が存在しない場合、この構造体は無効となる。デフォルト値は存在しない。

```
   "hash": bstr, ; ハッシュ値のバイト文字列
   "pad": bstr, ; スペースを埋めるために使用されるゼロ埋めバイト文字列
   ? "pad2": bstr, ; スペースを埋めるために使用されるオプションのゼロ埋めバイト文字列
   ? "name": tstr .size (1..max-tstr-length), ; (オプション) このハッシュがカバーする内容の人間が読める説明
   ? "url": uri, ; 未使用かつ非推奨。
}

EXCLUSION_RANGE-map = {
   "start": uint, ; 範囲の開始バイト位置 "length": uint, ; 除外するデータバイト数
}
```

CBOR診断表記法（RFC 8949、第8条）の例を以下に示す：

```
{
   "alg" : "sha256",
   "pad" : '0000',
   "hash": 'Auxjtmax46cC2N3Y9aFmBO9Jfay8LEwJWzBUtZ0sUM8gA=',"name": "JUMBF
   manifest"
   "exclusions": [
      {
         "start": 9960,
         "length": 4213
      }
   ],
}
```

通常、除外範囲の開始位置と長さの値は、推奨されるシリアライズ形式（すなわち「可能な限り短い形式」）で記述される。ただし、データハッシュアサーションを作成する必要があるが開始位置と長さの値がまだ不明な場合、それらは「可能な限り大きい形式」で作成される。これは32ビット整数として扱われる。

pad値は常に存在する必要があるが、複数パス処理中にバイトを置換（すなわち「パディング」）するために使用されない限り、長さ0のゼロ埋めバイト列とする。`pad2`はオプションのゼロ埋めバイト列であり、`pad`で目的のパディングが達成できない場合に使用される。

注記

セクション 10.4「複数ステップ処理」では、正しい値の記入方法とパディングの調整方法について説明しています。

18.5.3. JPEG 1 に関する特別な考慮事項

JPEG 1 (.jpg) ファイルをハッシュ処理し、C2PA マニフェストを埋め込む場合、APP11 マーカー (`FFEB`) および JUMBF データを含むすべての APP11 セグメントの長さ (`Lp`) は除外範囲に含めること。

注記

C2PA マニフェスト JUMBF を含むすべての APP11 セグメントは連続しているため、単一の範囲のみが必要です。

18.5.4. PNGに関する特別な考慮事項

C2PA マニフェストが埋め込まれる PNG (.png) ファイルをハッシュする場合、JUMBF データを含むチャンクの「長さ」および「caBX」（チャンクタイプを表す）を除外範囲に含めることが重要です。

18.6. BMFF ベースのハッシュ

18.6.1. 説明

クレーム生成者がハードバインディング（すなわち暗号ハッシュ）で一意に識別したいBMFFベース資産の部分は、BMFFベースのハッシュアサーションを用いて記述されるものとする。

BMFFベースのハッシュアサーションは、`c2pa.hash.bmff.v3`というラベルを持つものとする。

注記 この規格の以前のバージョンでは、`c2pa.hash.bmff` および `c2pa.hash.bmff.v2` アサーションも記載されていた。

重要 バリデータは、`c2pa.hash.bmff` アサーションを無視し、アサーションが存在しないかのようにマニフェストを処理しなければならない。

BMFF ベースのハッシュアサーションは、クラウドデータアサーションには出現してはならない。

この仕様の以前のバージョンでは、ハッシュ化されたデータの所在を示すポインタを提供する`url`フィールドが用意されていたが、実際には使用されなかった。このフィールドは現在、アセット参照アサーションに取って代わられ非推奨となっている。クレーム生成者はBMFFハッシュアサーションにこのフィールドを追加してはならず、消費者は存在する場合このフィールドを無視しなければならない。ただし、セクション15.10.3「アサーション検証」で説明される検証対象コンテンツの一部としてのこのフィールドの包含には影響しない。

18.6.2. ハッシュ計算

BMFFハッシュの値フィールドで指定されたハッシュを計算するには、除外配列内のいずれかの除外エントリに一致するBMFFボックスまたはそのサブセットを除き、ファイルの全バイトをハッシュに加算する。

ハッシュ生成に用いられる入力データには、ボックス全体が包含される場合、そのボックスヘッダーも含まれる。同様に、ボックス全体が除外される場合、そのボックスヘッダーもハッシュ生成用入力データから除外される。除外仕様のサブセットフィールドを使用してハッシュ生成用入力データからボックスが部分的に除外される場合、サブセットフィールド内の相対バイトオフセットで定義される除外対象のボックス部分は、ボックスヘッダーを含むボックスの先頭からのオフセットであり、ボックスコンテンツの先頭からのオフセットではない。これらのサブセット範囲はオフセット値の昇順で並べられ、重複してはならない。

`c2pa.hash.bmff.v2`（非推奨）および `c2pa.hash.bmff.v3` アサーションにおいて、その全体が除外されていないルートボックスについては、そのボックスのハッシュ生成に寄与する入力データは、バイナリ文字列の連結 `offset || data` で構成される。ここで `offset` は、ボックスの絶対ファイルオフセットをビッグエンディアン形式の 8 バイト整数として定義し、`data` はヘッダーを含むボックスの内容から除外さ

れた部分を差し引いたものを定義する。この定義において、「||」

は両者のバイナリ連結を表す。bmff-hash-mapがハッシュフィールドとマークルフィールドの両方を含む場合、マークルツリーハッシュではオフセットを含めてはならない。

さらに、`c2pa.hash.bmff.v2`（非推奨）および`c2pa.hash.bmff.`v3アサーションには以下の機能が含まれる：

- ハッシュに組み込まれるルートボックスの入力データ先頭には、絶対ファイルバイトオフセットが含まれます。これにより、ハッシュに含まれるルートボックスがファイル内で位置を変更できないことが保証されます。
- bmff-hash-mapがハッシュとマークル
  フィールドの両方を含む場合、mdatボックスは完全に除外されません。代わりに、除外リスト上の必須エントリによってボックスの大部分が除外されます。

> **注記**
> これらの2つの機能により、mdatがファイル内で位置を変更できないことが保証されると同時に、bmff-hash-mapがハッシュフィールドとマークルフィールドの両方を含む場合、個々のマークルツリーハッシュに対するオフセットの必要性を排除します。

ボックスが除外配列内の除外エントリと一致するのは、以下のすべての条件が満たされる場合に限る：

- ボックスのファイル内での位置は、除外マップエントリのxpathフィールドと一致します。例えば、除外xpath `/foo/bar[2]` は に一致します 次の位置に一致します `/foo[3]/bar[2]` および `/foo[2]/bar[2]` に一致します。 しかし ただし `/foo[3]/bar[1]` や `/foo[3]/bar[2]/baz[1]` は除外対象外です。
- 長さ（`length`）が除外マップエントリで指定されている場合、ボックスの長さは除外マップエントリの
  エントリの長さフィールドと完全に一致します。注記: 長さにはボックスヘッダーが含まれます。
- exclusions-map エントリで `version` が指定されている場合、ボックスは FullBox であり、ボックスの version は `exclusions-map` エントリの `version` フィールドと完全に一致します。
- フラグ（正確に3バイトのバイト配列）が除外マップエントリで指定され、ボックスがFullBoxである場合。exactがtrueに設定されているか指定されていない場合、ボックスのフラグ（bit(24)、つまり3バイト）も除外マップエントリのフラグフィールドと完全に一致します。`exact` が false に設定されている場合、ボックスのフラグ（bit(24)、つまり 3 バイト）と除外マップエントリのフラグフィールドのビット単位の AND が、除外マップエントリのフラグフィールドと完全に一致します（つまり、ボックスには少なくともそれらのビットが設定されていますが、追加のビットが設定されている場合もあります）。
- exclusions-map エントリで `data` (オブジェクトの配列) が指定されている場合、配列の各項目について、ボックスのバイナリデータがその項目の相対バイトオフセットフィールドで、その項目の `bytes` フィールドと完全に一致する。

`xpath` フィールドの文字列構文は、以下の厳密な部分集合に制限されるものとする。

- 省略形構文のみを使用すること。
- 完全なパスのみを使用すること。
- ノードまたはノード[整数]によるノード選択のみを使用すること。
- 子孫構文、すなわち `//` は使用してはならない。

- すべてのノードはBMFF `4cc`コードとする。

例5. `xpath` フィールドの完全な構文

```
xpath = '/' nodes nodes =
node
        | node '/' nodes node =
box4cc
       | box4cc '[' 整数 ']'
```

注記:

`box4cc` は、`ISO/IEC 14496-12` で `BMFF` ボックスに許可されている任意の `4cc` です。`integer` は、先頭のゼロを含まない任意の非ゼロの正の整数です。

任意の除外エントリは、ゼロ個以上のボックスに一致する可能性があります。除外エントリが正確に1つのボックスに一致する必要はありません。

非リーフのxpathノードは、自身のフィールドを持たない（つまりデータを含まず子ボックスのみを含む）かつFullBoxから継承しないコンテナボックスのみを指すものとします。これにより、C2PAバリデータは他のボックスを含む特殊なボックスの構文や意味論を認識する必要がなくなります。このような特殊なボックスの子ボックスを全体または一部除外する必要がある場合、`exclusions-map`エントリのxpathフィールドは特殊なボックス自体を指し、subset-mapフィールドは除外対象の子ボックスデータを含むバイト範囲を除外する。例えば、「`sgpd`」ボックスは他のボックスを含みますが、FullBoxから継承するという点で特殊です。したがって、「`sgpd`」から子ボックスを全体または一部除外する必要がある場合、アサーションは「`sgpd`」自体を指すxpathフィールドを使用します（ 例：`/moof/traf/sgpd`) を使用し、サブセットマップフィールドを用いて目的のバイトを除外する。

C2PA マニフェストがファイルに埋め込まれている場合、それを含むボックスは除外配列のエントリの 1 つでなければならない配列のエントリの一つである。詳細はセクションA.5「BMFFベースのアセットへのマニフェスト埋め込み」を参照のこと。

C2PAマニフェスト作成後に非ルート除外ボックスが削除された場合、他のボックスのハッシュ生成に寄与した入力データが無効化されないよう、同サイズの「空き」ボックスで置き換えること。C2PAマニフェスト作成後に圧縮マニフェストを使用してC2PAマニフェストの保存サイズを縮小する場合、オフセットが維持されるよう、その位置に「空き」ボックスを挿入すること。C2PAマニフェスト作成後に非ルート除外ボックスが追加される可能性がある場合、マニフェスト作成時に除外ボックス分の十分な空き容量を持つ「空き」ボックスを挿入し、その「空き」ボックスも完全なXPathを用いた除外エントリで除外対象とする。除外ボックスが追加された場合、またはC2PAマニフェストストアのサイズが増大した場合、追加されたデータを補償するために「空き」ボックスは縮小（または削除）される。ただし「空き」ボックスに十分なスペースがない場合、標準マニフェストを使用する。

C2PAデータをMP4ボックスを介してBMFFベースのアセットに埋め込むと、他のMP4ボックス内のファイルオフセットや、ルートボックスのハッシュ生成に使用される入力データに含まれる絶対ファイルバイトオフセットも変更されます。これらのボックスとオフセットは、埋め込み前の値ではなく埋め込み後の値でハッシュ生成に使用される入力データに含める必要があります。そうしないと、BMFFベースのハッシュアサーションが検証されません。

実装において、すべてのファイルバイトオフセットの埋め込み後値が確実にハッシュ化される方法は以下の3つです：

1. 「フリー」ボックスを使用する。

a. 埋め込むC2PAボックスの妥当な最大サイズを決定する。C2PA対応の全MP4ボックスは末尾に未使用のパディングバイトをサポートするため、「空き」ボックスのサイズを過大評価しても問題ない。余分なバイトは無視されるためである。

b. 指定サイズに相当する「フリー」ボックスをアセットファイルに挿入し、全てのオフセットを適切に更新する。

c. 除外リストに「/free」を含めてアセットのハッシュ処理を実行する。

d. マニフェストを作成し署名します。C2PAボックスを作成します。

e. C2PAボックスで「フリー」ボックスを上書きします。

2. 2 パス方式を使用します。

a. BMFF ベースのハッシュアサーションおよびマークルボックス（存在する場合）の正確なサイズを計算する。後者では、マークルツリーのサイズを決定するためにアセットファイルを解析する必要がある。

b. 最終マニフェストの正確なサイズを計算する。

c. アセットファイルのハッシュ処理を実行する。ハッシュ生成に用いる入力データに含める前に、ファイルオフセットを含むボックスをすべて正しい値に更新する。上記で説明した更新済み絶対ファイルオフセットを用いて、(オフセット || データ）を使用しハッシュ生成に用いる入力データを計算する。上記で示した通り、bmff-hash-map がハッシュフィールドとマークルフィールドの両方を含む場合、マークルツリーハッシュ生成に用いるデータにはオフセットは含まれない。

d. マニフェストを作成し署名する。C2PAボックスを作成する。

e. C2PAボックスを挿入する。

3. 更新されたマニフェストストアをBMFFファイルの末尾に配置してください。

a. マニフェストの `box_purpose` をオリジナルに設定します。

b. マニフェストを作成し署名します。

c. `box_purpose` を `update` に設定した C2PA ContentProvenanceBox を作成します。

d. 更新済みマニフェストを更新済みマニフェストとしてC2PAコンテンツプロバンスボックスに挿入します。

e. C2PA ContentProvenanceBoxをBMFFファイルの末尾に挿入します。

f. 更新マニフェストストアが存在する場合に標準マニフェストが追加されると、更新マニフェストストアの内容は「オリジナル」マニフェストに移動されます。

g. 更新されたマニフェストストアはファイル末尾から削除され、一般的なユースケースでは単一のマニフェストによる下位互換性が確保されます。

h. 「オリジナル」マニフェストストアの`box_purpose`は`manifest`に戻され、標準マニフェストは通常通り追加されます。

注記

box_purposeフィールドはハッシュに含まれず、マニフェストを無効化することなく変更可能です。
既存のハッシュを無効化しません。同様に、新しいC2PA ContentProvenanceBoxを追加しても既存のハッシュは無効化されません。

2パス方式は複雑さが格段に増すものの、最大顕在サイズの事前知識なしに正しいハッシュ処理を実現する。

マニフェストの最大サイズを事前に知る必要なく正しいハッシュ処理を可能にします。また最終的なアセットサイズを最小化します。ファイルオフセットを持つ一般的なボックス（網羅的**ではない**）には'iloc'、'stco'、'co64'、'tfhd'、'sidx'、'saio'などがあります。

更新されたマニフェストをBMFFファイル末尾に配置するオプションは、十分な空きボックスが存在しない場合や、2パス方式の複雑さを避けたい場合に更新を可能にします。このオプションは、部分的なデータオフセット情報を持つアトム「stco」ボックス内のチャンクオフセットもサポートします。

18.6.3. スキーマと例

c2pa.hash.bmff.v2（非推奨）およびc2pa.hash.bmff.v3アサーションのスキーマは、以下のCDDL定義における以下のCDDL定義内のbmff-hash-mapルールによって定義されます：

```
bmff-hash-map = {
   "exclusions": [1* exclusions-map],
   ? "alg": tstr .size (1..max-tstr-length), ; このハッシュを計算するために使用される暗号ハッシュアルゴリズムを識別する文字列。C2PAハッシュアルゴリズム識別子リストから取得される。このフィールドが存在しない場合、ハッシュアルゴリズムは包含構造体で定義されたものから取得される。両方が存在する場合、この構造体のフィールドが使用される。いずれの場所にも値が存在しない場合、この構造体は無効となる。デフォルト値は存在しない。
   ? "hash": bstr, ; 非フラグメント化MP4の場合、これは除外配列にリストされたボックスを除いたBMFFファイル全体のハッシュである。フラグメント化MP4の場合、このフィールドは存在してはならない。
   ? "merkle": [1* merkle-map], ; 単一の'mdat'ボックス、複数の'mdat'ボックス、および/またはアセット内の個々のフラグメントファイルの検証を可能にするために必要な、Merkleツリーの行と関連データの集合。
   ? "name": tstr .size (1..max-tstr-length), ; オプション） このハッシュがカバーする内容について、人間が読める説明。
   ? "url": uri, ; 未使用かつ非推奨。
}

;（オプション） 厳密に3バイトのCBORバイト文字列。 flag-type = bytes

flag-t = flag-type .eq 3

exclusions-map = {
   "xpath": tstr, ; ルートノードから始まるハッシュから除外するボックスの位置を、https://www.w3.org/TR/xpath-10/バージョンのxpath形式文字列で指定。構文は厳しく制限される。
   ? "length": uint, ; （オプション） ハッシュから除外するために最下位ボックスが持つべき長さ。
   ? "data": [1* data-map], ; （オプション） 指定された相対バイトオフセットの最も末端のボックス内のデータは、指定されたデータと同一でなければならず、そうすることでそのボックスはハッシュから除外される。
   ? "subset":[1* subset-map], ; （オプション） ハッシュから除外されるのは除外ボックスのこの部分のみ。配列の各エントリは単調増加する相対バイトオフセットを持つ必要がある。配列内のサブセットは重複してはならない。最後のエントリは長さがゼロでもよい。これは、その相対バイトオフセット以降のボックスの残りが除外されることを示す。ボックスの長さを超える相対バイトオフセットまたは相対バイトオフセットと長さの合計は許可される。ボックスの末尾を超えるバイトは決してハッシュ化されない。
   ? "version": int, ; （オプション） ハッシュから除外される葉ノード最上位ボックスに設定必須のバージョン。FullBoxを継承するボックスにのみ指定される。
   ? "flags": flag-t, ; （オプション） 厳密に3バイトのバイト文字列。ハッシュから除外されるために最下位ボックスに設定が必要な24ビットフラグ。FullBoxを継承するボックスにのみ指定される。
   ? "exact": bool, ; （オプション） flags が完全一致する必要があることを示す。指定しない場合、デフォルトは true。FullBox を継承するボックスに対してのみ指定され、かつ
```

```
flags も指定されている場合にのみ指定される。
}

data-map = { "offset":
   uint, "value" : bstr,
}
subset-map = { "offset":
   uint, "length": uint,
}

; マップの各エントリは、単一の検証を可能にするために必要な、Merkle ツリーの行と関連データです。
; アセット内の「mdat」ボックスまたは複数の「mdat」ボックス。", merkle-map = {
   "uniqueId": int, ; ファイル間で区別し、特定の'mdat'ボックスを検証するために使用するMerkleツリーを決定するための一意のID（1から始まる）。
   "localId": int, ; マークルツリーを示すローカルID。
   "count": int, ; マークルツリー内のリーフノードの数。ヌルノードはこのカウントに含まれない。
   ? "alg": tstr .size (1..max-tstr-length), ; このマークルツリー内のハッシュを計算するために使用される暗号ハッシュアルゴリズムを識別する文字列。C2PAハッシュアルゴリズム識別子リストから取得される。このフィールドが存在しない場合、ハッシュアルゴリズムは包含構造体が定義する `alg` 値が使用される。両方が存在する場合、この構造体のフィールドが優先される。いずれの場所にも値が存在しない場合、この構造体は無効となる。デフォルト値は存在しない。
   ? "initHash": bstr, ; 複数のファイルに分割されたフラグメント化MP4アセットの場合、このフィールドは必須であり、除外配列にリストされたボックスを除いた、このマークルツリーでハッシュ化されたチャンク全体の初期化セグメントファイルのハッシュである。単一のフラットMP4ファイルとして保存される断片化MP4アセットの場合、このフィールドは存在が必須であり、exclusions配列にリストされたボックスを除外した最初の'moof'ボックス以前に存在する全バイトのハッシュである。非断片化MP4の場合、このフィールドは存在してはならない。
   "hashes": [1* bstr], ; マークルツリーの単一行を表す順序付き配列。最下位行、ルート行、または中間行のいずれかである可能性がある。行の深さは、この配列内の項目数によって暗黙的に示される（計算される）。
   ? "fixedBlockSize": uint, ; mdatボックスが部分的に検証される非フラグメント化MP4アセットにおいて、このフィールドが存在する場合がある。このフィールドは、マークルツリー内の特定の葉ノードの非負のサイズ（バイト単位）を示す。フラグメント化MP4では、このフィールドは存在しない。
   ? "variableBlockSizes": [1* int], ; mdatボックスが部分的に検証される非フラグメント化MP4アセットにおいて、このフィールドが存在する場合がある。配列の各エントリは、Merkle ツリー内の特定のリーフノードの非負のバイトサイズに対応する。要素数は `count` に等しく、値の合計は mdat のペイロードサイズに等しい。フラグメント化 MP4 の場合、このフィールドは存在しない。
}
```

単一構造のMP4ファイルアセットにおいて、mdatボックスが単位として検証される場合のCBOR診断表記法（RFC 8949、第8条）の例を以下に示す：

```
{
   "hash": b64'EiAuxjtmax46cC2N3Y9aFmBO9Jfay8LEwJWzBUtZ0sUM8gA=',"name": "例
   `c2pa.hash.bmff.v2` アサーション",
   "exclusions": [
      {
         "data": [
            {
               "value": b64'2P7D1hsOSDySl1goh37EgQ==', "offset": 8
            }
```

```
        ],
        "xpath": "/uuid"
      },
      {
        "xpath": "/ftyp"
      },
      {
        "xpath": "/mfra"
      },
      {
        "xpath": "/moov[1]/pssh"
      },
      {
        "xpath":
        "/emsg","data": [
          {
            "value": b64'r3avWCpXHkmKHATFsV0Q5g==', "offset": 20
          }
        ]
      }
   ]
}
```

断片化されたMP4ファイルで構成されるアセットのCBOR診断表記法（RFC 8949、第8条）の例を以下に示す：

```
{
   "alg": "sha256",
   "name": "fMP4用 `c2pa.hash.bmff.v3` アサーション例","merkle": [
      {
        "count": 23,
        "hashes": [ b64'HvWZOxKMfkSatRAygs8DJfnEEcN/G1BNi359NdIDxbQ=', b64'HvWZOxKMfkSatRAygs8DJfnEEcN/G1BNi359NdIDxbQ='
],
        "localId": 19,
        "initHash": b64'Hf0IgeqbL0m+FTTLpUWwsDGR8pvhUR1AlwvaXjQ0qGY=',"uniqueId": 17
      },
      {
        "count": 69,
        "hashes": [ b64'9Zk7Eox+RJq1EDKCzwMl+cQRw38bUE2Lfn010gPFtB0=',
b64'9Zk7Eox+RJq1EDKCzwMl+cQRw38bUE2Lfn010gPFtB0=',b64'mTsSjH5EmrUQMoLPAyX5xBHDfxtQTYt+fTXSA8W0Hf0=',
b64'mTsSjH5EmrUQMoLPAyX5xBHDfxtQTYt+fTXSA8W0Hf0=',b64'OxKMfkSatRAygs8DJfnEEcN/G1BNi359NdIDxbQd/Qg=' ],
        "localId": 38,
        "initHash": b64'Hf0IgeqbL0m+FTTLpUWwsDGR8pvhUR1AlwvaXjQ0qGY=', "uniqueId": 34
      },
      {
        "count": 46,
        "hashes": [ b64'OxKMfkSatRAygs8DJfnEEcN/G1BNi359NdIDxbQd/Qg=' ],"localId": 57,
        "initHash": b64'Hf0IgeqbL0m+FTTLpUWwsDGR8pvhUR1AlwvaXjQ0qGY=', "uniqueId": 51
      }
   ],
   "除外": [
      {
```

```
        "data": [
          {
            "value": b64'2P7D1hsOSDySl1goh37EgQ==', "offset": 8
          }
        ],
        "xpath": "/uuid"
      },
      {
        "xpath": "/ftyp"
      },
      {
        "xpath": "/mfra"
      },
      {
        "xpath": "/moov[1]/pssh"
      },
      {
        "data": [
          {
            "value": b64'9Q==', "offset": 5
          },
          {
            "value": b64'UAJXD79SlkG9rfnmcsqTUA==', "offset": 20
          },
          {
            "value": b64'OxKM', "offset": 70
          }
        ],
        "flags": b64'ZDNx',
        "xpath": "/emsg","length":
        200,
        "サブセット": [
          {
            "length": 7,
            "offset": 5
          },
          {
            "length": 28,
            "offset": 20
          },
          {
            "length": 63,
            "offset": 45
          },
          {
            "length": 112,
            "offset": 80
          }
        ],
        "version": 1
      }
    ]
}


{
   "alg": "sha256",
   "name": "例 `c2pa.hash.bmff.v3` 非フラグメント化MP4用アサーション",
  "merkle": [
```

```
    {
      "count": 3,
      "hashes": [ b64'HvWZOxKMfkSatRAygs8DJfnEEcN/G1BNi359NdIDxbQ=',
b64'HvWZOxKMfkSatRAygs8DJfnEEcN/G1BNi359NdIDxbQ=' ],"variableBlockSizes": [ 100, 30, 20
      ],
      "localId": 19,
      "initHash": b64'Hf0IgeqbL0m+FTTLpUWwsDGR8pvhUR1AlwvaXjQ0qGY=', "uniqueId": 17
    }
  ],
  "除外条件": [
    {
      "data": [
        {
          "value": b64'2P7D1hsOSDySl1goh37EgQ==', "offset": 8
        }
      ],
      "xpath": "/uuid"
    },
    {
      "xpath": "/ftyp"
    },
    {
      "xpath": "/mfra"
    },
    {
      "xpath": "/moov[1]/pssh"
    },
    {
      "data": [
        {
          "value": b64'9Q==', "offset": 5
        },
        {
          "value": b64'UAJXD79SlkG9rfnmcsqTUA==', "offset": 20
        },
        {
          "value": b64'OxKM', "offset": 70
        }
      ],
      "flags": b64'ZDNx',
      "xpath": "/emsg","length":
      200,
      "サブセット": [
        {
          "length": 7,
          "offset": 5
        },
        {
          "length": 28,
          "offset": 20
        },
        {
          "length": 63,
          "offset": 45
        },
        {
          "length": 112,
          "offset": 80
```

```
          }
        ],
        "version": 1
      }
    ]
}
```

このアルゴリズムの擬似コード実装は、例6「BMFFベースのハッシュアサーションの擬似コード」にある。

例6. BMFFベースのハッシュアサーションの擬似コード

```
offset = 0
While (offset < ファイルの長さ)
    オフセットから開始し、除外配列の任意のエントリと一致する最初のボックスの最初のバイトを検索する。これを first_excluded_byte と呼ぶ
        該当するボックスが見つからない場合、first_excluded_byte = ファイルの長さ を設定するそのボックスの長さを決定し、これを excluded_byte_count と呼ぶ
        該当するボックスが見つからなかった場合、excluded_byte_count = 0 を設定
    ハッシュに、オフセットから first_excluded_byte マイナス 1 までのすべてのバイト（両端を含む）を追加する
    first_excluded_byte < ファイルの長さ かつ first_excluded_byte の値を決定した除外範囲内に部分配列が存在する
        next_included_begin = first_excluded_byte に設定
        first_excluded_byte の値を決定した除外範囲内の部分配列の各エントリについて
            next_excluded_begin = この部分集合配列エントリのオフセットフィールドに first_excluded_byte を加算した値に設定する
            next_excluded_begin > next_included_begin の場合
                ハッシュに、next_included_begin と next_excluded_begin の間の全バイト（両端を含む）から1を引いた値を追加する
            next_included_begin を設定 = このサブセット配列エントリの長さフィールドに next_excluded_begin を加算
        next_included_begin < first_excluded_byte + excluded_byte_count の場合ハッシュに、
            next_included_begin と first_excluded_byte + excluded_byte_count より 1 少ない値（端数を含む）までハッシュに追加する
    offset = first_excluded_byte + excluded_byte_count を設定する
```

マークルマップのハッシュ生成例は、例7「マークルマップの推奨例」にある。

例7. マークルマップの推奨例

```
`fixedBlockSize` および `variableBlockSizes` フィールドが存在しない場合
    ハッシュに、mdatペイロードの開始アドレスから最終アドレスまでの全バイトを追加する。`fixedBlockSize`フィールドが存在し、`variableBlockSizes`フィールドが存在しない場合
    (1) の場合
        next_address = begin_address + fixedBlockSize
        next_address > mdat ペイロードの last address の場合next_address = mdat ペイロードの
            last address に 1 を加算hash_complete = true
        ハッシュに、begin_address から next_address マイナス 1 までのすべてのバイトを追加する（開始アドレスを含む）
        hash_complete が true の場合 break
        begin_address = next_address
```

```
`variableBlockSizes`フィールドが存在し、`fixedBlockSize`フィールドが存在しない場合
    (blockSize in variableBlockSizes) についてnext_address =
        begin_address + blockSize
        next_address が mdat ペイロードの最終アドレスより大きい場合 next_address = mdat ペイロ
            ードの最終アドレスに 1 を加算 hash_complete = true
        ハッシュに、begin_address から next_address までの全バイト（両端を含む）を 1 減らして加算する
        hash_complete が true の場合 break
        begin_address = next_address
```

18.6.4. 除外リストのプロファイル

18.6.4.1. 概要

このセクションでは、事前定義された名前付き拡張リストのプロファイルセットについて説明します。

18.6.4.2. 基本プロファイル

典型的な非時間メディア（例：静止画）および時間メディア（例：音声トラックの有無にかかわらず、断片化されているか否かを問わない動画）は、除外リスト要件に記載された必須の除外項目のみを含める必要がある。

18.6.5. 断片化BMFFエンティティ図

図 15「断片化された BMFF エンティティ図」は、断片化された BMFF マニフェストを構成する C2PA オブジェクト間の関係を示しています。

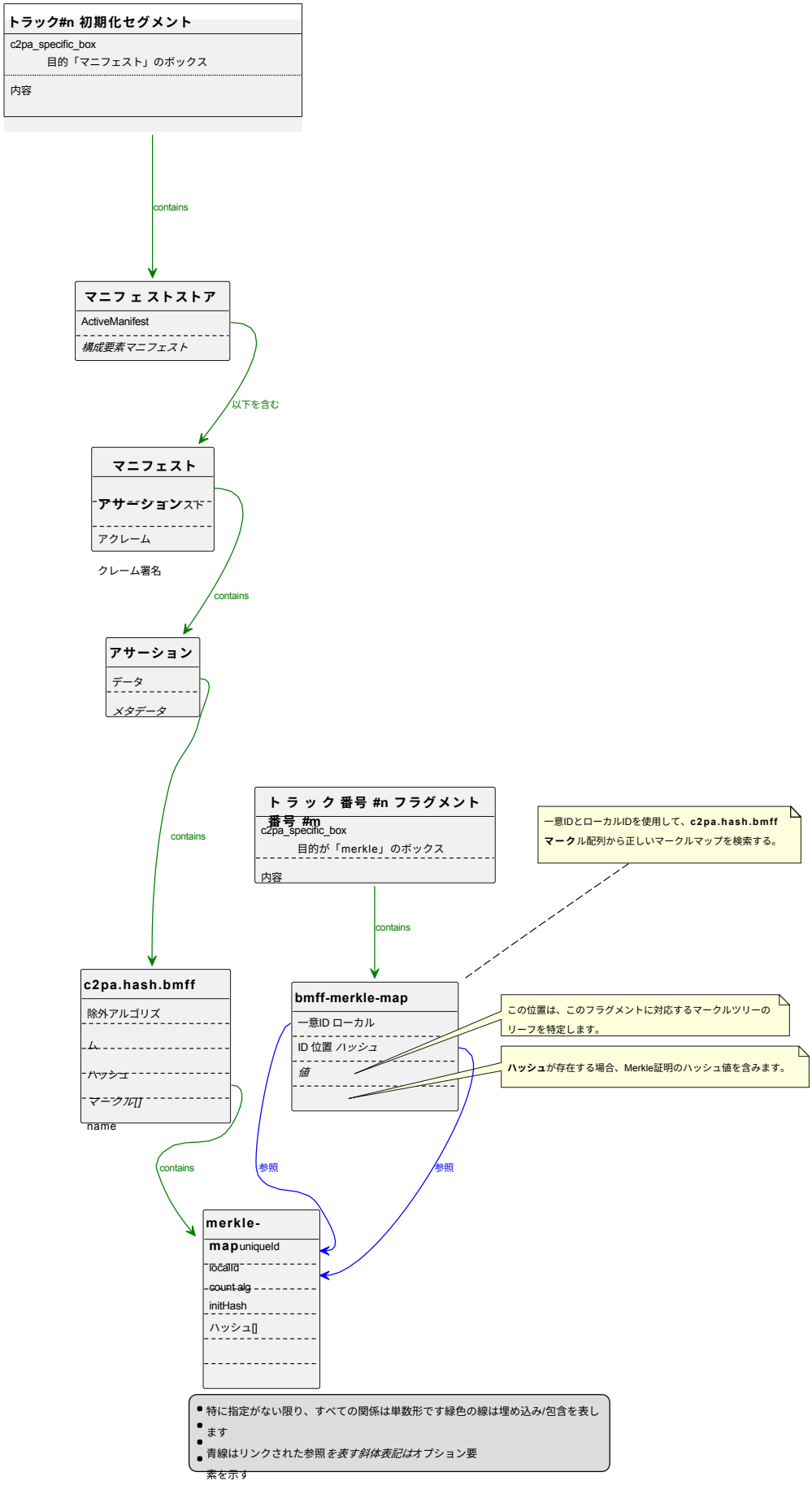

図15. 分割されたBMFFエンティティ図

18.6.6. 検証

指定されたチャンクの検証には、まず対応する初期化セグメントに対して `merkle-map` フィールドの `initHash` を検証し、次に `merkle-map` フィールドの `hashes` 配列から正しいエントリを見つけ、チャンクのデータのハッシュに対してそれを検証し、必要に応じて、チャンクの `bmff-merkle-map` で指定されたハッシュから Merkle 証明を使用してそのハッシュを導出する必要があります。

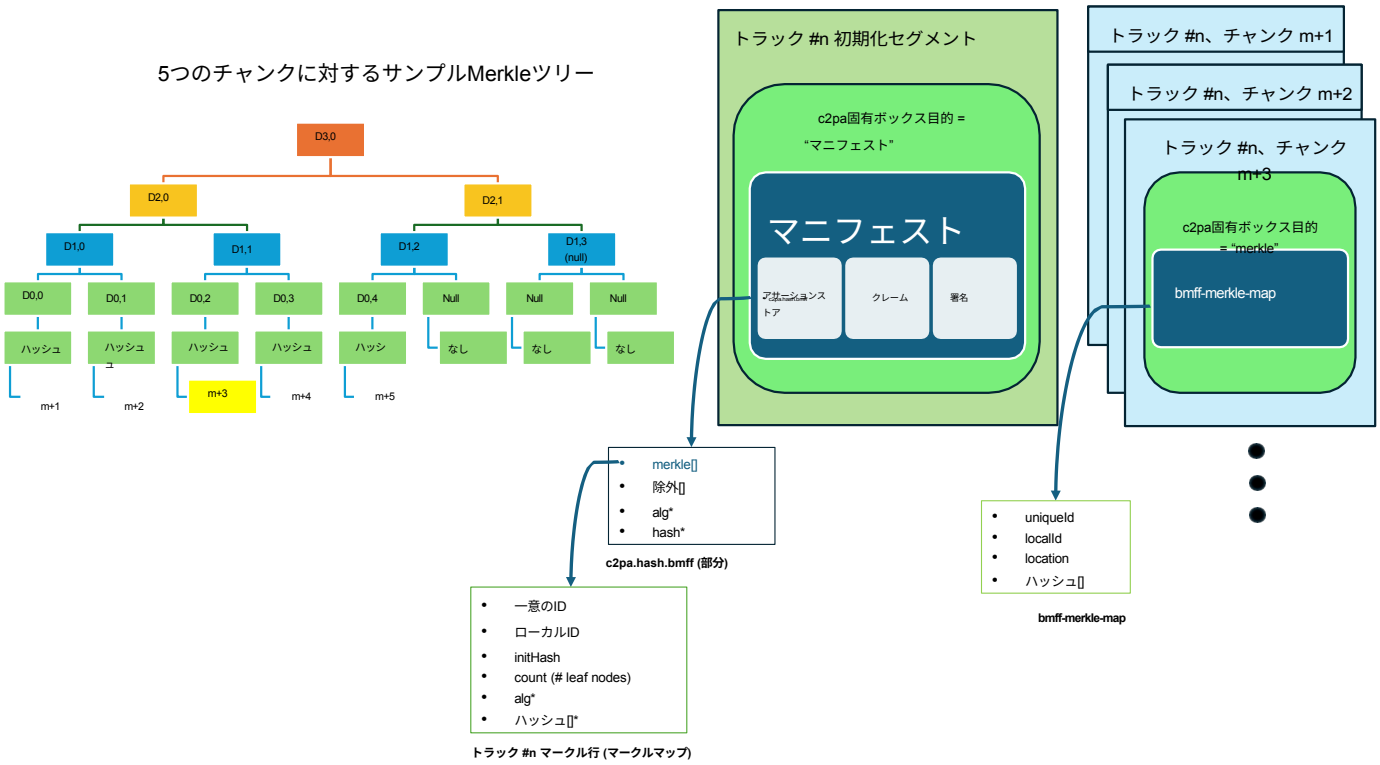

図16. 初期化セグメントとチャンクのデータの検証例

トラックチャンク **m+3** を検証するには、まず対応する初期化セグメントを検証する必要があります。各トラックの初期化セグメント内の c2pa 固有のマニフェストボックスには、マニフェストストアが含まれます。アセットに複数の初期化セグメントが含まれる場合、各セグメントのマニフェストストアは同一でなければなりません。これにより、バリデータはトラックがより大きなセットに属することを検証できます。アクティブなマニフェストの`c2pa.bmff.hash`アサーションには、各トラックごとに1つずつ、マークルマップオブジェクトの配列を含むマークルフィールドが含まれます。

18.6.6.1. 手順

1. チャンクのc2pa固有のマークルボックス内の`bmff-merkle-map`から`uniqueId`と`localId`を取得します。`uniqueId`と`localId`を使用して、初期化セグメント内の`c2pa.bmff.hash`アサーションマークル配列から一致するマークルマップを検索します。
2. `c2pa.bmff.hash`除外ルールとマークルマップアルゴリズムを用いた初期セグメントのハッシュが、initHashと一致する場合、初期化セグメントは検証済みとなります。

> **注記**
>
> `bmff-hash-map` の最上位にあるパラメータ `alg` および `hash` はモノリシック MP4 に使用され、`merkle-map` 内の `alg` および `hashes` はフラグメント化 MP4 に使用されます。

チャンク **m+3** の検証を完了するには：ステップ 1 で見つかったトラック #n のマークルマップを調べます。この例では、マークルツリーの 2 行目、**D2,0** および **D2,1** が含まれています。

3. `c2pa.hash.bmff`除外配列と`merkle-map`のアルゴリズムを用いてハッシュチャンク**m**+3を処理し、D**0,2(derived)**
4. チャンク m+3 の `bmff-merkle-map` ハッシュ配列（Merkle 証明）には、チャンク m+4 **(D0,3)** のハッシュと、行 1 のハッシュ値 **D1,0** が含まれる。
5. **D0,2(派生)**と**D0,3**を**ハッシュ**処理し、**D1,1(派生)**を生成する。**D1,0**と**D1,1(派生)**をハッシュ処理し、**D2,0(派生)**を生成する。
6. **D2,0(導出値)** がアサーション・マークルマップハッシュパラメータに格納された **D2,0** と等しく、かつ対応する初期化セグメントがステップ 2 で検証された場合、チャンク m+3 は検証済みである。

18.7. 汎用ボックスハッシュ

18.7.1. 説明

クレームジェネレータは、JPEG、PNG、GIF などの BMFF ベース以外のボックス形式を使用する資産の完全性を、ハードバインディング（すなわち、暗号ハッシュ）を用いて検証するために、汎用ボックスハッシュアサーションを使用すべきである。

一般的なボックスハッシュアサーションは、`c2pa.hash.boxes` のラベルを持つものとする。このアサーションは構造体の配列で構成され、各構造体は一つ以上のボックス（名前/識別子で指定）と、それらのボックスのデータ（およびそれらの間にファイル内に存在する可能性のあるあらゆるデータ）をカバーするハッシュ、ならびに使用されたハッシュアルゴリズムを列挙する。ボックスは、C2PAマニフェストを含むボックスを含め、アセット内の出現順序と同じ順序でアサーションに記述されなければならない。本アサーションに明示的に含まれていないボックスがアセット内に存在する場合、またはボックスの順序が異なる場合、セクション15.12.3「一般ボックスハッシュの検証」に記載の通り、検証時にマニフェストは拒否される。

ボックスには除外フィールドを指定できる。これはブール値で、検証時にバリデータがこのボックス（および関連するハッシュ）を無視できるかを示す。このフィールドが存在しない場合、または存在して値が`false`の場合、ボックスはハッシュ化され値が比較される。excludedフィールドが`true`の値を持つボックスについては、クレーム生成器は互換性のために正確なハッシュを含めるべきである。これは`excluded`フィールドを認識しない古いバリデータとの互換性を確保するためである。クレーム生成器が後方互換性を考慮しない場合、ハッシュとしてバイナリ文字列`00` 値0の単一バイト）を書き込むべきである。

同一のボックスタイプが複数存在するケース（例：JPEG 1ファイル内の複数のAPP1セグメント）では、各インスタンスをアサーション内で個別に列挙する。同一のセグメント識別子を共有するフラグメントであるJPEGセグメントも、別個のボックスとして列挙される。ただし、C2PAマニフェストストアを構成するセグメント（後述）は例外とする。

ハッシュの作成についてはセクション13.1「ハッシュ化」で説明されており、その値はハッシュフィールドに存在しなければならない。ボックスの範囲に対するハッシュ値は、範囲内の最初のボックスの先頭から最後のボックスの末尾までを基に計算される。これにはボックス間に存在する可能性のある任意のバイトも含まれる。

注記 ボックスの範囲を使用する場合、最初のボックスの開始位置から最後のボックスの終了位置までの全データが

ハッシュに含まれます。ただし、各ボックスを個別に列挙する場合、追加データは含まれず、列挙されたボックス内のデータのみが対象となります。

C2PAマニフェストストアを含むボックス（例：PNGの`caBX`、GIFの`21FF`）も、独自の配列でリスト化される。C2PAマニフェストボックスであることを明確に識別するため、名前は`C2PA`とし、ハッシュの値はバイナリ文字列`00`（値0の単一バイト）とする。C2PAマニフェストストアは、JPEGファイルのようにボックスが複数の`APP11`マーカーセグメントに分割されている場合でも、単一のボックスとして表現される。

注記 検証ツールはファイル解析ツールの出力を組み合わせて使用されることが多いため、セキュリティ上のベストプラクティスとしてC2PAマニフェストストア以外のファイルコンテンツ全体をハッシュ化します。これにより、メディアとリンクされたマニフェストの完全性が保証されます。

パディング値は常に存在し、ゼロ埋めされたバイト列でなければならない。ただし、複数パス処理中に他の値に置き換えられた場合は、パディングは存在しない。

注記 セクション 10.4「複数ステップ処理」では、正しい値の記入方法とパディングの調整方法について説明しています。

一般的なボックスハッシュアサーションは、クラウドデータアサーションには出現してはならない。

18.7.2. マルチパートアセットの特別な取り扱い

複数のパートで構成されるファイル形式（セクション18.9「マルチアセットハッシュ」で説明）をサポートするため、1つ以上のパートのデータがプライマリパートのボックスベースデータの後ろに配置される場合に対応する追加の論理ボックスを定義する。このボックスは`c2pa.after`（ボックス構造の末尾を超えた任意のデータ）とラベル付けされる。`c2pa.after`ボックスが存在する場合、それはリストの最後のボックスであり、そのハッシュは最後のボックスの直後のバイトから物理ファイルの終わりまでを対象に計算される。

アセット全体をカバーするハードバインディングアサーションは、`c2pa.after`ボックスを含むことができる唯一のアサーションである。
ボックスを含めることができる唯一の断言である。個々のパートに対するハッシュ断言は、そのパート自身のコンテンツのみをカバーし、他のいかなるパートもカバーしてはならない。

18.7.3. 特定のフォーマットの処理

18.7.3.1. JPEG固有の処理

JPEGを扱う場合、C2PA以外の規格（例：JPEG 360）ではAPP11ボックスが使用される。この場合、C2PA以外のすべてのAPP11ボックスはハッシュ処理対象ボックスのリストに含める必要があります。C2PAマニフェストストアを含むAPP11ボックスは`C2PA`で識別します。その他のボックスはすべて、ISO 10918-1:1994の表B.1に記載されている記号で識別します。

C2PAマニフェストストアは、セクション11.1.4.2「マニフェストストア」で説明されているように、ラベル`c2pa`を持つJUMBFスーパーボックスであり、JUMBFタイプ`UUID`が`63327061-0011-0010-8000-00AA00389B71`であることで識別できます。

注記 スキャン開始ボックスおよび再起動ボックス（ラベル `SOS` および `RST[n]`）には、それぞれのマーカーに続くエントロピー符号化セグメントが含まれます。

この方法では、JPEGのマルチピクチャーフォーマット（MPF）拡張もサポート可能です。具体的には、ファイル内に存在する全てのボックスを、出現順に列挙します。ただし、ある個別画像のEOIと次の個別画像のSOIの間にデータが存在しないことを前提とします。ボックスリストは、MPF内の各個別画像から得られるセグメントを順に列挙します（SOI、…, EOI, SOI, …, EOI, …）。ただし、クレーム生成者がMPFファイルをマルチパート資産として扱う場合、c2pa.afterボックスは、最初の個別画像（プライマリパート）のEOIに続く追加パートのハッシュ生成に使用されるものとする。

18.7.3.2. PNG固有の処理

PNGファイルは常に8バイトのヘッダー（89 50 4E 47 0D 0A 1A 0A）で始まる。これを包含するには、特別な値
ボックスリストの最初のボックスとしてPNGhを指定し、画像の最初のバイトからハッシュ処理を開始する。

18.7.3.3. TIFF固有の処理

TIFFファイルは常に8バイトのヘッダーで始まります。これを包含するには、ボックスリストの最初のボックスとして特別な値TIFhを使用します。

TIFFファイルは、1つ以上のIFD（画像ファイルディレクトリ）で構成され、これらは「スーパーボックス」に相当します。各IFDには「ボックス」を表す「IFDエントリ」または「TIFFフィールド」と呼ばれるエントリの配列が含まれます。各IFDエントリのボックス名は、Tagフィールドの値を10進値の文字列に変換した値とします。

他のボックス形式とは異なり、IFDエントリのデータはエントリ内に含まれない場合があります（データ長が4バイト以下の場合を除く）。代わりに、ファイル内の別の場所に存在します。

（IFDエントリのCountフィールドで決定される）	IFDエントリのデータ長は、データ値の数（ ）に各データ値のサイズ（ IFDエントリ内の型フィールド）。

IFDエントリのハッシュは、IFDエントリの12バイトに対して計算される。IFDエントリの長さが4バイトを超える場合、ハッシュは、IFDエントリのValue Offsetフィールドで指定されたバイトオフセットから始まり、データの長さにわたる、エントリが参照するファイルのバイトと、それらの12バイトの連結から計算される。

一部の既知のIFDエントリ（StripOffsets（273）、TileOffsets（324）、FreeOffsets（288））では、IFDエントリが参照するデータ自体が実際のデータへのオフセットのリストである。この場合、ハッシュ計算の対象となるデータは、以下の順序で連結されたものとする：

1. IFDの先頭12バイト
2. 値オフセットから始まるバイト列（オフセットを含む型のサイズ×カウントの値）

3. 各オフセットについて、そのオフセット位置にあるバイトを、型に関連付けられたバイトカウントエントリ（StripByteCounts (279)、TileByteCounts (325)、FreeByteCounts (289)）で指定された長さの分だけ取得する。

注記

したがって、TIFF 内の画像データは、「オフセット」と「バイトカウント」のこの組み合わせによってハッシュ化されます。

TIFF は、1 つ以上の IFD を参照によって組み込む SubIFD という IFD タイプもサポートしています。これには、`SubIFD` (330) と呼ばれるタイプだけでなく、`EXIF` (34665)、`GPS` (34853)、および `Interoperability` (40965) も含まれます。これら全てのIFDタイプ、および同様の方法で他のIFDを参照するその他のIFDタイプにおいて、ハッシュ計算の対象となるデータは、以下の順序で連結されたものとする：

1. IFDの先頭12バイト
2. 以下のいずれか：
    a. `N = 1` の場合、参照される IFD のオフセットを含む型のサイズ分のバイトを、値オフセットから開始して取得する。または
    b. `N > 1` の場合、IFD オフセット配列へのオフセットを含む型サイズの値オフセットから始まるバイトと、各「ツリー状」IFD へのオフセットを含む型サイズの `Count` 倍の長さの、そのオフセットから始まるバイトを連結する。
3. 参照される各IFDについて、本節で規定されるように、そのオフセットにおける当該IFDのハッシュ用データを再帰的に計算する。

18.7.3.4. GIF固有の処理

'Packed Fields' 属性を含むボックスのハッシュは、その属性によって示されるオプションデータもハッシュする。例えば、イメージ記述子にはローカルカラーテーブルブロックが含まれ、論理画面記述子にはグローバルカラーテーブルブロックが含まれる（それらが存在する場合）。

ブロックラベルを含むすべてのボックスについて、命名規則は「<ブロックラベル>」とする。

すべての拡張ブロックの命名規則は次のとおりです：「<拡張導入部><拡張ラベル>」。上記の命名規則に該当しないブロック

は他に以下のものだけです：

- ヘッダーには「GIF89a」のマークが付与されます。
- テーブルベース画像データは「TBID」とマークされます。
- 論理画面記述子は「LSD」とマークされます。

例：

- ヘッダー: "GIF89a"
- トレーラー: "3B"。
- 画像記述子: "2C"。
- コメント拡張子: "21FE"。

18.7.3.5. RIFF固有の処理

RIFFファイルのチャンクは、任意の深さのツリー構造にネストされることがある。この構造のルートは、1つ以上のL0チャンクで構成され、各チャンクの識別子は`RIFF`です。これらの`RIFF`チャンクは以下の構造で定義されます：

- バイト 0-3: チャンク識別子、常に `RIFF`。
- バイト 4-7: チャンク長（チャンク識別子とチャンク長フィールド分の8バイトを差し引いた値）。
- バイト 8-11: メディアタイプ識別子。
- バイト 12-n: チャンクデータ（すべての **L1** チャンク）。

メディアタイプ識別子の後、`RIFF`チャンクは1つ以上の**L1**サブチャンクを含むことがあり、各サブチャンクは以下の構造を持つ：

- バイト 0-3: チャンク識別子。
- バイト 4-7: チャンク長（チャンク識別子とチャンク長フィールド分の8バイトを差し引いた値）。
- バイト 8-n: チャンクデータ。
- バイト n+1: パディングバイト（必要な場合）。

`LIST`という特別なチャンク識別子を使用することで、L1チャンク内にチャンクをネストできます。これらの`LIST`チャンク**はL0** RIFFチャンクの構造を模倣します:

- バイト 0-3: チャンク識別子、常に `LIST`。
- バイト 4-7: チャンク長（チャンク識別子とチャンク長フィールド分の8バイトを除く）。
- バイト 8-11: リスト型識別子。
- バイト 12-n: チャンクデータ（すべての **L2** チャンク）。
- バイト n+1: パディングバイト（必要な場合）。

一般ボックスハッシュの計算において、各**L0**チャンクはサイズが正確に12バイトの単一ボックスとして扱われ、ボックス名はメディアタイプ識別子（バイト8-11）と同一とする。各非`LIST` **L1**チャンクは、チャンク識別子（バイト0-3）と等しい名前を持つボックスとして扱われ、その内容はチャンク識別子の先頭（バイト0）からパディングバイト（存在する場合）までを含む。各`LIST` **L1**チャンクは、リストタイプ識別子（バイト8-11）と一致する名前を持ち、チャンク識別子の先頭（バイト0）からパディングバイト（存在する場合）までを含む内容を持つボックスとして扱われる。`LIST` **L1** チャンク内にネストされたすべてのチャンク（**L2** 以上）は、`LIST` **L1** チャンクの内容の一部として扱われ、単一のボックスとしてハッシュ処理される。

いずれの場合も、パディングバイトは先行するチャンクの内容の一部として扱われ、そのボックスのハッシュに含まれるものとする。

18.7.3.6. フォント固有の処理

フォントのテーブルは、`C2PA` テーブルを含め、ハッシュボックスに直接対応します。テーブルは、フォン

トのテーブルディレクトリに現れる順序で常に列挙されます。

テーブルディレクトリ自体はハッシュ対象コンテンツの一部ではないため、どのボックスにも含まれないことに注意してください。

チェックサム調整値は、ヘッダーテーブルを含むボックスのハッシュを計算する際にゼロ (`0`) として扱われる。

ヘッドテーブルを含むボックスのハッシュを計算する際、チェックサム調整値はゼロ（0）として扱わなければならない。

一般ボックスハッシュアサーションにおけるフォントテーブルのグループ化（または非グループ化）は、クレーム生成器に委ねられる。

注:広範な配布を目的として作成されたフォントでは、各テーブルを個別のボックスに割り当てることで利点が得られる場合があります。これにより、フォントが別の形式で再パッケージ化されても、そのハッシュは正しく検証され続けます。対照的に、サブセッターなど大量のフォントを自動生成するシステムでは、処理効率化のためにテーブルを少数のボックスに統合する選択が可能です。この場合、テーブル間のパディングが含まれるため、形式変換後にボックスハッシュが検証されない可能性があります。

フォント消費者は認識できないテーブルに対して反応してはならないため、既存のフォント処理インフラストラクチャは、ヘッダーテーブルのチェックサム調整値がC2PAテーブル自体の最終確定コンテンツ（ローカルマニフェスト全体を含む）を組み込むことを想定する。

18.7.4. スキーマと例

このタイプのスキーマは、CDDL for Box Hash の CDDL 定義における `box-map` ルールによって定義される：

CDDL for Box Hash

```
box-map = {
   "boxes": [1* box-hash-map],
   ? "alg":tstr .size (1..max-tstr-length), ; このアサーションでハッシュを計算するために使用される暗号ハッシュアルゴリズムを識別する文字列。C2PAハッシュアルゴリズム識別子リストから取得される。このフィールドが存在しない場合、ハッシュアルゴリズムは包含構造体の`alg`値が採用される。両方が存在する場合、この構造体のフィールドが使用される。いずれの場所にも値が存在しない場合、この構造体は無効となる。デフォルト値は存在しない。
}

box-hash-map = {
   "names": [1* box-name], ; 出現順に並んだボックス識別子を表す文字列の配列（例: `APP0`, `IHDR`）
   ? "alg": tstr .size (1..max-tstr-length), ; このアサーションでハッシュを計算するために使用される暗号ハッシュアルゴリズムを識別する文字列。C2PAハッシュアルゴリズム識別子リストから取得される。このフィールドが存在しない場合、ハッシュアルゴリズムは囲み構造体の `alg` 値が使用される。両方が存在する場合、この構造体のフィールドが使用される。いずれの場所にも値が存在しない場合、この構造体は無効となる。デフォルト値は存在しない。
   "hash": bstr, ; ハッシュ値のバイト列
   ? "excluded": bool, ; バリデータが検証時にこのボックス（および関連するハッシュ）を無視できるかどうかを示すブール値。このフィールドが存在しない場合、ボックスはハッシュ化され値が比較される。
   "pad": bstr, ; スペースを埋めるために使用されるゼロ埋めバイト文字列
}

box-name /= tstr .size (1..10)
```

CBOR診断表記法（RFC 8949、第8条）による5つの例を例示ボックスハッシュで示す：

1. JPEG;
2. PNG;
3. GIF;

4. DNG (TIFF)、SubIFD付き;

5. TTF。

例：ボックスハッシュ

```
// JPEG 例 //
{
     "alg" : "sha256", "boxes": [
          {
               "names" : ["SOI", "APP0", "APP2"],
               "hash" : b64'...', "pad"
               : b64'',
          },
          {
               "names" : ["C2PA"],
               "hash" : b64'AA==',
               "pad" : b64'',
          },
          {
               "names" : ["DQT", "SOF0", "DHT", "SOS", "RST0", "RST1", "EOI"],
               "hash" : b64'', "pad" :
               b64'',
          }
     ]
}

// PNG 例 //
// XMPボックスを除外 //
{
     "alg" : "sha256", "boxes": [
          {
               "names" : ["PNGh", "IHDR"],
               "hash" : b64'...', "pad"
               : b64'',
          },
          {
               "names" : ["C2PA"],
               "hash" : b64'AA==',
               "pad" : b64'',
          },
          {
               "names" : ["sBIT"],
               "hash" : b64'...',"pad" :
               b64'',
          },
          {
               "names" : ["iTXt"],
               "hash" : b64'...',
               "excluded": true, "pad" :
               b64'',
          },
          {
               "names" : ["IDAT", "IEND"],
               "hash" : b64'...', "pad"
               : b64'',
          }
     ]
}
```

```
// GIFの例 //
{
     "alg" : "sha256", "boxes": [
          {
               "names" : ["GIF89a", "LSD"]
               "hash" : b64'...', "pad"
               : b64'',
          },
          {
               "names" : ["2C", "TBID", "2C", "TBID"],
               "hash" : b64'...', "pad"
               : b64'',
          },
          {
               "names" : ["21FE"],
               "hash" : b64'...', "pad"
               : b64'',
          },
          {
               "names" : ["21F9"],
               "hash" : b64'...', "pad"
               : b64'',
          },
          {
               "names" : ["3B"],
               "hash" : b64'...', "pad"
               : b64'',
          },
     ]
}

// TIFF/DNG の例 //
{
     "alg" : "sha256", "boxes": [
          {
               "名前" : ["TIFh", "254", "256", "257", "258", "259", "262"],
               "hash" : b64'...', "pad" :
               b64'',
          },
          {
               "names" : ["273", "277", "278", "279", "284"],
               "hash" : b64'...', "pad" :
               b64'',
          },
          {
               // これは二次画像を含むサブIFDです //
               "names" : ["330", "254", "256", "257", "258", "259", "262", "277", "278", "279",
"284"],
               "hash" : b64'...', "pad" :
               b64'',
          },
          {
               "names" : ["700", "34665"],
               "hash" : b64'...', "pad" :
               b64'',
          },
          {
               "names" : ["C2PA"],
               "hash" : b64'AA==',
               "pad" : b64'',
```

```
          }
     ]
}

// TTF の例 //
{
     "alg" : "sha256", "boxes": [
          {
               "names" : ["C2PA"],
               "hash" : b64'AA==',
               "pad" : b64'',
          },
          {
               "names" : ["PCLT"],
               "hash" : b64'...',"pad" :
               b64'',
          },
          {
               "names" : ["cmap"],
               "hash" : b64'...', "pad"
               : b64'',
          },
          {
               "names" : ["cvt"],
               "hash" : b64'...', "pad"
               : b64'',
          },
          {
               "names" : ["fpgm"],
               "hash" : b64'...', "pad"
               : b64'',
          },
          {
               "names" : ["gasp"],
               "hash" : b64'...', "pad"
               : b64'',
          },
          {
               "names" : ["glyf"],
               "hash" : b64'...', "pad"
               : b64'',
          },
          {
               "names" : ["head"],
               "hash" : b64'...', "pad"
               : b64'',
          },
          {
               "names" : ["hhea"],
               "hash" : b64'...', "pad"
               : b64'',
          },
          {
               "names" : ["hmtx"],
               "hash" : b64'...', "pad"
               : b64'',
          },
          {
               "names" : ["loca"],
               "hash" : b64'...', "pad"
               : b64'',
          },
```

```
            {
                 "names" : ["maxp"],
                 "hash" : b64'...', "pad" :
                 b64'',
            },
            {
                 "names" : ["name"],
                 "hash" : b64'...', "pad" :
                 b64'',
            },
            {
                 "names" : ["post"],
                 "hash" : b64'...', "pad" :
                 b64'',
            },
            {
                 "names" : ["prep"],
                 "hash" : b64'...', "pad" :
                 b64'',
            }
       ]
}
```

18.8. コレクションデータハッシュ

18.8.1. 説明

C2PAマニフェストが単一のアセットではなくアセットの集合を参照することが事前に判明しているワークフローにおいては、集合データハッシュアサーションを、集合内のアセットに対するハードバインディング（すなわち暗号ハッシュ）を指定する方法として使用しなければならない。

注記	AI/MLモデルのトレーニングデータセットの各フォルダは、完全なトレーニングデータセットのマニフェストにおいて個別の構成要素として記述することが可能です。

コレクションデータハッシュアサーションは、`c2pa.hash.collection.data` のラベルを持つものとし

ます。コレクションデータハッシュアサーションは、クラウドデータアサーションには出現してはなりませ

ん。

18.8.2. スキーマと例

このタイプのスキーマは、以下のCDDL定義における`collection-data-hash-map`ルールによって定義されます：

```
; URI とそれに対応するハッシュの配列
$collection-data-hash-map /= { "uris": [1* uri-
   hashed-data-map],
   "alg": tstr .size (1..max-tstr-length), ; `uris`配列の各エントリのハッシュ計算に使用される暗号ハッシュアルゴリズムを識別する文字
列。C2PAハッシュアルゴリズム識別子リストから取得。
     ? "zip_central_directory_hash" : bstr,
}

; URIとそのハッシュへの参照を格納するために使用されるデータ構造。
$uri-hashed-data-map /= {
```

```
   "uri": relative-url-type, ; 相対URI参照"hash": bstr, ; ハッシュ値を含むバイト列
   ? "size": size-type, ; データのバイト数
   ? "dc:format": フォーマット文字列, ; データのIANAメディアタイプ
   ? "data_types": [1* $asset-type-map], ; データタイプに関する追加情報
}

; CBOR ヘッダー (#) およびテール ($) は正規表現で導入されるため、明示的に指定する必要はありません。relative-url-type /= tstr
.regexp "[-a-zA-Z0-9@:%._\\+~#=]{2,256}\\.[a-z]{2,6}\\b[-a-zA-Z0-9@:%_\\+.~#?&//=]*"
```

CBOR診断表記法（RFC 8949、第8条）の例を以下に示す：

```
// リモートURLのリストの例 //
{
     "alg" : "sha256", "uris": [
          {
               "uri": "photos/id/870.jpg"
               "hash": b64'+ddHMTUUEpuSF6dNaHFa9uFc1sSnY+O3l3MMPFvX5Ws=', "dc:format": "image/jpeg"
          },
          {
               "url": "deepmind/bigbigan-resnet50/1","hash" :
               b64'...',
               "dc:format": "application/octet-stream", "data_types": [
                    {
                         "type": "c2pa.types.generator",
                    },
                    {
                         "type": "c2pa.types.model.tensorflow", "version":
                         "1.0.0",
                    },
                    {
                         "type": "c2pa.types.tensorflow.hubmodule", "version": "1.0.0",
                    }
               ]
          }
     ]
}

// (相対)ファイルURIのリストの例 //
{
     "alg" : "sha256", "uris": [
          {
               "uri": "image1.png",
               "hash": b64'U9Gyz05tmpftkoEYP6XYNsMnUbnS/KcktAg2vv7n1n8='
          },
          {
               "uri": "document.pdf",
               "hash": b64'G5hfJwYeWTlflxOhmfCO9xDAK52aKQ+YbKNhRZeq92c='
          },
     ]
}

// EPUB（ZIP形式）内の相対パス一覧の例 //
{
```

```
      "alg" : "sha256", "uris":
      [
            {
                  "uri": "mimetype"
                  "hash": b64'+ZXhhbXBsZSBvZiBhIGxpc3Qgb2YgcmVsYXRpdmUgc8=', "dc:format": "text/text"
            },
            {
                  "uri": "META-INF/container.xml"
                  "hash": b64'+ddHMTUUEpuSF6dNaHFa9uFc1sSnY+O3l3MMPFvX5Ws=',"dc:format": "text/xml"
            },
            {
                  "uri": "cover_page.svg",
                  "hash": b64'U9Gyz05tmpftkoEYP6XYNsMnUbnS/KcktAg2vv7n1n8='
            },
            {
                  "uri": "chapter1.html",
                  "hash": b64'G5hfJwYeWTlflxOhmfCO9xDAK52aKQ+YbKNhRZeq92c='
            },
      ]
}
```

18.8.3. フィールド

urisフィールドは、アセットの集合を表すuri-hashed-data-map値の配列で構成される。algフィールドは、セクション13.1「ハッシュ化」で説明されている通りであり、これをここに配置することで、リスト内の全コンテンツ項目が同一アルゴリズムでハッシュ化されることが保証される。

各uri-hashed-data-mapに対して、uriフィールドが存在し、有効な相対URIでなければならない。すべてのURIは、マニフェストの場所（ローカル、コンテナ内（例：ZIP）、クラウド内を問わず）に対する相対URIと見なされる。相対URIにはナビゲーション要素（例：../）が含まれる可能性があるため、マニフェストと同じフォルダにないコンテンツ項目を参照することが可能となり、これはセキュリティ上の問題となる。クレーム生成器は、使用前にURIを検証またはサニタイズし、URIの一部として「.」も「..」も出現しないことを保証しなければならない。

ハッシュフィールドは、コンテンツ項目の有効なハッシュ値を表すバイト列であり、アルゴリズムによって決定される。
フィールドによって決定される有効なハッシュ値を表すバイト列です。ハッシュはコンテンツ項目の全バイ

ト（0からnまで）に対して計算されなければならず、例外は認められません。残りのフィールドは成分ア

サーションのそれと同等です。

18.8.4. コレクションのメンバーのハッシュ化

コレクション内の各ファイルは、algフィールドで定義された特定のハッシュアルゴリズムを用いて個別にハッシュ化される。結果として得られるハッシュ値は、ファイルへのURIに関連付けられたuri-hashed-data-mapのhashフィールドに格納される。

特定の階層内の全ファイルをハッシュ済みコレクションに含める必要はない。

注記

これはハッシュ化する必要のないファイルが存在する場合に有用ですが、同時に攻撃者がバインディングを無効化せずにファイルを追加する余地も生じさせます。

18.9. マルチアセットハッシュ

18.9.1. 説明

複数の個別の画像ファイルを単一ファイルに集約する場合など、複数のパートで構成され、各パート自体が有効なファイル形式であるファイル形式がいくつか存在します。例としては以下が挙げられます：

- CIPAマルチピクチャーフォーマット（MPF）
- Android Ultra HDR フォーマット（MPFを使用）
- ISO 21496 HDR（MPFを使用）
- Android モーションフォト形式（MPFを使用しないが、同一ファイル内でMPFと共存可能）

場合によっては、ファイル全体ではなく個々の部分の完全性を検証することが望ましい、あるいは必須となることがあります。したがって、現行のハードバインディングアサーションでは、各部分の完全性を個別に検証するには不十分です。さらに、個々の部分には独自のC2PAマニフェストが存在し、記録が必要となる場合があります。マルチアセットハッシュアサーションはこの機能を提供するために使用されます。

もう1つの特異なケースとして、個々のパーツがオプションである場合が挙げられる。つまり、信頼された署名者を介さないワークフローの一環として削除される可能性があるが、ファイルの残りの部分の完全性を検証する機能は依然として必要とされる場合である。

18.9.2. 詳細

マルチアセットハッシュアサーションのラベルは`c2pa.hash.multi-asset`とする。ハッシュを含みハードバインディングの処理を変更するが、ハードバインディングとは見なされない。

マルチアセットハッシュアサーションは、クラウドデータアサーションに含めてはならない。

マルチアセットハッシュアサーションは、圧縮マニフェストと併用してはならない。

> **注記**
>
> 両者の間に技術的な非互換性が存在するかは不明であるため、さらなる評価が完了するまでは併用を避けることが推奨される。

プライマリパートを含む各パートは、parts配列内の`part-hash-map`オブジェクトとして表現されるものとする。`location`フィールドは、ファイル内のパートの位置を記述する`locator`オブジェクトを含むものとする。`locator`オブジェクトは、`bmffBox`フィールド、または`byteOffset`フィールドと`length`フィールドのいずれかを含むものとする。byteOffsetフィールドはファイル内のパートのバイトオフセット（ファイルの物理的開始位置からのオフセット）を、`length`フィールドはパートの長さ（バイト単位）を格納する。`bmffBox`フィールドは、パートがプライマリパート内に含まれるが特定のBMFFボックス（例：Motion Photoで使用されるmpvd）として存在する場合、そのパートのBMFFボックスを格納する。`bmffBox`フィールドで記述されるパートについては、そのパートの内容はボックスヘッダーを除き、当該ボックスのみのペイロードとみなされる。

parts配列内のパーツは、ファイル内の出現順にリストされ、パーツは連続し、重複せず、アセットの全バイトをカバーしなければならない。

注記

ファイル内での出現とは、ファイルのバイト 0 から最後のバイトまでスキャンした場合の、それらの連続した順序を定義する。

hashAssertionフィールドには、パーツのハッシュアサーションへのハッシュ化されたURIを含めること。パーツのハッシュアサーションは標準のハードバインディングアサーション（例：`c2pa.hash.data`）とするが、ラベルには文字列`.part`と任意の多重インスタンス識別子を付加すること。例：`c2pa.hash.data.part 2`。

注記

これらのラベル接尾辞を追加することで、パートのハードバインディングアサーションが考慮対象外であることが明確になる
標準的なハードバインディングアサーションとは見なされないことを明確にし、したがってC2PAマニフェスト内にそれらの複数インスタンスが存在し得ることを示す。

オプションフィールドは、パーツの存在がオプションであるかどうかを示すブール値とする。存在しない場合のデフォルトは`false`である。

パーツが独自のC2PAマニフェストを持ち、かつそのパーツ内に自己完結していない場合（例：マルチフレームアセット内の個々のフレーム）、そのC2PAマニフェストをアセットのマニフェストストアに保存し、参照するためのcomponentOfイングリディエントを作成することを推奨します。

18.9.3. スキーマと例

このタイプのスキーマは、以下のCDDL定義における`multi-asset-hash-map`ルールによって定義される：

```
multi-asset-hash-map = {
   "parts": [* part-hash-map] ; マルチパートファイルの各パートに対応する1つ以上のハッシュの配列
}

byte-range-locator = (
   "byteOffset": uint           ; ファイル内のパートのバイトオフセット"length": uint       ; パ
   ートの長さ
)

; これは選択肢の特別なCDDLマップです（つまり、以下のうち1つだけが存在できます）
locator-map = {
   byte-range-locator //             ; ファイル内のパーツのバイトオフセットと長さ "bmffBox": tstr    ; パートのBMFF
   ボックスへのXPath
}

part-hash-map = {
   "location" : locator-map, ; ファイル内のパーツの位置"hashAssertion": $hashed-uri-map, ; パーツのハッシュアサーシ
   ョンへのハッシュURI
   ? "optional": bool, ; パートがオプションで破棄可能かどうか
}
```

CBOR診断表記法（RFC 8949、第8条）の例を以下に示す：

```
// マルチアセットハッシュアサーション //
// 33,333バイトの資産は、バイト範囲[0,11111)のJPEGパートと別の
// 別の部分（バイト範囲 [11111,33333)）で構成される。
{
    "parts" : [
```

```
      {
        "location": { "byteOffset":
           0,
           "length": 11111
        },
        "hashAssertion": "self#jumbf=c2pa.assertions/c2pa.hash.boxes.part"
      },
      {
        "location": { "byteOffset":
           11111,
           "length": 22222
        },
        "hashAssertion": "self#jumbf=c2pa.assertions/c2pa.hash.data.part"
      }
   ]
}

// c2pa.hash.boxes.part - アセットの最初の部分のボックスハッシュ //
{
   "alg" : "sha256",
   "boxes": [
      {
        "names" : ["SOI", "APP0", "APP2"],
        "hash" : b64'...', "pad" :
        b64'',
      },
      {
        "names" : ["C2PA"],
        "hash" : b64'AA==',
        "pad" : b64'',
      },
      {
        "names" : ["DQT", "SOF0", "DHT", "SOS", "RST0", "RST1", "EOI"],
        "hash" : b64'', "pad" :
        b64'',
      }
   ]
}

// c2pa.hash.data.part - アセットの第二部分のデータハッシュ //
{
   "alg" : "sha256",
   "pad" : '0000',
   "hash" : b64'...',
}

// c2pa.hash.boxes - 全体的なアセットハッシュ、2部構成のアセット全体をカバー //
{
   "alg" : "sha256",
   "boxes": [
      {
        "names" : ["SOI", "APP0", "APP2"],
        "hash" : b64'...', "pad" :
        b64''
      },
      {
        "names" : ["C2PA"],
        "hash" : b64'AA==',
        "pad" : b64''
      },
      {
        "names" : ["DQT", "SOF0", "DHT", "SOS", "RST0", "RST1", "EOI"],
        "hash" : b64'...',
```

```
        "pad" : b64''
      },
      {
        "names" : ["c2pa.after"],
        "hash" : b64'...',
        "pad" : b64''
      }
    ]
}
```

このようなマルチアセットハッシュアサーションのサンプルは、[_multi_asset_hdr_image] に示すように画像に含まれる可能性があります。

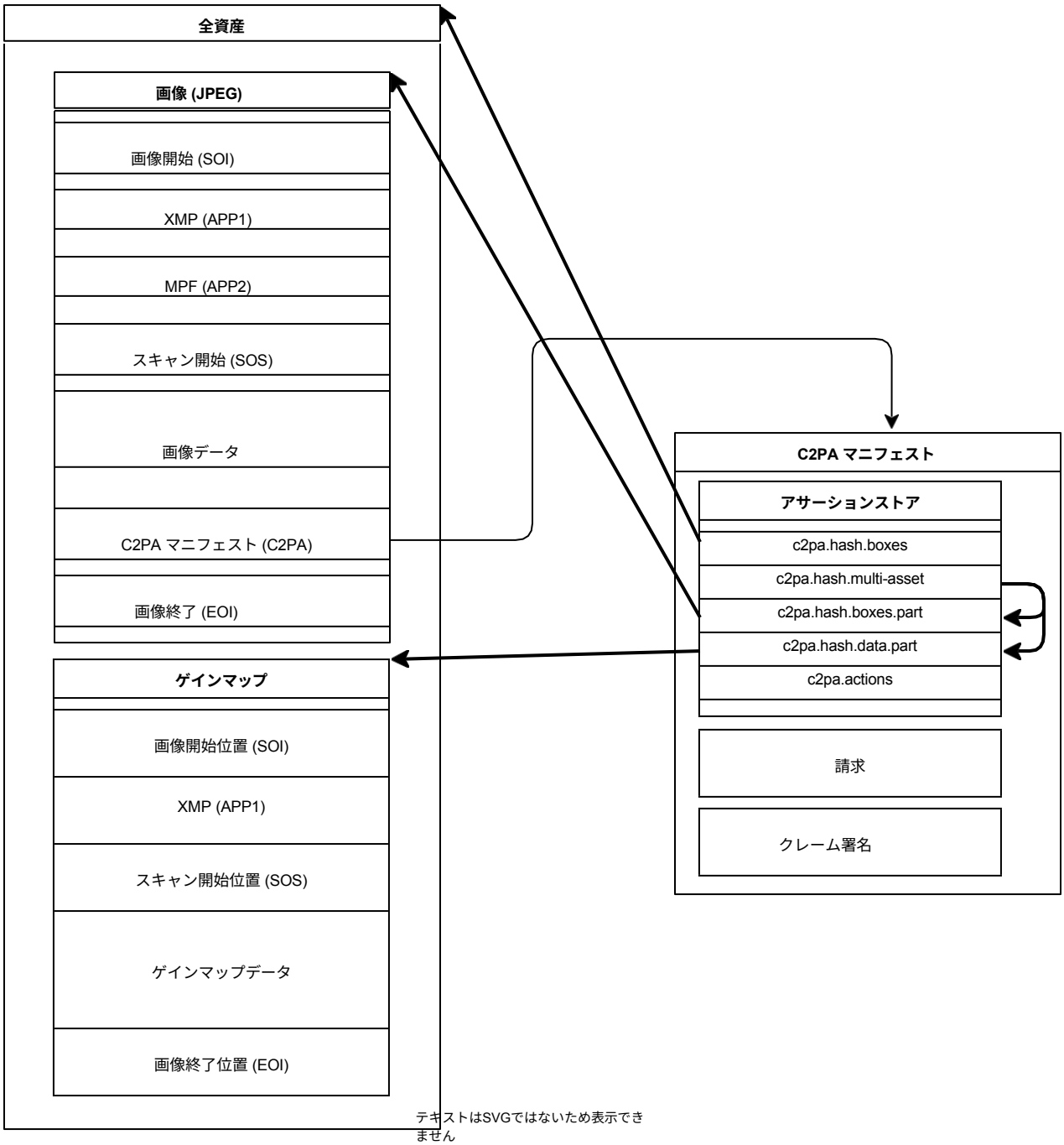

図17. HDRゲインマップに使用されるマルチアセットハッシュアサーションの例

18.10. ソフトバインディング

18.10.1. 説明

クレーム生成器が資産の内容に対してソフトバインディングを提供する場合には、ソフトバインディングアサーションを用いて記述しなければならない。当該アサーション内で作成および保存可能なソフトバインディングの種類については、セクション18.10「ソフトバインディング」で説明する。

この仕様の以前のバージョンでは、ハッシュ化されたデータの所在を示すポインタを提供するURLフィールドが用意されていたが、実際には使用されなかった。このフィールドは現在、アセット参照アサーションに取って代わられ非推奨となっている。クレーム生成者はソフトバインディングアサーションにこのフィールドを追加してはならず、消費者はこのフィールドが存在する場合にこれを無視しなければならない。ただし、セクション15.10.3「アサーション検証」で説明されている検証対象コンテンツの一部としてのこのフィールドの包含には影響しない。

本仕様の以前のバージョンでは、ソフトバインディングアサーションの対象となるデジタルコンテンツの一部をアルゴリズム固有の形式で記述するため、スコープフィールド内にエクステントフィールドを提供していました。このフィールドは現在、リージョンフィールドに取って代わられ非推奨となっています。クレーム生成者はソフトバインディングアサーションにこのフィールドを追加してはならず、コンシューマーは存在する場合このフィールドを無視すべきである。これは、セクション15.10.3「アサーション検証」で説明されている検証対象コンテンツの一部としてのフィールドの包含には影響しない。

ソフトバインディングアサーションのラベルは `c2pa.soft-binding` とする。

18.10.2. スキーマと例

このタイプのスキーマは、以下のCDDL定義における`soft-binding-map`ルールによって定義される：

```
関心領域オブジェクト構造を、ソフトバインディングアサーションにおいて他の目的で使用されるものと整合させる
;# 関心領域を含める

;アセットのコンテンツの一部または全体にわたる1つ以上のソフトバインディングを格納するために使用されるデータ構造
soft-binding-map = {
   "alg": tstr, ; 値を計算するために使用されるソフトバインディングアルゴリズムとそのバージョンを識別する文字列。C2PAソフトバインディングアルゴリズムリストから取得される。このフィールドが存在しない場合、アルゴリズムは囲み構造体の`alg_soft`値から取得される。両方が存在する場合、この構造体のフィールドが使用される。いずれの場所にも値が存在しない場合、この構造体は無効となる。デフォルト値は存在しない。
   "blocks": [1* soft-binding-block-map],
   "pad": bytes, ; スペースを埋めるために使用されるゼロ埋めバイト文字列
   ? "pad2": bytes, ; スペースを埋めるために使用されるオプションのゼロ埋めバイト文字列
   ? "name": tstr .size (1..max-tstr-length), ; (オプション) このハッシュがカバーする内容の人間が読める説明
   ? "alg-params": bstr, ; (オプション) ソフトバインディングアルゴリズムのパラメータを記述するCBORバイト文字列
   ? "url": uri, ; 未使用かつ非推奨。
}

soft-binding-block-map = { "scope": soft-
   binding-scope-map,
   "value": bstr, ; アルゴリズム固有の形式で記述された、
```

```
このデジタルコンテンツのブロックに対して計算されたソフトバインディングの値をアルゴリズム固有の形式で記述するCBORバイト文字列"
}

soft-binding-scope-map = {
  ? "範囲": bstr, ;非推奨, デジタルコンテンツのうちソフトバインディング値が計算された部分をアルゴリズム固有の形式で記述するCBORバイト文字列"
  ? "timespan":soft-binding-timespan-map,
  ? "region": region-map, ; regions-of-interest.cddl で定義された CBOR オブジェクト
}

soft-binding-timespan-map = {
  "start": uint, ; ソフトバインディング値が計算された時間範囲の開始点（メディア開始からのミリ秒単位）
  "end": uint,        ; ソフトバインディング値が計算された時間範囲の終了点（メディア開始からのミリ秒単位）
}
```

CBOR診断表記法（RFC 8949、第8条）による例を以下に示す：

```
{
  "alg": "phash",
  "pad": h'00',
  "url": 32("http://example.c2pa.org/media.mp4"),"blocks": [
    {
      "scope": {
        "timespan": {"end":
          133016
          "start": 0,
        }
      },
      "value": b64'dmFsdWUxCg=='
    },
    {
      "scope": {
        "timespan": {"end":
          245009
          "start": 133017,
        }
      },

    }
  ]
}
```

18.10.3. 要件

使用するソフトバインディングアルゴリズムは alg フィールドの値として存在し、適用対象となったブロックは `blocks` フィールドに列挙される。使用したアルゴリズムが追加パラメータを必要とする場合、それらは `alg-params` の値として存在すべきである。

スコープフィールドには、ソフトバインディングが計算されたデジタルコンテンツの部分を記述するために、リージョンフィールドまたはタイムスパンフィールドのいずれかを含めることができる。リージョンフィールドが存在する場合、それはリージョンマップオブジェクト（セクション18.2「関心領域」で定義）を含む。タイムスパンフィールドが存在する場合、それはコンテンツの開始時点からのミリ秒単位で、ソフトバインディングが計算された時間間隔を記述する。

18.10.4. ソフトバインディングアルゴリズム一覧

ソフトバインディングアルゴリズムリストは、algフィールドの許容値を機械可読形式で列挙したものです。`alg`フィールドは、当該リストに掲載されているアルゴリズムの`alg`フィールドに対応しなければなりません。`alg-params`フィールドおよび`value`フィールドの形式はアルゴリズム固有であり、リスト内の`alg`エントリ内で`informationalUrl`によって参照される人間可読情報ページを通じて記述されます。

このリストはC2PAにより以下の場所にあるJSON文書として管理される：https://github.com/c2pa-org/softbinding-algorithm-list

ソフトバインディングアルゴリズムリスト内のエントリで、`deprecated`フィールドが`true`の場合、そのエントリは非推奨と見なされ、マニフェスト内でソフトバインディングアサーションを作成するために使用してはならない。非推奨とマークされたソフトバインディングアルゴリズムはソフトバインディングの解決に使用できるが、この動作は推奨されない。

ソフトバインディングアルゴリズムリスト内のエントリに対するJSONスキーマは以下に示す通りです：

```
{
     "$schema": "https://json-schema.org/draft/2020-12/schema", "type": "object",
     "properties": {
          "identifier": {
               "type":
               "integer","minimum": 0,
               "maximum": 65535,
               "description": "この識別子は、ソフトバインディングアルゴリズムがリストに追加された際に割り当てられます。"
          },
          "deprecated": { "type":
               "boolean", "default":
               false,
               "description": "このソフトバインディングアルゴリズムが非推奨であるかどうかを示します。
非推奨のアルゴリズムはソフトバインディングの作成に使用してはなりません。非推奨のアルゴリズムはソフトバインディングの解決に使用できますが、この動作は推奨されません。"
          },
          "alg": {
               "type": "string",

               "description": "C2PAアサーションラベル用に指定されたエンティティ固有のネームスペース。Javaパッケージの定義と同様に、エンティティのインターネットドメイン名で始まる必要がある（例：`com.example.algo1`、`net.example.algos.algo2`）"
          },
          "type": {
               "type": "string","enum": [
                    "watermark",
                    "fingerprint"
               ],
               "description": "このアルゴリズムで実装されるソフトバインディングの種類。"
          },
          "decodedMediaTypes": { "type":
               "array", "minItems": 1,
               "items": {
                    "type": "string",
                    "enum": [
                         "application",
```

```
                        "audio",
                        "image",
                        "モデル",
                        "テキスト
                        ", "動画"
                    ],
                    "description": "このソフトバインディングアルゴリズムが適用されるIANAトップレベルメディアタイプ（レンダリング済み）"
               }
          },
          "encodedMediaTypes": { "type":
               "array", "minItems": 1,
               "items": {
                    "type": "string",
                    "description": "このソフトバインディングアルゴリズムが適用されるIANAメディアタイプ、例: application/pdf",

\\+]+)*)$"
               }
          },
          "entryMetadata": { "type":
               "object", "properties": {
                    "description": { "type":
                        "string",
                        "description": "アルゴリズムの人が読める説明。"
                    },
                    "dateEntered": { "type":
                        "string",
                        "format": "date-time",
                        "description": "このアルゴリズムの登録日。"
                    },
                    "contact": {
                        "type": "string",
                        "format": "email"
                    },
                    "informationalUrl": {
                        "type": "string",
                        "format": "uri",
                        "description": "アルゴリズムの詳細を記載したウェブページ。",
                    }
               },
               "required": [
                    "description",
                    "dateEntered", "contact",
                    "informationalUrl"
               ]
          },
          "softBindingResolutionApis": {"type": "array",
               "items": {
                    "type": "string",
                    "format": "uri"
               },
               "description": "このアルゴリズムをサポートするソフトバインディング解決APIの一覧
アルゴリズムをサポートするソフトバインディング解決APIの一覧です。"
          }
     },
     "required": [
          "identifier", "alg",
```

```
            "type",
            "entryMetadata"
      ],
      "oneOf": [
            {
                  "required": [
                        "decodedMediaTypes"
                  ]
            },
            {
                  "必須": [
                        "encodedMediaTypes"
                  ]
            }
      ]
}
```

ソフトバインディングアルゴリズムリストのエントリのJSON例を以下に示します:

```
{
      "identifier": 1,
      "deprecated": false,
      "alg": "com.example.product", "type":
      "watermark", "decodedMediaTypes": [
            "audio",
            "video",
            "text",
            "image"
      ],
      "entryMetadata": {
            "description": "Foo Inc.の透かしアルゴリズムバージョン1.2","dateEntered": "2024-04-23T18:25:43.511Z",
            "contact": "foo.bar@example.com ","informationalUrl":
            "https://example.com/wmdetails"
      },

      ]
}
```

アルゴリズムの固有名は `alg` フィールドに指定され、そのアルゴリズムを使用するソフトバインディングアサーションの `alg` フィールドで使用される文字列に対応する。この名前は名前空間要件に従い、アルゴリズムの所有者を表すものとする。各アルゴリズムには固有の数値識別子も割り当てられる。アルゴリズムの異なるバージョンが提供される場合、それぞれがソフトバインディングアルゴリズムリストに個別のエントリを持つものとする。

アルゴリズムのタイプは、不可視の透かし（ウォーターマーク）を表す場合は 'watermark'、指紋（フィンガープリント）を表す場合は 'fingerprint' のいずれかでなければならない。

アルゴリズムの非推奨状態は `deprecated` フィールドで示される。バリデータは非推奨アルゴリズムを使用するソフトバインディングを解決してはならない。C2PA マニフェストは非推奨のソフトバインディングを使用して記述してはならない。

ソフトバインディングアルゴリズムリストのエントリは、`encodedMediaTypes` または `decodedMediaTypes` のいずれかとしてサポートされ

るメディアタイプのリストを含むものとする。`decodedMediaTypes` でサポートされるメディアタイプは、以下のいずれかに対応するものとする。

最上位のIANAメディアタイプは以下で構成される：「application」、「audio」、「image」、「model」、「text」、「video」。encodedMediaTypesでサポートされるメディアタイプは、前文に記載されたdecodedMediaTypeの登録済みIANAサブタイプの一つ以上に対応するものとします。これらのIANAトップレベルおよびサブタイプはhttps://www.iana.org/assignments/media-types/media-types.xhtmlに記載されています。

ソフトバインディングアルゴリズムリストの各エントリには、entryMetadataフィールド内に追加情報を付随させるものとする。これらはアルゴリズムの人が読める説明（description）と、ソフトバインディングアルゴリズムリストへの登録が提案された日付（dateEntered）である。

エントリ所有者の連絡先はメールアドレス（contact、必須）として提供される。ソフトバインディングアルゴリズムの特性を説明する人間が読めるページを参照する情報用URL（informationalUrl、必須）を提供しなければならない。当該ページの情報は制約されないが、ソフトバインディングレジストリ内の値フィールド（アルゴリズム固有形式でエンコード）の解釈方法などの詳細を含む場合がある。

18.10.5. ソフトバインディング解決API

ソフトバインディング解決APIは、ソフトバインディング値、マニフェスト識別子、またはアセットを指定して、ソフトバインディング解決APIエンドポイントからC2PAマニフェストストアを取得する標準的な方法を提供するWeb APIです。ソフトバインディングアルゴリズムリストエントリには、softBindingResolutionApisフィールドにソフトバインディング解決APIのURIリストが含まれる場合があります。複数のURIが指定されている場合、いずれのAPIもソフトバインディング解決に使用できます。

API仕様とドキュメントはこちらから入手可能です。

18.10.5.1. ソフトバインディング一致の検証

ソフトバインディングの一般的な用途は、C2PAマニフェストが存在しない、または無効なアセットについて、マニフェストリポジトリからアクティブなマニフェストを発見することです。

C2PAマニフェストの発見は、C2PAソフトバインディングアルゴリズムリスト内のalgフィールドで識別される1つ、または複数のアルゴリズムの組み合わせを使用して実行される。このリストは、C2PAによって以下の場所でJSONドキュメントとして管理されている：https://github.com/c2pa-org/softbinding-algorithm-list

マニフェストリポジトリ内でC2PAマニフェストが見つかり、そのマニフェストに1つ以上のソフトバインディングアサーションが含まれる場合、マッチャーは、検出に使用されたソフトバインディングと、検出されたマニフェスト内のすべてのソフトバインディングアサーションが一致することを保証しなければならない。

ソフトバインディングアサーションは、アサーション内で記述されたアルゴリズム識別子（alg）と値（value）の両方が、照合を実行するために使用されたアルゴリズム識別子（alg）と値（value）と一致する場合に一致と見なされる。照合は、指定されたアルゴリズムによって規定された方法で実行される。

18.11. クラウドデータ

18.11.1. 説明

アセット内に埋め込むよりも、クラウドなどの遠隔地にアサーションのデータを保存する方が適しているユースケースが存在します。特にデータ量が大きい場合に有効です。そのようなケースでは、当該情報への参照として機能する特殊なタイプの断言を使用することが可能です。プライバシーと信頼性の観点から、クラウドデータ断言を介して参照されるデータはオプション扱いとすべきです：マニフェスト検証の一環としてその内容を取得すべきではありません。検証者は、出所履歴の詳細な調査などアプリケーション固有の必要性に応じて、後から内容を取得することがあります。

アサーションメタデータが別のアサーションの一部として含まれる場合、それ自体もクラウドデータアサーションから参照される情報の一部となります。他のアサーションタイプと同様に、個別のアサーションメタデータアサーションをリモートで保存することも可能です。

クラウドデータアサーションは、`c2pa.cloud-data` のラベルを持つものとします。

クラウドデータアサーションは、`c2pa.hash.data`、`c2pa.hash.boxes`、`c2pa.hash.collection.data`、`c2pa.hash.bmff.v2`（非推奨）、または`c2pa.hash.bmff.v3`のラベルを持つアサーションを参照してはならない。

18.11.2. スキーマと例

この型のスキーマは、以下のCDDL定義における`cloud-data-map`ルールによって定義されます：

```
; クラウドに格納された実際の断言を参照する断言cloud-data-map = {
   "label": tstr, ; クラウドベースのアサーションのラベル（例: c2pa.actions） "size": size-type, ; データのバ
   イト数
   "location": $hashed-ext-uri-map, ; クラウドホスト型アサーションの所在先となるhttp(s) URL
   "content_type": tstr .regexp "^[-\\w.]+/[-+\\w.]+$", ; データのメディア/MIMEタイプ
   ? "metadata": $assertion-metadata-map, ; アサーションに関する追加情報
}

; サイズは最小1で、1.0の倍数であること size-type = int
.ge 1
```

CBOR診断表記法（RFC 8949、第8条）の例を以下に示す：

```
{
   "size": 98765,
   "label": "c2pa.thumbnail.claim","location": {
      "url": "https://some.storage.us/foo",
      "hash": b64'zP84FPSremIrAQHlhw+hRYQdZp/+KggnD0W8opXlIQQ='
   },
   "content_type": "application/jpeg"
}
```

18.12. 埋め込みデータ

18.12.1. 説明

本仕様の以前のバージョンでは、サムネイル、アイコン、inputTo成分など、C2PAマニフェストへのデータの任意の埋め込みを可能にする手段として、JUMBFボックスの特殊なタイプであるデータボックスの概念が使用されていました。しかし、新しいタイプのボックスを介してこれを行うことは、データボックスの編集不可といった不要な複雑さや機能不足をもたらすことが判明しました。したがって、この概念は廃止され、標準のJUMBF埋め込みファイルコンテンツタイプボックスを使用してデータを格納する標準アサーションが採用されました。

埋め込みデータアサーションは、`c2pa.embedded-data` で始まるラベルを持ち、複数インスタンスに関するアサーションラベルの規則に従わなければならない。さらに、他のいくつかのアサーションタイプは技術的に埋め込みデータアサーションと同等であるが、独自の一意のラベルを持つ（例：`c2pa.thumbnail.claim`）。

18.12.2. 技術的詳細

埋め込みデータアサーションはJUMBF埋め込みファイルコンテンツタイプボックスに基づくため、その埋め込みファイル記述ボックスは *MEDIA TYPE* フィールドの値としてIANAメディアタイプ（例：`image/png`）を含み、*FILE NAME* フィールドの値としてファイル名を含む場合がある。`External` トグルビットは設定してはならない。

注記	IANA　構造化　サフィックス　(https://www.iana.org/assignments/media-type-structured-suffix/media-type-structured-suffix.xhtml）のような+jsonや`+zip`といった接尾辞も、*MEDIA TYPE* フィールドの値としてサポートされる。

埋め込みデータアサーションのバイナリデータボックスは、クレーム生成者が希望する任意の形式（ラスター画像やテキストプロンプトなど）のファイルのビットで構成されるものとし、埋め込みファイル記述ボックスで指定されたメディアタイプと一致するものとする。

18.13. サムネイル

18.13.1. 説明

サムネイルアサーションは、資産のライフサイクルにおける特定のイベント時点での資産のおおよその視覚的表現を提供します。現在、以下の2つの特定のイベントが定義されています：

- 成分インポートとクレーム作成。
- それぞれアサーションに固有のラベルを使用します。

18.13.1.1. クレームサムネイル

サムネイル　作成　　クレーム　作成　時間,　　サムネイル　アサーション　　ラベル `c2pa.thumbnail.claim`。C2PAマニフェスト内では、このラベルを持つサムネイルアサーションは1つだけ存在しなければならない。本仕様

の以前のバージョンでは、サムネイルのIANAレジストリメディアタイプを

ラベル名に含めることを要求していた（例：`c2pa.thumbnail.claim.png`）。この命名規則は廃止された。

18.13.1.2. 成分サムネイル

材料をインポートする際（セクション10.3.2.2「材料の追加」参照）、その材料自身のマニフェストに保存されたサムネイルを参照すべきである。ただし、一部の材料にはサムネイルアサーションが含まれていない場合や、マニフェスト自体が存在しない場合もある。その場合、材料の新しいサムネイルを生成し、アクティブなマニフェスト内に新しいサムネイルアサーションを作成する必要がある。

成分のサムネイルアサーションは、`c2pa.thumbnail.ingredient` で始まるラベルを持ち、複数インスタンスに関するアサーションラベルの規則に従うものとする。例えば、成分サムネイルのラベルは `c2pa.thumbnail.ingredient 1` となる可能性がある。

本仕様の旧バージョンでは、サムネイルのIANAレジストリメディアタイプをラベル名に含めることが要求されていた（例：`c2pa.thumbnail.claim.png`）。この命名規則は廃止された。

本仕様の以前のバージョンでは、最初のインスタンスに「`_1`」サフィックスを付与し、アンダースコアを1つ使用することが要求されていました。現在の仕様では、すべてのアサーションとの命名規則を統一するため、最初のインスタンスには`c2pa.thumbnail.ingredient`、2番目には`c2pa.thumbnail.ingredient 1`などを使用します。以前の命名規則は廃止されました。

18.13.1.3. 技術的詳細

サムネイルアサーションは、埋め込みデータアサーションの一種ですが、この特定のユースケースを識別する特別なラベルが付与されています。

18.14. アクション

18.14.1. 説明

アクションアサーションは、アセットの内容に影響を与える編集やその他のアクションに関する情報を提供します。一連のアクションが配列として存在します。各アクションは、アセットに対して*何*が行われたか、および（オプションで）いつ行われたかを宣言し、さらにアクションを実行したソフトウェアなどの追加情報を含む場合があります。セクション18.14.2「少なくとも1つのアクションアサーションの必須存在」で特に明記されている場合を除き、この配列内のアクションの順序は規定されておらず、アクションが実行された順序を示すものではありません。

アクションアサーションには2つのバージョンがあります。オリジナルのv1（ラベルは`c2pa.actions`とする）と、新しく改良されたv2（ラベルは`c2pa.actions.v2`とする）です。アクションはXMP ResourceEventsをモデルとしていますが、C2PA固有の調整がいくつか施されています。

v1アクションは、そのactions配列内で完全に指定されます。一方、v2では、アクションはactions配列内の要素で完全に指定されるか、またはtemplates配列内の同一アクション名を持つ要素から派生される場合があります。

アクション配列またはテンプレート配列のいずれかに存在する各アクションについて、actionフィールドの値は、以下のいずれかでなければなりません。事前定義済み アクション name (`c2pa.resized`, `c2pa.edited`, など) または エンティティ固有 アクショ

ン name

(`com.fabrikam.gaussianBlur` など)。

`c2pa.` で始まる事前定義済み名の一覧は、表8「事前定義済みアクション一覧」に記載されています：

表8. 事前定義アクション一覧

アクション	意味
c2pa.addedText	(可視) テキストレイヤーやキャプションなど、アセットにテキストコンテンツが挿入されました。
c2pa.adjustedColor	トーン、彩度などの変更。
c2pa.changedSpeed	ビデオまたはオーディオトラックの再生速度の低下または増加
c2pa.color_adjustments	[非推奨] トーン、彩度などの変更
c2pa.converted	アセットのフォーマットが変更されました。
c2pa.created	アセットが最初に作成されました。
c2pa.cropped	資産のデジタルコンテンツの一部が切り取られました。
c2pa.deleted	アセットのデジタルコンテンツの一部が削除されました。
c2pa.drawing	ブラシや消しゴムなどの描画ツールを使用した変更。
c2pa.dubbed	オーディオの変更が行われました。通常は複合アセットの1つ以上のトラックが対象です。
c2pa.edited	コンテンツの編集的変換と見なされる一般的な操作。
c2pa.edited.metadata	アセットのデジタルコンテンツではなく、アセットのメタデータまたはメタデータアサーションに対する変更。
c2pa.enhanced	ノイズ低減、マルチバンド圧縮、シャープニングなどの適用された強化処理。これらはコンテンツの非編集的変換を表す。
c2pa.filtered	適用されたフィルターやスタイルなどによる外観の変更。
c2pa.opened	既存のアセットが開かれ、`parentOf` として設定されています。 ingredientとして設定中。
c2pa.orientation	コンテンツの方向と位置の変更。
c2pa.placed	1つ以上の成分をアセットに追加/配置しました。
c2pa.published	アセットがより広い対象に公開されました。
c2pa.redacted	1つ以上のアサーションが編集されました。
c2pa.removed	成分の `componentOf` が削除されました。

c2pa.repackaged	ある包装または容器形式から別の形式への変換。コンテンツはトランスコーディングなしで再包装されます。この操作は親成分に対する編集的でない変換と見なされます。
c2pa.resized	コンテンツの寸法、ファイルサイズ、またはその両方の変更
c2pa.transcoded	あるエンコーディングから別のエンコーディングへの変換。解像度スケーリング、ビットレート調整、エンコーディング形式の変更を含む。この操作は親要素の編集的でない変換と見なされる。
c2pa.translated	コンテンツの言語変更。
c2pa.trimmed	コンテンツの時間範囲の削除。
c2pa.unknown	何かが起こったが、claim_generatorはそれが何かを特定できない。
c2pa.watermarked	ソフトバインディングを作成する目的で、デジタルコンテンツに目に見えない透かしが挿入されました。

さらに、フォントアセット専用に、font. で始まる以下の事前定義済み名前セット（表 9「フォントアクション一覧」参照）が使用されます：

注記 この仕様の以前のバージョンではこれらを`c2pa.font`としていましたが、より短いfontプレフィックスが推奨されるため廃止されました。

表9. フォントアクション一覧

アクション	意味
font.charactersAdded	文字または文字セットが追加されました。
font.charactersDeleted	削除された文字または文字セット。
font.charactersModified	追加および削除された文字または文字セット。
font.createdFromVariableFont	フォントが、全体または一部が可変フォントからインスタンス化されました。
font.edited	フォントは、より具体的なアクションで記述されていない編集アクションを受けました。
font.hinted	ヒント処理が適用されました。
font.merged	フォントは先行フォントの組み合わせです。
font.openTypeFeatureAdded	OpenType 機能がフォントに追加されました。
font.openTypeFeatureModified	OpenType 機能が変更されました。
font.openTypeFeatureRemoved	OpenType 機能がフォントから削除されました。
font.subset	フォントは、任意の（独自の）文字サブグループをサポートするために縮小されています。

18.14.2. 少なくとも 1 つのアクションアサーションが必須

There 必須 be 少なくとも 少なくとも 一つ アクション アサーション 現在 in いずれか the created_assertions または

標準的なC2PAマニフェストのClaimのgathered_assertions配列。さらに：

- アセットが*新規に*作成された場合（例：クリエイティブツールでの「ファイル→新規」操作、写真や動画の撮影、生成AIモデルによるメディア生成の結果）、クレームのcreated_assertions配列またはgathered_assertions配列内の最初のc2pa.actionsアサーションのactions配列は、最初の要素としてc2pa.createdアクションを持つ必要があります。
    - すべての資産について、c2pa.createdアクションと共に、適切な値を持つ対応するdigitalSourceTypeフィールドを記録し、資産の生成時の性質を示すものとする。デジタルコンテンツなしで資産が生成された場合、digitalSourceTypeフィールドの値はhttp://c2pa.org/digitalsourcetype/emptyとする。
- 既存の資産を編集用の親となる成分として開くことで資産が作成された場合、クレームのcreated_assertionsまたはgathered_assertions配列内の最初のc2pa.actionsアサーションのactions配列は、最初の要素としてc2pa.openedアクションを持つものとする。c2pa.openedアクションと組み合わせてdigitalSourceTypeフィールドは不要である。

注記 この要件は更新マニフェストには適用されません。

注記 gather_assertions にアクションを記録する場合は、これらのアサーションは署名者に帰属しないことに留意してください（第 10 章「*クレーム*」を参照）。

C2PAマニフェスト内のアクションアサーションの完全なセットには、c2pa.created または c2pa.opened のいずれかのタイプを持つアクションが1つ以上含まれてはならない。これらのアクションのいずれかが created_assertions 内に現れる場合、gather_assertions 内に現れてはならず、また、いずれかが gathered_assertions 内に現れる場合、created_assertions 内に現れてはならない。

例：生成AIモデルがテキストプロンプトに応答して動画を生成する場合、結果の動画アセットのアクティブマニフェストには、c2pa.created アクション（値：http://cv.iptc.org/newscodes/digitalsourcetype/trainedAlgorithmicMedia）で始まるc2pa.actionsアサーションが含まれ、対応するdigitalSourceTypeフィールドには が設定される。 対応するdigitalSourceTypeフィールドに格納されます。

例：ユーザーがソーシャルメディア投稿用の画像を作成するため、Emily's Mobile Poster Makerを開く。ユーザーはテンプレートを選択し、カスタマイズを開始。その過程で既存の写真をいくつかインポートする。結果として生成された画像アセットのアクティブマニフェストには、c2pa.createdアクションで始まるc2pa.actionsアサーションが含まれ、digitalSourceTypeフィールドは存在しません。これは新規ファイルとして作成されたことを示します。また、ユーザーがインポートした各写真に対してc2pa.placedアクションが含まれ、それぞれがcomponentOf関係が示された対応するingredientアサーションを指します。最後に、ユーザーが実行したその他の操作に対して追加のアクションが記録されます。

例：新聞社のメディアデスクは、フォトジャーナリストがC2PA対応カメラで撮影した写真を編集したいと考えています。メディア編集者は写真を開き、トリミングとビネット処理を適用します。編集後の写真アセットのアクティブマニフェストには、c2pa.actionsアサーションが含まれ、c2pa.openedアクションが元の写真の成分アサーションを指し示し、parentOf関係が示されます。また、

トリミングとビネット編集のアクションも含まれる。

18.14.3. すべてのアクションが含まれる

actions-map-v2には、ブール値であるフィールドallActionsIncludedを含めることができます。allActionsIncludedが存在し、値がtrueの場合、クレーム生成者は、その資産に対して実行されたアクションは、actionsアサーションに列挙されたもののみであると表明しています。allActionsIncludedが存在しないか、値がfalseの場合、マニフェストコンシューマーは、他のアクションが実行されたが列挙されていないと想定できます。

18.14.4. アクションアサーションのフィールド

18.14.4.1. 説明

アクションは、その動作内容を説明する自由記述形式の説明を description フィールドに含めることができます。これは非標準アクションにおいて最も有用ですが、標準アクションに関する追加情報を提供する手段としても使用可能です。例えば、c2pa.edited アクションには「ペイントブラシツール」という説明を付与できます。

18.14.4.2. 理由

存在する場合、理由フィールドには、アクションの根拠を示す以下の標準値のいずれか、またはエンティティ固有の名前空間と同じ構文に準拠したカスタム値を含めること：

- c2pa.PII.present;
- c2pa.invalid.data;
- c2pa.trade-secret.present;
- c2pa.government.confidential.

> **注記** 理由フィールドはあらゆるアクションに使用可能ですが、現時点では編集に焦点を当てたc2pa値のみが定義されています。

c2pa.redacted アクションを使用する場合、理由フィールドには編集の根拠を含める必要があります。c2pa.redacted アクションに関する追加要件は、セクション 18.14.4.7「パラメータ」に記載されています。

18.14.4.3. 使用時

また、whenフィールドにはアクションが発生した日時が含まれる場合がある。含まれる場合、whenフィールドの値はCBORの日時（RFC 8949、3.4.1）に準拠しなければならない。
はCBOR日付/時刻（RFC 8949、3.4.1）に準拠しているものとします。

> **注記** whenフィールドは単純な非信頼タイムスタンプとして機能します。UTCベースの時刻の使用が推奨されます。

18.14.4.4. SoftwareAgent

アクションを実行するために使用されたソフトウェアまたはハードウェアは、`softwareAgent`フィールドを介して識別できます。v1アクションでは、これは単純なテキスト文字列です。ただし、v2では、`softwareAgent`はより豊富な`generator-info-map`構造を使用します。

セクション10.2.3.2「ジェネレータ情報マップ」で説明されている。セクション18.14.6.2「ソフトウェアエージェント」で説明されているように複数のソフトウェアエージェントが使用される場合、softwareAgentIndexフィールドはsoftwareAgents配列内の0ベースのインデックスによってソフトウェアエージェントを参照するために使用される。特定のアクションは、`softwareAgent`フィールドまたは`softwareAgentIndex`フィールドのいずれか一方のみを持つものとする。

注記

これらのフィールドは、ソフトウェアエージェントがクレーム生成プログラムと同一でない場合に有用である。

注記

この仕様の以前のバージョンには `actors` フィールドも含まれていたが、バージョン 2.0 で削除された。

18.14.4.5. デジタルソースタイプ

アクションには `digitalSourceType` キーを含めることができ、その値は IPTC で定義された用語のいずれか、または以下のリストにある C2PA 固有の値のいずれかでなければなりません：

`http://c2pa.org/digitalsourcetype/empty`

デジタルコンテンツが実質的に空であるメディア。例えば、空白のキャンバスやゼロ長動画など。

`http://c2pa.org/digitalsourcetype/trainedAlgorithmicData`

サンプリングされたコンテンツとデータから導出されたモデルをアルゴリズム的に使用した結果のデータ。`http://cv.iptc.org/newscodes/digitalsourcetype/trainedAlgorithmicMedia`とは異なり、結果はメディアタイプ（例：画像や動画）ではなくデータ形式（例：CSV、pickle）である。

注記

One common use case for the `digitalSourceType` key is in conjunction with the `c2pa.created` アクションは、メディアアイテムの作成方法を指定する手段を提供します。例えば「デジタルキャプチャ」「ネガからのデジタル化」「訓練済みアルゴリズムメディア」などです。

生成AIによって作成されたような「訓練済みアルゴリズム」資産およびデータについては、資産の生成につながった入力に関する情報を提供するために、1つ以上の成分をC2PAマニフェストに追加することができます。これらは、例8「生成AIのためのアクションの例」に示すように、`c2pa.placed` または `c2pa.created` アクションから参照することができます。

18.14.4.6. 変更点

操作は、資産の一部（ビデオの特定のフレーム範囲や画像の特定領域など）に限定される場合があります。v1では、値は単純なテキスト文字列でした。v2では、変更フィールドを使用して識別され、その値は領域マップオブジェクトの配列です（セクション18.2「関心領域」で定義）。

18.14.4.7. パラメータ

アクションには、事前定義されたパラメータキーに加え、任意のカスタムフィールド（および関連する値）を自由に追加することで、アクション固有の情報を指定するためのパラメータキーを含めることができます。カスタムフィールドは、エンティティ固有の命名規則と同じ構文（例：`com.litware.someFieldName`）に準拠する必要があります。

注記

これは特定のワークフローやC2PAマニフェストコンシューマーにとって有用な追加情報を提供するために有用です。

同じパラメータと設定で同じアクションを繰り返し実行するクレームジェネレータは、`multipleInstances`フィールドを使用して、アクションが複数回実行されたかどうかを示すことができます。`multipleInstances`フィールドが存在しない場合、アクションが複数回実行されたかどうかは不明です。

`c2pa.opened` または `c2pa.placed` アクションを使用する場合、パラメータオブジェクト内の `ingredient` フィールド（v1）または `ingredients` フィールド（v2）には、関連する 1 つ以上の ingredient アサーションへのハッシュ化された JUMBF URI を含める必要があります。`c2pa.removed` アクションでは、このフィールドには別のマニフェスト内の `componentOf` ingredient アサーションへのハッシュ化された JUMBF URI を含める必要があります。場合によっては、食材の一部のみがアクションに関連する場合がある。そのような場合、食材アサーションには、食材の関連領域を指定するために使用されるregionOfInterestフィールドを含むアサーションメタデータを含めるべきである（セクション18.15.13「食材メタデータ」で説明）。

注記

この仕様の以前のバージョンでは、`c2pa.transcoded` および `c2pa.repackaged` アクション
は、先行する
`c2pa.opened`アクションによって参照されるparentOf成分アサーションを参照する必要がありました。クレーム生成者は、古いバリデータとの互換性のためにこれを行うことができます。

`c2pa.translated` アクションを使用する場合、
パラメータオブジェクトの sourceLanguage および targetLanguage フィールドは、RFC 5646、BCP 47 言語コードを含むものとします。

例 8. 生成AI向けアクションの例

生成AIモデルによって作成された画像に対する`c2pa.created`アクションは、CBOR診断表記法（RFC 8949、第8条）では以下のように記述される可能性があります：

```
// 生成AIの出力を記述するために使用されるアクションアサーション //
{
   "actions": [
      {
         "action": "c2pa.created",

         "softwareAgent" : {
               "name": "Joe's Photo Editor","version": "2.0",
               "operating_system": "Windows 10"
         },
         "digitalSourceType": "http://cv.iptc.org/newscodes/digitalsourcetype/trainedAlgorithmicMedia",
         "parameters" : {"ingredients" : [
               {
                  "url": "self#jumbf=c2pa.assertions/c2pa.ingredient.v3","alg": "sha256",
                  "hash" : b64'...',
               },
               {
                  "url": "self#jumbf=c2pa.assertions/c2pa.ingredient.v3 1","alg": "sha256",
                  "hash" : b64'...',
               }
            ]
```

```
         }
      }
   ]
}
```

c2pa.redactedアクションを使用する場合、パラメータオブジェクト内のredactedフィールドには、編集されたアサーションへのJUMBF URIを含める必要があります。

18.14.5. 透かし

c2pa.watermarked アクションを使用する場合、挿入された透かしを記述するために、ソフトバインディングアサーションも C2PA マニフェストに含める必要があります。

18.14.6. アクションテンプレート

18.14.6.1. テンプレート

v2アクションにおけるテンプレート配列の要素は、アクションに関する共通要素とテンプレート固有の値を組み合わせて記述されます。これらの値はC2PAマニフェストコンシューマーによって結合され、

同じ名前のアクション、またはすべてのアクション（アクションフィールドの値が特別な　値の場合）に対して、アクションの全体像を把握します。同じアクションに適用される複数のテンプレートがある場合、値はテンプレート（存在する場合）から始まり、テンプレート配列に現れる順序で適用され、その後マージされます。

アクションに適用される複数のテンプレートがある場合、値はテンプレート（存在する場合）から開始してマージされ、その後テンプレート配列に記述された順序で適用されます。

例9. アクションテンプレートの例

CBOR診断表記法（RFC 8949、第8条）におけるアクションとテンプレート：

```
// 単一テンプレートが複数アクションに適用される例 //
{
     "actions": [
          {
               "action": "com.joesphoto.filter", "when":
               0("2020-02-11T09:00:00Z")
          },
          {
               "action": "c2pa.edited",

          },
          {
               "action": "com.joesphoto.filter","when":
               0("2020-02-11T09:20:00Z")
          },
          {
               "action": "c2pa.cropped",

          }
     ],
     "templates": [{
          "action": "com.joesphoto.filter","description": "マ
          ジックフィルター",
```

```
            "digitalSourceType": "http://cv.iptc.org/newscodes/digitalsourcetype/compositeSynthetic",
            "softwareAgent" : {
                  "name": "Joe's Photo Editor","version": "2.0",
                  "operating_system": "Windows 10"
            }
      }]
}

// すべてのアクションを対象とする特殊なテンプレート `*` の使用例 //
{
      "actions": [
            {
                  "action": "c2pa.created", "when":
                  0("2024-03-09T20:04Z")
            },
            {
                  "action": "c2pa.edited",

            },
            {
                  "action": "c2pa.cropped",

            }
      ],
      "templates": [{
            "action": "*","digitalSourceType":
"http://cv.iptc.org/newscodes/digitalsourcetype/humanEdits","softwareAgent" :
            {
                  "name": "ジェーンズ・ヒューマン・オーサリング・ツール", "version":
                  "1.0"
            }
      }]
}
```

C2PAマニフェストコンシューマーは、テンプレートから値を取得し、アクション自体からの値を上書きします。これにより、同じ名前の値はすべて置き換えられます。

図18. アクションテンプレートフロー

テンプレートには、他の任意のキー（および関連する値）を含めることを可能にする `templateParameters` キーを含めることができます。これは、特定のワークフローや C2PA マニフェストコンシューマーにとって有用な追加情報を提供するために役立ちます。

18.14.6.2. SoftwareAgents

複数の softwareAgents を使用する場合、代わりに `softwareAgents` フィールドに列挙できます。このフィールドは `generator-info-map` オブジェクトの配列であり、それぞれが異なるソフトウェアまたはハードウェアを記述します。これらは、特定のアクションまたはテンプレートの `softwareAgentIndex` フィールドを介してインデックスで参照できます。

例 10. ソフトウェアエージェントの例

複数のアクションにまたがる複数のエージェントを指定する例（CBOR診断表記法、RFC 8949、第8条）：

```
{
     "actions": [
          {
               "action": "com.joesphoto.magic-avatar","when": 0("2020-
               02-11T09:00:00Z"),
               "softwareAgentIndex" : 0
          },
          {
               "action": "c2pa.edited",

               "softwareAgentIndex" : 1
          },
          {
               "action": "com.joesphoto.beauty-filter","when":
               0("2020-02-11T09:20:00Z"),
               "softwareAgentIndex" : 0
          },
          {
               "action": "com.joesphoto.all-smiles", "when": 0("2020-
               02-11T09:40:00Z"),
               "softwareAgentIndex" : 0
          },
          {
               "action": "c2pa.cropped",

               "softwareAgentIndex" : 1
          },
          {
               "action": "com.joesphoto.green-screen","when": 0("2020-
               02-11T09:50:00Z"),
               "softwareAgentIndex" : 0
          }
     ],
     "softwareAgents": [
          {
               "name": "Joe's AI Filter",
               "version": "1.0",
               "operating_system": "Windows 10"
          }
          {
               "name": "Joe's Photo
               Editor","version": "2.0",
               "operating_system": "Windows 10"
          }
     ]
}
```

18.14.6.3. アイコン

テンプレートにはアイコンを含めることもできます。これは画像（ラスターまたはベクター）であり、C2PAマニフェストコンシューマーのユーザーエクスペリエンス内でアクションのグラフィック表現として使用できます。マニフェストコンシューマーはすべての

定義されたアクションにおいて、そのようなアイコンはエンティティ固有のアクション用テンプレートにのみ存在すべきである。

アイコンフィールドの値が存在する場合、ハッシュ化されたURIでなければならない。このハッシュ化されたURIは、埋め込みデータアサーションまたはクラウドデータアサーションのいずれかにリンクするものである。埋め込みデータアサーションを使用する場合、そのラベルは`c2pa.icon`とし、複数インスタンスに関するアサーションラベルの規則に従わなければならない。

注記 このアイコンフィールドの構造は、クレームのジェネレーター情報マップ内のアイコンフィールドと同一である。

マニフェストコンシューマーは、本仕様の以前のバージョンで推奨されていたデータボックス方式もサポートすべきである。

18.14.7. ローカライゼーション

アクションアサーションのメタデータにテンプレートのローカライゼーション辞書が含まれている場合、そのローカライゼーションは、そのテンプレートに基づくあらゆるアクションにも適用されるものとする。

18.14.8. 関連アクション

一連のアクションが相互に関連し、通常は同時に実行される場合、それらを適切に関連付けることが有用です。v2アクションの関連フィールドは、関連する追加アクションを列挙する場所を提供します。各関連アクションは主アクションのサブセットであり、異なるフィールドのみを含める必要があります。アクションテンプレートと同様に、C2PAマニフェストコンシューマーによって各関連アクションの全体像を得るため、値は主アクションの値とマージされます。

18.14.9. アセットのレンディション

インターネット上でメディアを配信する際、アセットのレンディションは一般的な現象です。これらのレンディションは、接続環境や画面解像度などが異なる環境下でメディアを消費者に届ける目的で作成されることがよくあります。アクションアサーションを使用することで、消費側アクターが特定のクレーム作成者がアセットレンディションを作成する意図を理解する手助けができます。

c2pa.actionsアサーション内に`c2pa.published`、`c2pa.transcoded`、`c2pa.repackaged`アクションのみが存在する場合、マニフェストコンシューマーに対して「署名者が、構成要素アセットと本アセットの間でデジタルコンテンツへの編集変更が行われていないことを主張している」というシグナルを提供します。

単一の「parentOf」イングリディエントが追加で存在する場合、マニフェストコンシューマーに対して、署名者が当該アセットがその親から直接派生したことを主張しているというさらなるシグナルを提供します。

18.14.10. ソフトバインディング検索

C2PAマニフェストを含まないアセットに対して`c2pa.opened`または`c2pa.placed`アクションを実行する場合、クレームジェネレータはソフトバインディング検索を使用して当該アセットのC2PAマニフェストを検索することがあります。成功した場合、クレームジェネレータは検索されたC2PAマニフェストをingredientアサーションの`activeManifest`フィールドの値として追加する必要があります。この場合、成分アサーションにはsoftBindingsMatchedフィールド（値：true）と`softBindingAlgorithmsMatched`フィールド（値：配列内に少なくとも1つのエントリを

含む）も含まれる必要があります。

注記 これらのフィールドを追加することで、C2PA マニフェストのコンシューマーに対して、ソフトバインディング検索が使用されたことを示します。

注 ほとんどのソフトバインディングで復元されたマニフェストには、検索対象の資産と一致しないハードバインディングアサーションが含まれるため、イングレディエントアサーションで検証失敗が報告されることが予想されます。

ソフトバインディングを介してマニフェストが取得されたことを示す成分アクションの例（CBOR診断表記法、RFC 8949、第8条）：

```
// ソフトバインディングによりマニフェストが復元された成分アサーション //
{
   "dc:title": "image 1.jpg", "dc:format":
   "image/jpeg", "relationship": "parentOf",
   "softBindingsMatched": true,
   "softBindingAlgorithmsMatched": [
      "com.foo.watermark.1"
   ]
   "activeManifest": {
         "url": "self#jumbf=/c2pa/urn:c2pa:5E7B01FC-4932-4BAB-AB32-D4F12A8AA322","hash":
         b64'1kjJTO108b71cL95UxgfHD3eDgk9VrCedW8n3fYTRMk='
   },
}



// 成分を指すアクションアサーション //
{
   "actions": [
      {
         "action": "c2pa.opened",

         "softwareAgent": {
            "name": "Joe's Photo Editor","version": "2.0",
            "operating_system": "Windows 10"
         },
         "parameters":
            {"ingredients": [
               {
                  "url": "self#jumbf=c2pa.assertions/c2pa.ingredient.v3","alg": "sha256",
                  "hash": "b64'...'"
               }
            ]
         }
      }
   ]
}
```

18.14.11. 非推奨のアクション

以下の操作は、この仕様の以前のバージョンに含まれていましたが、現在は非推奨となっています：

- c2pa.copied;

- c2pa.formatted;
- c2pa.version_updated;
- c2pa.printed;
- c2pa.managed;
- c2pa.生成済み;
- c2pa.saved.

これらは、c2pa.actions または c2pa.actions.v2 アサーションには書き込まれなくなりますが、既存の C2PA マニフェストには表示される場合があります。

18.14.12. スキーマと例

c2pa.actions のスキーマは actions-map ルールによって定義され、c2pa.actions.v2 のスキーマは以下の CDDL 定義における actions-map-v2 ルールによって定義されます：

アクション用CDDL

```
actions-map = {
   "actions" : [1* action-items-map], ; アクションのリスト
   ? "メタデータ": $assertion-metadata-map, ; アサーションに関する追加情報
}

$action-choice /= "c2pa.addedText"
$action-choice /= "c2pa.adjustedColor"
$action-choice /= "c2pa.changedSpeed"
$action-choice /= "c2pa.color_adjustments"
$action-choice /= "c2pa.converted"
$action-choice /= "c2pa.copied"
$action-choice /= "c2pa.created"
$action-choice /= "c2pa.cropped"
$action-choice /= "c2pa.deleted"
$action-choice /= "c2pa.drawing"
$action-choice /= "c2pa.dubbed"
$action-choice /= "c2pa.編集済み"
$action-choice /= "c2pa.編集済み.メタデータ"
$action-choice /= "c2pa.filtered"
$action-choice /= "c2pa.フォーマット済み"
$action-choice /= "c2pa.管理済み"
$action-choice /= "c2pa.opened"
$action-choice /= "c2pa.orientation"
$action-choice /= "c2pa.produced"
$action-choice /= "c2pa.placed"
$action-choice /= "c2pa.印刷済み"
$action-choice /= "c2pa.published"
$action-choice /= "c2pa.redacted"
$action-choice /= "c2pa.removed"
$action-choice /= "c2pa.再パッケージ化済み"
$action-choice /= "c2pa.resized"
$action-choice /= "c2pa.保存済み"
$action-choice /= "c2pa.transcoded"
$action-choice /= "c2pa.translated"
$action-choice /= "c2pa.trimmed"
$action-choice /= "c2pa.unknown"
```

```
$action-choice /= "c2pa.version_updated"
$action-choice /= "c2pa.watermarked"
$action-choice /= "font.edited"
$action-choice /= "フォントサブセット"
$action-choice /= "font.createdFromVariableFont"
$action-choice /= "フォントの文字追加"
$action-choice /= "font.charactersDeleted"
$action-choice /= "font.charactersModified"
$action-choice /= "font.hinted"
$action-choice /= "font.openTypeFeatureAdded"
$action-choice /= "font.openTypeFeatureModified"
$action-choice /= "font.openTypeFeatureRemoved"
$action-choice /= "font.merged"
$action-choice /= tstr .regexp "([\\da-zA-Z_-]+\\.)+[\\da-zA-Z_-]+" buuid = #6.37(bstr)

; 注記: この仕様の以前のバージョンには「actors」フィールドも含まれていましたが、バージョン2.0で削除されました。
action-items-map = { "action":
   $action-choice,
   ? "when": tdate, ; アクション発生時刻のタイムスタンプ
   ? "softwareAgent": tstr .size (1..max-tstr-length), ; アクションを実行したソフトウェアエージェント。
   ? "changed": tstr .size (1..max-tstr-length), ; 前のイベント履歴以降に変更されたリソースのパーツをセミコロン区切りで列挙。存在しない場合は未定義とみなされる。変更を追跡する際、変更されたコンポーネントの範囲が不明な場合、あらゆる変更が想定される。
   ? "instanceID": buuid, ; 変更された（出力）リソースの xmpMM:InstanceID プロパティの値
   ? "parameters": parameters-map, ; アクションの追加パラメータ。これらはアクションの種類によって異なることが多い
   ? "digitalSourceType": tstr .size (1..max-tstr-length), ; https://cv.iptc.org/newscodes/digitalsourcetype/ で定義されたソースタイプの一つ
}

parameters-map = {
   ? "ingredient": $hashed-uri-map, ; このアクションが作用する成分アサーションへのハッシュ付きURI
   ? "description": tstr .size (1..max-tstr-length) ; アクションの追加説明
   * tstr => any
}


; アクションアサーションのバージョン 2 (v2)

$action-reason /= "c2pa.PII.present"
$action-reason /= "c2pa.invalid.data"
$action-reason /= "c2pa.tradesecret.present"
$action-reason /= "c2pa.government.confidential"
$action-reason /= tstr .regexp "([\\da-zA-Z_-]+\\.)+[\\da-zA-Z_-]+"

actions-map-v2 = {
   "actions" : [1* action-item-map-v2],                ; アクションのリスト
   ? "templates": [1* $action-template-map-v2], ; アクション用テンプレートのリスト
   ? "softwareAgents": [1* $generator-info-map], ; アクションを実行したソフトウェア/ハードウェアの一覧
   ? "metadata": $assertion-metadata-map,               ; アサーションに関する追加情報
   ? "allActionsIncluded": bool                         ; 存在し & true の場合、アクションリストに含まれていないアクションが発生しなかったことを示す。
}

action-common-map-v2 = {
```

```
   ? "softwareAgent": $generator-info-map, ; アクションを実行したソフトウェア/ハードウェアの説明
   ? "softwareAgentIndex": int, ; actions-map-2内のsoftwareAgents配列への0ベースのインデックス
   ? "description": tstr .size (1..max-tstr-length), ; アクションの詳細説明（カスタムアクションで重要）
   ? "digitalSourceType": tstr .size (1..max-tstr-length), ; https://cv.iptc.org/newscodes/digitalsourcetype/ または本仕
様で定義されたソースタイプの一つ
}

; 注記：この仕様の以前のバージョンには「actors」フィールドも含まれていましたが、バージョン2.0で削除されました。
action-item-map-v2 = {
   "action": $action-choice , ; アクションの種類action-common-map-v2, ;
   追加の共通項目
   ? "when": tdate, ; アクション発生時刻のタイムスタンプ
   ? "changes": [1* region-map], ; 変更されたリソースの関心領域のリスト。存在しない場合、未定義とみなされる。
   ? "related": [1* action-item-map-v2], ; 関連アクションのリスト
   ? "reason": $action-reason, ; このアクションが実行された理由。アクションが `c2pa.redacted` の場合必須
   ? "parameters": parameters-map-v2 ; アクションの追加パラメータ。これらはアクションの種類によって異なることが多い
}

action-template-map-v2 = {
   "action": $action-choice / "*", ; テンプレートは追加の特殊オプション "*" をサポートしますaction-common-map-v2,  ; 追加
   の共通項目
   ? "icon": $hashed-uri-map, ; 埋め込みデータアサーションへのハッシュ付きURI参照
   ? "templateParameters": parameters-common-map-v2 ; テンプレートの追加パラメータ。
}

parameters-common-map-v2 = (
   * tstr => any
)

parameters-map-v2 = {
   ? "redacted": $jumbf-uri-type, ; 編集済みアサーションへのJUMBF URI。アクションが`c2pa.redacted`の場合に必須
   ? "ingredients": [1* $hashed-uri-map], ; このアクションが作用する成分（v2 または v3）アサーションへのハッシュ付き JUMBF URI の
リスト
   ? "sourceLanguage": tstr .size (1..max-tstr-length), ; `c2pa.translated`アクションのソース言語のBCP-47コード
   ? "sourceLanguage": tstr .size (1..max-tstr-length), ; `c2pa.translated`アクションのソース言語のBCP-47コード
   ? "multipleInstances": bool, ; このアクションは複数回実行されたかparameters-common-map-v2, ; 共通
   パラメータの任意のもの
}
```

フォントアセット固有の標準アクションは以下に記載されています:

フォントアクションのCDDL

```
; フォント固有アクションのマップ、範囲、パラメータ

; 複数のフォントアクションは、Unicode値の範囲に対して機能します。font-unicode-range-map = {
   "start": uint, ; 開始値を含む "stop": uint, ;
   終了値を含む
}
```

```
; font.subset、font.charactersAdded、
; font.charactersDeleted, および font.charactersModified で使用されるフォ
ントパラメータ。font-parameter-unicode-ranges-map = {
   "ranges": [1* font-unicode-range-map] ; ユニコード範囲の配列
}

; フォントインスタンス化パラメータ用の範囲
font-weight-range = 1..1000 ; フォントの有効な太さまたは太さ。400が標準。font-width-range = 0.0..1000.0 ; 標準からのパー
センテージ（0%〜1000%）。100%が標準幅。
font-slant-range = -90.0..90.0 ; 傾斜角度。0度は傾斜なし。

; 可変フォントからインスタンスを作成する際に使用するフォントパラメータ。
; フォントの異なる「バリエーション軸」の詳細がここに記述される。軸の
; 各パラメータのコメント内の括弧内に記載されている。
; パラメータのコメント内に括弧書きで記載されています。
font-parameter-created-from-variable-font-map = {
   ? "weight": font-weight-range, ; インスタンス化されるフォントのウェイト(wght)または太さ。
   ? "width": font-width-range, ; インスタンス化されるフォントの文字幅（wdth）または狭さ。
   ? "italic": bool, ; フォントのイタリック体(ital)を取得する。
   ? "slant": font-slant-range, ; フォントの傾斜角(slnt)。
   ? "optical-size": int / float, ; フォントの光学サイズ(opsz)。通常は要求されたフォントサイズに一致させる。
   * tstr → any ; カスタム軸の名前と型。
}
```

例 11. v2 アクションの例

v2アクションの例を、CBOR診断表記法（RFC 8949、第8条）で以下に示す：

```
{
   "actions": [
      {
         "action": "c2pa.filtered",

         "parameters": {

         },
         "softwareAgent" : {
               "name": "Joe's Photo Editor","version": "2.0",
               "operating_system": "Windows 10"
         }
      },
      {
         "action": "c2pa.cropped",

      }
   ],
   "metadata": {

      "reviewRatings": [
         {
            "value": 1,
            "説明": "コンテンツバインディングの検証に失敗しました"
         }
      ]
   }
```

```
}
```

18.15. 材料

18.15.1. 説明

アセットを組み合わせて作成する場合（例：画像編集ツールで画像をレイヤーに配置する、動画編集ツールで音声クリップを動画に挿入する）、配置されたアセットからのあらゆる権利主張に関する情報を新しいアセットに記録することが重要です。これにより、新しく組み合わされたアセットの全履歴を把握する手段が提供されます。既存のアセットを使用して派生アセットやアセットのレンディションを作成する場合も同様です。

成分のもう一つの一般的な用途は、プロセスへの入力として使用された資産やデータを記述することです。例えば、AI/MLモデルに関連するトレーニングや推論のリクエストなどが該当します。

ingredients アサーションには 3 つのバージョンがあります。オリジナルの v1（ラベルは `c2pa.ingredient`）、改良版の v2（ラベルは `c2pa.ingredient.v2`）、さらに改良された v3（ラベルは `c2pa.ingredient.v3`）です。v3 は、編集後の成分の検証の問題に対処しています。

注記	複数の成分アサーションが存在することが想定されるため、単調増加型 増加する インデックス 増加 the ラベル は be 使用される (例:

c2pa.ingredient.v3、c2pa.ingredient.v3 1、c2pa.ingredient.v3 2）。

18.15.2. 一意の識別子の確立

追加される成分がC2PAマニフェストを含む場合、その一意の識別子は、当該成分の有効なC2PAマニフェストを含むJUMBFスーパーボックスのマニフェストラベルから取得され、成分アサーションのオプションフィールドinstanceIDを指定する必要はない。クレーム生成者が成分アサーションのオプションの instanceID フィールドを提供する場合、一意の識別子の値は、セクション 8.3「非 C2PA 資産の識別」で規定されているとおりに決定されるものとします。

注記 クレーム生成者は、成分に C2PA マニフェストがある場合でも、成分アサーションに instanceID フィールドを提供することができます。

18.15.3. 関係

材料を追加する際には、その材料と現在の資産との関係を記述すること。

関係フィールドの可能な値とその意味は、表10「成分の関係」に示されている。

表10. 成分の関係性

値	意味
parentOf	現在のアセットは、このイングリディエントの派生アセットまたはアセットレンディションです。この関係値は更新マニフェストでも使用されます。
componentOf	現在のアセットは複数のパーツで構成されており、この成分はその一つです。
inputTo	この成分は、AI/MLモデルなどの計算プロセスへの入力として使用され、このアセットの作成または変更につながりました。

成分アサーションを追加する際、クレームジェネレータは、アクティブなマニフェストにまだ存在しない場合、c2pa.actionsアサーション（セクション18.14「アクション」参照）を追加しなければならない。成分のタイプに応じて、以下のいずれかの新規エントリをc2pa.actionsアサーションのactions配列に追加しなければならない。

- 親要素を持つ関係で材料を追加する場合、c2pa.openedアクションを actions配列に追加される。
- componentOf 関係を持つ成分を追加する場合、c2pa.placed アクションを この要件は、アクションの記録がサポートされているマニフェストタイプであるスタンダードマニフェストにのみ適用されます。

この要件は、アクションの記録がそのマニフェストタイプでのみサポートされているため、標準マニフェストにのみ適用されます。

18.15.4. タイトル

dc:title が存在する場合、その値は素材の成分に対する人間が読める名称とする。この名称は、アセットの XMP またはローカルもしくはリモート（例：クラウドベース）ファイルシステムにおけるアセット名から取得できる。成分に固有の名称がない場合、代わりに成分の説明を使用できる。

18.15.5. Format

dc:format が存在する場合、その値は成分の IANA メディアタイプとする。クレーム生成者はこのフィールドを提供することが推奨され、有効な値を含めるものとする。AI/ML モデルのデータセットなど複数ファイルからなる成分を記述する場合、dc:format フィールドは multipart/mixed に設定するものとする。

18.15.6. スキーマと例

このタイプのCDDL定義は次の通り：

```
; アセットで使用される構成要素を記述するアサーション ingredient-map = {
  "dc:title": tstr, ; 成分名
  "dc:format": format-string, ; 成分のメディアタイプ
  ? "documentID": tstr, ; 成分の `xmpMM:DocumentID` の値 "instanceID": tstr, ; 一意の識別子、例えば成分の
`xmpMM:InstanceID`
```

```
   "relationship": $relation-choice, ; この成分が属するアセットとの関係
   ? "c2pa_manifest": $hashed-uri-map, ; 成分のC2PAマニフェストへのハッシュURI参照
   ? "thumbnail": $hashed-uri-map, ; 成分サムネイルへのハッシュ付きURI参照
   ? "validationStatus": [1* $status-map] ; 成分の検証ステータス
   ? "metadata": $assertion-metadata-map ; アサーションに関する追加情報
}

; バージョン2 (v2) の成分アサーション
; アセットで使用される食材を記述するアサーション ingredient-map-v2 = {
   "dc:title": tstr, ; 成分名
   "dc:format": format-string, ; 成分のメディアタイプ
   "relationship": $relation-choice, ; この要素が属するアセットとの関係
   ? "documentID": tstr, ; 成分の `xmpMM:DocumentID` の値
   ? "instanceID": tstr, ; 一意の識別子（例：成分の
`xmpMM:InstanceID` の値など
   ? "data" : $hashed-uri-map / $hashed-ext-uri-map, ; 埋め込みデータアサーションへのハッシュ付きURI参照、または外部データへのハッシュ付き拡張URI
   ? "data_types": [1* $asset-type-map], ; 成分V2構造に対するデータ型の追加情報。
   ? "c2pa_manifest": $hashed-uri-map, ; 成分のC2PAマニフェストへのハッシュURI参照
   ? "thumbnail": $hashed-uri-map, ; 埋め込みデータアサーション内のサムネイルへのハッシュ付きURI参照
   ? "validationStatus": [1* $status-map] ; 成分の検証ステータス
   ? "description": tstr .size (1..max-tstr-length) ; 成分の追加説明
   ? "informational_URI": tstr .size (1..max-tstr-length) ; 成分またはそのデータに関する情報ページへのURI
   ? "metadata": $assertion-metadata-map ; アサーションに関する追加情報
}

; 成分アサーションのバージョン3 (v3)
; アセットで使用される素材を記述するアサーション ingredient-map-v3 = {
   ? "dc:title": tstr, ; 成分名
   ? "dc:format": format-string, ; 成分のメディアタイプ
   "relationship": $relation-choice, ; この成分が属するアセットとの関係
   ? "validationResults": $validation-results-map, ; クレームジェネレータによる成分アセットの完全検証結果
   ? "instanceID": tstr, ; 成分の値など一意の識別子
`xmpMM:InstanceID` の値など
   ? "data" : $hashed-uri-map / $hashed-ext-uri-map, ; 埋め込みデータアサーションへのハッシュ付きURI参照、または外部データへのハッシュ付き拡張URI
   ? "dataTypes": [1* $asset-type-map], ; 成分V3構造に対するデータのタイプに関する追加情報
   ? "activeManifest": $hashed-uri-map, ; 成分のアクティブマニフェストに対応するボックスへのハッシュドURI
   ? "claimSignature": $hashed-uri-map, ; 成分のC2PAマニフェスト内のクレーム署名ボックスへのハッシュURI
   ? "thumbnail": $hashed-uri-map, ; 埋め込みデータアサーション内のサムネイルへのハッシュURI参照
   ? "description": tstr .size (1..max-tstr-length), ; 成分の追加説明
   ? "informationalURI": tstr .size (1..max-tstr-length), ; 成分またはそのデータに関する情報ページへのURI
   ? "softBindingsMatched": bool, ; ソフトバインディングが一致したかどうか
   ? "softBindingAlgorithmsMatched": [1* tstr] ; ソフトバインディングの発見に使用されたアルゴリズム名の配列
アクティブなマニフェストを発見するために使用されたアルゴリズム名の配列
```

```
   ? "metadata": $assertion-metadata-map ; アサーションに関する追加情報
}

format-string = tstr .regexp "^\\w+\/[-+.\\w]+$"

; 素材が資産と関連する理由を説明する選択肢
$relation-choice /= "parentOf"
$relation-choice /= "componentOf"
$relation-choice /= "inputTo"
```

CBOR診断表記法（RFC 8949、第8条）の例：

```
{
   "dc:title": "image 1.jpg", "metadata": {

      "reviewRatings": [
         {
            "value": 5,
            "explanation": "コンテンツバインディングの検証が完了しました"
         }
      ]
   }
   "dc:format": "image/jpeg","thumbnail": {
        "url": "self#jumbf=c2pa.assertions/c2pa.thumbnail.ingredient", "hash":
        b64'UjRAYWiAq4lfCRDmksWAlDJN/XtHHFFwMWymsZsm3j8='
   },
   "relationship": "parentOf", "activeManifest":
   {
        "url": "self#jumbf=/c2pa/urn:c2pa:5E7B01FC-4932-4BAB-AB32-D4F12A8AA322","hash":
        b64'1kjJTO108b71cL95UxgfHD3eDgk9VrCedW8n3fYTRMk='
   },
   "claimSignature": {


   },
   "validationResults": {  "activeManifest": {
        "success": [
        {
           "code": "claimSignature.validated",

        },{
           "code": "signingCredential.trusted",

        },{
           "code": "timeStamp.validated",

        },{
           "code": "timeStamp.trusted",

        },{
           "code": "assertion.hashedURI.match",
```

```
D4F12A8AA322/c2pa.assertions/c2pa.ingredient.v3"
        }
        ],
        "informational": [{
           "code": "signingCredential.ocsp.skipped",

        }],
        "failure": []
      },
      "ingredientDeltas": [{
        "ingredientAssertionURI": "self#jumbf=/c2pa/urn:c2pa:5E7B01FC-4932-4BAB-AB32-
D4F12A8AA322/c2pa.assertions/c2pa.ingredient.v3",
        "validationDeltas": { "success":
           [],
           "informational": [],
           "failure": [{
              "code": "assertion.hashedURI.mismatch",

        }]
        }
      },{
        "ingredientAssertionURI": "self#jumbf=/c2pa/urn:c2pa:F095F30E-6CD5-4BF7-8C44-
CE8420CA9FB7/c2pa.assertions/c2pa.ingredient.v3",
        "validationDeltas": { "success":
           [],
           "informational": [],
           "failure": [{
              "code": "signingCredential.untrusted",

        }]
        }
      }
      ]
   }
}
```

18.15.7. 説明フィールド

説明フィールドには、成分の性質や用途を自由記述で記載できます。タイトルや形式だけでは不十分な場合に有用です。

18.15.8. 成分データ

18.15.8.1. 標準的な使用法

生成AIなどの特定のユースケースでは、成分のデータが提供されることが重要となる場合があります。データはC2PAマニフェストに埋め込むか、データを参照するURLを介して提供されます。これは成分のデータフィールドを通じて実現され、埋め込みデータアサーションを指すにはハッシュ付きURIを、外部参照を指すにはハッシュ付き外部URIを使用します。

注記	本仕様の旧バージョンでは、`hashed_uri`がデータボックスを指すことが許可されていました。

埋め込みデータアサーションを使用すると、その内容は本C2PAマニフェストおよび将来のC2PAマニフェスト（編集されない限り）に埋め込まれ、当該資産を成分として含むことになります。クレーム生成者は、データを埋め込むかどうかの判断において、このフィールドのサイズを考慮すべきです。

例12. データ付き成分の例

CBOR診断表記法（RFC 8949、第8条）によるデータ付き成分の例：

```
// プロンプトのデータボックス //
{
   "dc:format": "text/plain",
   "data" : '肩に鳥を乗せた海賊'"dataTypes": [{
      "type": "c2pa.types.generator.prompt",
   }]
}

// 材料（プロンプト） //
{
   "dc:title": "プロンプト",
   "dc:format": "text/plain",
   "relationship": "inputTo", "data":
   {
      "url": "self#jumbf=c2pa.assertions/c2pa.embedded-data", "alg" :
      "sha256",
      "hash" : b64'...',
   }
}



// ingredient （モデル） //
{
   "dc:title": "model",
   "dc:format": "application/octet-stream","dataTypes": [
      {
         "type": "c2pa.types.generator",
      },
      {
         "type": "c2pa.types.model.tensorflow", "version":
         "1.0.0",
      },
      {
         "type": "c2pa.types.tensorflow.hubmodule", "version":
         "1.0.0",
      }
   ],
   "relationship": "inputTo","data": {
      "url": "https://tfhub.dev/deepmind/bigbigan-resnet50/1?tf-hub-format=compressed","alg" : "sha256",
      "hash" : b64'...',
   },
   "description": "ImageNetで訓練された、より小さなエンコーダアーキテクチャ（ResNet-50）を用いた教師なしBigBiGAN画像生成＆表現学習モデル。",
   "informationalURI": "https://tfhub.dev/deepmind/bigbigan-resnet50/1",
}
```

また、成分データに関する情報を特定することが重要であるものの、埋め込みも有効なURLの提供も不可能なユースケースも存在します。例えば、非公開/内部AIモデルの使用を説明するケースなどです。そのようなケースでは、data_typesフィールドの値としてアセットタイプを指定することで、そのデータの形式と説明をより明確にすることができます。

例13. data_typesを持つ成分の例

ハッシュ付きURIを持たない成分の例（CBOR診断表記法、RFC 8949、8項）：

```
// ingredient (private model) //
{
      "dc:title": "model",
      "dc:format": "application/octet-stream","relationship": "inputTo",
      "dataTypes": [
            {
                  "type": "c2pa.types.generator",
            },
            {
                  "type": "c2pa.types.model.tensorflow","version":
                  "1.5.0",
            }
      ],
      "description": "Joeのプライベート生成AIモデル","informationalURI":
      "https://www.example.com/joes-model-info.html"
}
```

18.15.8.2. 複数ファイル構成要素

場合によっては、AI/MLモデルのトレーニングデータセットなど、複数のファイルの集合として表現される構成要素があります。そのような場合、構成要素アサーションにC2PAマニフェストを含めることを推奨します。また、完全なデータセットのC2PAマニフェストには、それらのファイルの所在を参照するアセット参照アサーションを含める必要があります。

注記 この方法は、すべてのファイルが同じ階層構造に含まれていないアセットのコレクションを扱う場合に適しています。

18.15.9. 情報用URI

AI/MLモデルの詳細情報など、成分に関する情報を提供するウェブページのURLが必要な場合、そのURLは成分アサーションの`informationalURI`フィールドの値として配置する必要があります。

注記 `informationalURI`は、成分自体のコンテンツへの認証済みリンクではなく、より一般的に人間のユーザーにとって関心のあるものです。

注 古い（および非推奨の）バージョンの成分assertionでは、このフィールドがnamed `informational_URI`.

18.15.10. サムネイル

材料を追加する際、インポート時点での材料の状態を把握しやすくするため、材料のサムネイル画像も併せて含めることが有用である。この目的のため、サムネイル画像はサムネイルアサーションとして追加され、ハッシュ付きURI参照を介して本記述内で参照されるものとする。

マニフェストコンシューマーは、本仕様の以前のバージョンで推奨されていたデータボックス方式もサポートすべきである。

18.15.11. 既存のマニフェスト

18.15.11.1. 一般

コンポーネントに既存のC2PAマニフェストストアが存在する場合、当該コンポーネントのC2PAマニフェストストア内で検証済みであり、かつアセットのC2PAマニフェストストアに既に存在しないすべてのC2PAマニフェストは、クレームジェネレータによってアセットのC2PAマニフェストストアにコピーされるものとする。ただし、セクション18.15.12「既存マニフェストのコピー」に規定される場合、または（ユーザー入力や設定などにより）コピーしないよう指示された場合はこの限りではない。「既存マニフェストのコピー」で規定する場合、または（ユーザー入力や設定などにより）コピーしないよう指示された場合を除く。

クレーム生成器は、検証されなかった追加のC2PAマニフェスト、およびC2PAマニフェストストア内に存在するがC2PAマニフェストとして認識されない追加のJUMBFボックスやスーパーボックスも、アセットのC2PAマニフェストストアにコピーする必要があります。

注記	これらの追加要素をコピーすることで、カスタムアサーションや将来の

18.15.12. 既存マニフェストのコピー

18.15.12.1. 必要性の判断

既存のマニフェストが原料のC2PAマニフェストストアからアセットのC2PAマニフェストストアへコピーされる必要があるかどうかを判断するため、クレームジェネレータは以下の手順を実行する：

1. セクション18.15.12.4「成分の検証」に記載されたプロセスに従い、当該成分を検証する。検証に失敗した場合、クレーム生成器は指示がある場合（例：ユーザー入力または設定による）、以降のステップをスキップしてもよい。
2. 成分のC2PAマニフェストストア内の各マニフェストについて、そのURN識別子を、資産のC2PAマニフェストストアに既に存在する各C2PAマニフェストのURN識別子と比較する。
    a. 一致が見つかった場合、成分の C2PA マニフェストストアからのマニフェストボックスのハッシュを計算し、資産の C2PA マニフェストストアからの一致するマニフェストボックスのハッシュと比較する。
        i. ハッシュが一致する場合、クレーム生成器はイングリディエントのC2PAマニフェストストアからアセットのC2PAマニフェストストアへマニフェストをコピーしてはならない。
        ii. ハッシュが一致しない場合：

A. クレーム生成器は、いずれかのマニフェストからのアサーションが編集されたかどうかを確認する（必要に応じて「明示的な検証プロセスの実行」で作成された編集リストを利用する）。

I. 検証者が、ハッシュ値の差異が編集のみに起因すると判断できる場合：

1. アセットのC2PAマニフェストストアに既に存在するマニフェストに対して全ての編集が適用されている場合、クレーム生成器はコンポーネントのC2PAマニフェストストアからアセットのC2PAマニフェストストアへマニフェストをコピーしてはならない。

2. すべての編集が構成要素のマニフェストストアから取得したマニフェストに対して適用された場合、クレーム生成器は資産のC2PAマニフェストストア内のマニフェストを、構成要素のC2PAマニフェストストアから取得したマニフェストで置き換える。

3. 異なる編集が、成分のC2PAマニフェストストアと資産のC2PAマニフェストストアの両方から取得したC2PAマニフェストに対して適用された場合、クレーム生成器は、資産のC2PAマニフェストストア内の既存マニフェストから必要な数のアサーションを編集し、2つの編集セットの和集合となるようにしなければならない。

II. それ以外のすべてのケースでは、クレーム生成器は、成分のC2PAマニフェストストアからマニフェストをコピーし、「一意の識別子」で説明されているプロセスに従って更新されたURNで再ラベル付けし、再ラベル付けされたバージョンを資産のC2PAマニフェストストアに挿入しなければならない。

注記

ハッシュが編集による差異のみか否かを判定するプロセスは、バリデータに委ねられる。

18.15.12.2. 例

例：原料を輸入するクレーム生成器 D を考えます。まず原料 B を輸入し、原料 B 自体に原料マニフェスト A が含まれています。両マニフェストを検証後、クレーム生成器 D はマニフェスト B と A を資産 D の C2PA マニフェストストアにコピーします。次に原料Cを輸入する。原料Cにも編集済みバージョンのマニフェストAが含まれている。マニフェストCと編集済みマニフェストAを検証した後、マニフェストAの両バージョンのハッシュを比較する。原料Cに含まれるマニフェストAのバージョンが編集済みであることを認識したクレーム生成者Dは、資産DのC2PAマニフェストストアに既に存在するマニフェストAのバージョンを、原料Cからの編集済みマニフェストAで上書きする。

例：上記と同じシナリオを想定するが、成分C内のマニフェストAのバージョンが検証に失敗した（アサーションの1つがハッシュ比較に失敗したため）。この状況では、クレーム生成器Dは成分Cからマニフェスト Aをコピーし、新しいURNで再ラベル付けし、再ラベル付けされたコピーを資産DのC2PAマニフェストストアに配置する。

注記

C2PAマニフェストストアには、C2PAマニフェストではないJUMBFボックスやスーパーボックスが含まれる場合があります。これらは本プロセスの一部としてコピーする必要はありません。

18.15.12.3. 成分アサーションへのマニフェスト参照の追加

成分のアクティブマニフェストがアセットのC2PAマニフェストストアにコピーされている場合、成分アサーションの`active`Manifestフィールドの値として、その成分のアクティブC2PAマニフェストボックスへのURI参照を格納する。さらに、activeManifestのC2PAクレーム署名ボックスへの追加URI参照を`claimSignature`の値として格納する。

C2PA マニフェストストアに存在する C2PA マニフェストについては、`hashed_uri` を、構成要素アサーションの `activeManifest` フィールドと `claimSignature` フィールドの両方の値として使用しなければならない。

注記

両方の値を提供することで、効率的な成分検証が可能となり、また、成分のアサーションのいずれかが削除された場合の検証もサポートされます。

18.15.12.4. 成分の検証

18.15.12.4.1. 一般

さらに、成分アサーションがC2PAマニフェストを参照する場合、クレームジェネレータはバリデータとしても機能し、検証手順に記載された通り成分の検証を実行する。その検証結果（全成功コード、情報コード、失敗コード）は、以下に記載する通り成分アサーションのvalidationResultsまたはvalidationStatusフィールドの値として使用される。このフィールドは将来の検証で使用可能とするため必須である。

注記

検証ステータス（ingredient v2）または検証結果（ingredient v3）に失敗ステータスが含まれる場合、これはクレーム生成者が、アクターが成分のC2PAクレーム自体における検証エラーを認識し、それでも処理を続行することを選択した旨の明示的な表明と見なされる。
成分の組み込みを継続することを選択したことを明示的に表明したものとみなされる。

セクション15.3「マニフェスト情報の表示」で説明されているように、クレーム生成器は、欠陥のある来歴履歴を持つ資産を利用するアクターが、どのように進めるかについて情報に基づいた判断を下せるよう、警告を目立つように表示することが望ましい。

18.15.12.4.2. v2 成分アサーション（非推奨）

c2pa_manifestフィールドを持たないv2成分アサーションにおいて、validationStatusフィールドはオプションですが、存在する場合、空の配列を含む可能性があります。

v2形式のc2pa_manifestフィールドを含む要素アサーションにおいて、validationStatus配列の各オブジェクトは、マニフェストの特定部分の検証状態を記述するコードフィールドと、そのコードが成功（true）または失敗（false）を示すブール値のオプションの`success`フィールドで構成される。検証状態に関する追加の説明が必要な場合、説明フィールドに人間が読める説明を記載できる。さらに、各status-mapオブジェクトにはurlフィールドがあり、失敗の場合には、そのステータスが参照するマニフェスト内の特定要素へのJUMBF URI参照を含める必要があります。コードに応じて、`url`はC2PAクレーム、C2PAクレーム署名、または特定のC2PAアサーションを指します。ステータスコードはセクション15.2.2「標準ステータスコード」で定義されています。

クレーム生成者がプロセス固有のステータス情報を記録する必要がある場合、カスタムステータスコードの使用が許可される。コードはエンティティ固有名前空間（例：`com.litware.malformedFrobber`）と同じ構文に準拠し、`validation`Statusオブジェクトには成功を示すブール値を含めること。

18.15.12.4.3. v3 成分アサーション

v3イングリディエントアサーションにおいて`active`Manifestフィールドが存在しない場合、validationResultsフィールドは存在してはならない。

v3イングリディエントアサーションにおいてactiveManifestフィールドが存在する場合は、validationResultsフィールドは`validation-results-map`オブジェクトを含み、その内部には以下が含まれる：

1. `activeManifest`：当該成分のアクティブマニフェストに対する完全な検証結果。
2. `ingredientDeltas`：当該成分のC2PAマニフェストストア内の全マニフェストにおいて、activeManifestフィールドを含む各成分アサーションに対する差分検証結果。成分アサーションの差分検証結果には以下を含めること：
    a. `ingredientAssertionURI`: 成分アサーションのURI。
    b. `validationDeltas`では、ingredientアサーション（またはv1/v2のingredientアサーションの場合はvalidationStatusフィールド）のvalidationResultsフィールド内のactiveManifestフィールドに存在するステータス値を除外した、当該ingredientアサーションが参照するマニフェストの検証結果を格納する。このステータス値の比較では、ステータスタイプ（success、informational、failure）、コード、`URL`を考慮し、その他のフィールドは無視する。

例：複雑な系譜を持つ複数成分のマニフェスト`E`を考えます。クレーム生成器`E`は、成分アサーションを介してマニフェスト`C`とマニフェスト`D`を成分として追加します。マニフェストC自体には、成分アサーションを介してマニフェストAとマニフェストBが追加されている。マニフェストDにも、成分アサーションを介してマニフェストAが追加されている。マニフェストCを追加する際、クレーム生成器Eは検証結果オブジェクトを持つ成分アサーションを作成する。このオブジェクトは、`activeManifest`にCのアクティブマニフェストに対する検証結果を、`ingredientDeltas`にマニフェストAとBに対する差分検証結果を格納する。ingredientDeltas配列には2つの要素が含まれる：1つはマニフェストCの成分アサーションにおけるマニフェストBの検証結果オブジェクト内のactiveManifestオブジェクトとの比較による差分結果（マニフェストC内の当該成分アサーションへのハッシュ付きURIリンク付き）、もう1つはマニフェストCのマニフェストA成分アサーションに対する同様の属性を持つ要素。同様にマニフェストDを追加する際、クレーム生成器Eは、Dの`active`Manifestに対する検証結果と、Dのingredientアサーション内のManifest Aに対する検証結果オブジェクト内のactiveManifestオブジェクトとの比較によるデルタ検証結果を単一配列要素で保持する`ingredientDeltas`を格納するingredientアサーションを生成する。

> **注記**
> これは意図的に作り上げた例ではありますが、validationResultsデータ構造の使用方法に関する期待を明確にするために設計されています。

各検証結果（ステータスコードマップを用いて記述）は、成功コード、情報コード、失敗コードの配列で構成されます。各コードはステータスマップオブジェクトとして表現され、ステータスコードを含むコードフィールドを有します。さらに、特定のJUMBF URIへの参照を含む`URL`フィールドを含む場合があります。

マニフェスト内のステータスが参照する要素、およびステータスに関する人間が読める説明を含むオプションの説明フィールド。ステータスコードはセクション15.2.2「標準ステータスコード」で定義される。

クレーム生成者がプロセス固有のステータス情報を記録する必要がある場合、カスタムステータスコードの使用が許可される。コードはエンティティ固有の名前空間（例：`com.litware.malformedFrobber`）と同じ構文に準拠しなければならない。

18.15.13. 成分メタデータ

アサーションメタデータで説明されているように、構成要素アサーションのメタデータフィールドには、生成日時やアサーションデータの由来・真実性に関する判断材料となるその他のデータなど、構成要素に関するメタデータを含めることができます。

メタデータフィールドの一般的な用途の一つは、アセットの作成または編集において素材の一部のみが使用される場合です。このような場合、メタデータフィールドには、使用された素材の関連部分を記述するregionOfInterestフィールド（セクション18.3.6「関心領域」で説明）を含める必要があります。この例は例14「領域を含むメタデータを持つ素材の例」で確認できます。

NOTE

このフィールドには単一の関心領域しか含まれていませんが、region-mapオブジェクトでは以下を指定できます
複数の領域を`region`フィールドの値として指定できます。これは単一の素材の複数部分が関与する場合に有用です。

例14. 領域を含むメタデータを持つ食材の例

メタデータに領域オブジェクトを含む原料の例（CBOR診断表記法、RFC 8949 8条）：

```
{
  "dc:title": "someVideo.mp4", "metadata": {
    "regionOfInterest" : {
         "description": "10秒間の音声", "region": [
              {
                   "type": "temporal", "time": {
                        "type": "npt",
                        "start": "10",
                        "end": "20"
                   }
              },
              {
                   "type": "identified","item": {
                        "identifier": "track_id", "value":
                        "3"
                   }
              }
         ]
    }
  }
  "dc:format": "video/mp4",
```

```
    "relationship": "componentOf",
    "activeManifest" : {
         "url": "self#jumbf=/c2pa/urn:c2pa:98782815-5116-4d78-93de-3f5d8b4f4615",
         "hash":b64'TEWww2UCIR/e8mmR0XvzkFVZYTJ59Q8Ip4nkYxrS/Ys='
    },
    "claimSignature" : {

         "hash": b64'ICJkYzpmb3JtYXQiOiAiaW1hZ2UvanBlZyIsCiAgImR='
    },
    "validationResults": { ... }
}
```

18.15.14. ソフトバインディング

アクティブなマニフェストは、ソフトバインディング検索によって発見されたC2PAマニフェストを（parentOf関係を通じて）構成要素として含む場合がある。クレームジェネレータがそのようなC2PAマニフェストを含む場合、trueを示すsoftBindingsMatchedフィールドと、（構成要素であるC2PAマニフェストを発見するために使用されたソフトバインディングアルゴリズム名の文字列配列を含む）`softBindingAlgorithmsMatched`フィールドを含めなければならない。アルゴリズム名は、C2PAソフトバインディングアルゴリズムリストに、当該リストのエントリのalgフィールド内で特定される通り記載されなければならない。

18.16. メタデータ

18.16.1. 説明

この仕様の以前のバージョンでは、各メタデータ標準（例：IPTC、EXIF）ごとに個別のアサーションが存在していました。本バージョンでは、標準化されたシリアライゼーションでメタデータを表現するために使用されるアサーションのカテゴリが新たに設けられています。メタデータをアサーション内に配置することで、そのアサーション内のメタデータが重要であることを確立します。これは、C2PAマニフェストに明示的に含まれ、特定の署名者によって署名されているためです。これにより、データの暗号的検証と帰属が可能になります。さらに、共通のシリアライゼーションを使用することで、マニフェストの消費者が一貫した方法で処理できるようになります。

注記	これらのアサーションは既存の標準を表すこともあれば、プライベート仕様である場合もある。

18.16.2. 共通要件

メタデータアサーションは、文字列 `.metadata` で終わるラベルを有し、その前には標準の `c2pa` 識別子、またはエンティティ固有のネームスペースと同じ構文に準拠する任意の識別子が置かれる。例えば、`com.litware.metadata` アサーションは有効である。

各メタデータアサーションは、1つ以上のメタデータ値のJSON-LDシリアライズを含む単一のJSONコンテンツタイプボックスを含まなければならない。JSON-LDオブジェクト内の@contextプロパティを含め、指定されるメタデータ標準のコンテキスト／名前空間を提供するために使用しなければならない。このJSON-LDオブジェクトを作成する推奨手順は、まずメタデータのXMPデータモデル表現を作成し、XMPのJSON-LDシリアライゼーションに従ってそれをJSON-LDにシリアライズすることである。その後、JSON-LDをJSONコンテンツタイプボックスとして保存する。

18.16.3. c2pa.metadataアサーション

この仕様は、c2pa.metadata というラベルを持つ 1 つのメタデータアサーションを定義する。これは、あらゆる C2PA マニフェストで使用可能な共通メタデータスキーマのサブセットを表すために使用される。このアサーションに含まれる可能性のあるメタデータフィールドは、付録 B「*c2pa.metadata の実装詳細*」に記載されている。

> **注記** カスタムラベル付きメタデータアサーションは、任意のスキーマからの任意の値を含めることができる。

例15. 画像用のc2pa.metadataアサーション

画像用の c2pa.metadata アサーションの例：

```
{
     "@context" : {
          "exif": "http://ns.adobe.com/exif/1.0/", "exifEX":
          "http://cipa.jp/exif/2.32/", "tiff":
          "http://ns.adobe.com/tiff/1.0/",
          "Iptc4xmpExt": "http://iptc.org/std/Iptc4xmpExt/2008-02-29/", "photoshop" :
          "http://ns.adobe.com/photoshop/1.0/"
     },
     "photoshop:DateCreated": "Aug 31, 2022", "Iptc4xmpExt:DigitalSourceType":
"http://cv.iptc.org/newscodes/digitalsourcetype/digitalCapture", "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": "18:22:57",
     "exif:GPSDateStamp": "2019:09:22",
     "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",
     "exif:ExposureTime": "1/100", "exif:FNumber": 4.0,
     "exif:ColorSpace": 1,
     "exif:DigitalZoomRatio": 2.0, "tiff:Make":
     "CameraCompany", "tiff:Model": "Shooter S1",
     "exifEX:LensMake": "CameraCompany",
     "exifEX:LensModel": "17.0-35.0 mm",
     "exifEX:レンズ仕様": { "@list": [ 1.55, 4.2, 1.6, 2.4 ] }

}
```

例 16. PDF 用の c2pa.metadata アサーション

PDF用のc2pa.metadataアサーションの例:

```
{
```

```
      "@context" : {
         "dc" : "http://purl.org/dc/elements/1.1/", "xmp" :
         "http://ns.adobe.com/xap/1.0/","pdf" :
         "http://ns.adobe.com/pdf/1.3/", "pdfx":
         "http://ns.adobe.com/pdfx/1.3/"
      },
      "dc:created": "2015年2月3日","dc:title": [
            "これはテストファイルです"
      ],
      "xmp:CreatorTool": "TeX", "pdf:Producer": "pdfTeX-
      1.40.14", "pdf:Trapped": "Unknown",
      "pdfx:PTEX.Fullbanner": "This is pdfTeX, Version 3.1415926-2.5-1.40.14 (TeX Live 2013) kpathsea version
6.1.1"
}
```


18.16.4. `c2pa.metadata` の編集

編集プロセスは、アサーション全体のみを編集可能とする仕組みで動作する（セクション6.8「アサーションの編集」参照）。しかし更新マニフェストを使用することで、元のバージョンを削除し、更新マニフェスト内に新しい縮小版を配置することにより、部分的な編集が可能となる。この新しいアサーションは、編集されたC2PAマニフェストの署名者ではなく、更新マニフェストの署名者と関連付けてユーザーエクスペリエンスに提示されます。

たとえば、位置データとカメラ情報の両方を含むメタデータアサーションから位置データを編集する必要がある場合、カメラ情報のみを含む新しいメタデータアサーションをアップデートマニフェストで作成することで対応できます。

18.17. タイムスタンプ

18.17.1. 説明

一部のプロバンスワークフローでは、標準または更新マニフェストがオフラインで作成されるため、署名時にTSAから信頼できるタイムスタンプ（RFC 3161に準拠）を取得することが不可能です。しかし、そのような場合でも署名証明書は一定期間後に失効するため、C2PAマニフェストが無効化される事態を招きます。

この失効を防ぐため、後続の時点（証明書がまだ失効していない場合に限る）で信頼できるタイムスタンプを追加することで、当該C2PAマニフェストおよび（アクティブマニフェストの場合）関連資産に対する「存在証明」を提供できます。このタイムスタンプアサーションは、そのようなC2PAマニフェストに信頼できるタイムスタンプを提供するために使用されます。

18.17.2. スキーマと例

この型のスキーマは、以下のCDDL定義におけるタイムスタンプマップ規則によって定義されます：

```
; タイムスタンプ「ブロブ」へのマニフェスト URN の配列を格納するために使用されるデータ構造
; マニフェストURNからタイムスタンプ付き「ブロブ」へのマッピング
$time-stamp-map /= {
```

```
   * $$time-stamp-entry => bstr
}

タイムスタンプエントリ /= tstr .regexp "^urn:c2pa:[\\da-zA-Z_-]+$"
```

CBOR診断表記法（RFC 8949、第8条）の例を以下に示す：

```
{
   "urn:c2pa:d61c74e0-ce26-4439-b92d-690dcce6b58e" : h'...',
   "urn:c2pa:ab8c2751-8711-455a-9a8b-37143bfc92c2" : h'...'
}
```

18.17.3. 要件

タイムスタンプアサーションは `c2pa.time-stamp` のラベルを持つものとし、C2PA マニフェストごとに最大 1 つのタイムスタンプアサーションが存在するものとする。

タイムスタンプアサーションは、タイムスタンプマップとして定義されるCBORマップで構成され、少なくとも1つのキー値ペア（タイムスタンプエントリとして定義される）を含むものとする。キーは、ここで定義されるタイムスタンプ対象のC2PAマニフェストのC2PAマニフェストURNとし、値は後述する内容を持つCBORバイト文字列とする。

各タイムスタンプエントリの値は、RFC 3161準拠のタイムスタンプ機関（TSA）（RFC 3161）から応答として受信したTimeStampResp構造体のtimeStampTokenフィールドに存在するバイナリデータと同一でなければならない。タイムスタンプレスポンス自体は、セクション10.3.2.5.3「タイムスタンプの取得」に記載されたプロセスと同様の方法で取得するものとする。ただし、ペイロードの値は、タイムスタンプが付けられているC2PAマニフェストのC2PAクレーム署名ボックスに含まれるCOSE_Sign1_Tagged構造体の署名フィールドの値とする。

18.18. 証明書ステータス

18.18.1. 説明

一部のプロバンスワークフローでは、署名時に失効情報（OCSP経由）を取得できないオフライン環境で標準または更新マニフェストが作成される。検証プロセス中にその情報が利用できない場合、検証者は証明書の失効ステータスを確認するためにオンライン接続が必要となる。このアサーションは、事後的に情報を追加することで、そのようなC2PAマニフェストの信頼できる証明書ステータスを提供する。

18.18.2. スキーマと例

このタイプのスキーマは、以下のCDDL定義における`cert-status-map`ルールによって定義されます：

```
certificate-status-map = { "ocspVals": [1*
   bstr]
}
```

CBOR診断フォーマット（.cbordiag）の例を以下に示す：

```
{
  "ocspVals" : [ h'...',
    h'...'
  ]
}
```

18.18.3. 要件

証明書ステータスアサーションは、c2pa.certificate-status のラベルを持つものとし、C2PA マニフェストは、最大 1 つの証明書ステータスアサーションを含むものとする。

証明書ステータスアサーションは、CBORマップ（certificate-status-mapとして定義）で構成され、ocspVals配列に少なくとも1つのエントリを含まなければならない。セクション 14.5.2「証明書の失効」で説明されているように、クレーム生成器は、署名証明書によって示される OCSP サービスを照会し、応答をキャプチャし、rVals ヘッダーの ocspVals 配列の要素として保存される場合と同じバイナリ形式で保存しなければならない（例 3「rVals の CDDL」を参照）。

18.19. 資産参照

18.19.1. 説明

このアサーションは、アセットのコピーを入手可能な1つ以上の場所を示すために使用される。各場所はアセット参照アサーションを用いて記述されなければならない。場所はURIによって表現される。URIは単一のアセットを指す場合もあれば、ディレクトリを参照する場合もある。後者の場合、コレクションデータハッシュによってハッシュ化されるアセットの集合の場所を提供するために機能する。

注記

URIを表現することで、資産をWeb上の場所や分散型リポジトリから取得する柔軟性が提供される。ファイルシステム　　IPFS　（後者については　https://docs.ipfs.tech/how-to/address-ipfs-on-web/#subdomain-gateway を参照）。

資産参照アサーションは c2pa.asset-ref のラベルを持つものとする。

アサーションメタデータ内のタイムスタンプは、参照として記述されたリンクの新鮮さを判断するための基礎を提供する。

18.19.2. スキーマと例

このタイプのスキーマは、以下のCDDL定義におけるasset-ref-mapルールによって定義されます：

```
;アセット参照アサーション（ARA）は、アセットのコピーを入手できる場所を記述します。asset-ref-map = {
  "references": [1* ara-reference-block-map]
}
```

```
ara-reference-block-map = { "reference": ara-
   reference-uri-map,
   ? "description": tstr, ; 場所に関する人間が読める説明。
}

ara-reference-uri-map = {
   "uri": tstr, ; アセットのコピーが入手可能な場所を参照するURI
}
```

CBOR診断表記法（RFC 8949、第8条）の例を以下に示す：

```
{
   "references": [
      {
         "description": "Web上の資産のコピー","reference": {
               "uri": "https://some.storage.us/foo"
            }
      },
      {
         "description": "IPFS上のアセットのコピー", "reference": {
               "uri": "ipfs://cid"
            }
      }
   ]
}
```

18.20. アセットタイプ

18.20.1. 説明

アセットタイプアサーションは、アセットをより完全に記述する方法を提供します。具体的には、アセットの解析やその他の処理方法に関する追加のコンテキストを提供します。多くのアセットは単一のメディアタイプ値では完全に記述できない形式を持つため、このアサーションではIANAメディアタイプ値および/または追加のタイプ情報を指定できます。

アセットタイプアサーションのラベルは `c2pa.asset-type.v2` とする。C2PA マニフェスト内には、最大で 1 つのアセットタイプアサーションのみを含めること。

注記 本規格の以前のバージョンでは`c2pa.asset-type`アサーションが記載されていましたが、これは現在非推奨となっています。

`dc:format`フィールドが存在する場合、その値はアセットのIANAメディアタイプとする。

存在する場合、`types` フィールドの値は、アセットに関連付けられたタイプを指定する 0 個以上のマップ (`asset-type-map`) の配列とする。このマップの `type` フィールドの値は、表 11「アセットタイプ値」から取得するか、またはセクション 6.2.2「ラベルの命名」で定義されたアサーションラベルの構文に準拠したエンティティ固有のネームスペース（例：`com.litware.types.abc`）を使用する。関連する場合、アセットのバージョン（例：データセットやモデルのバージョン）は、`asset-type-map` の `version` フィールドに文書化できる。

注記

将来、C2PAがAI/ML（人工知能/機械学習）資産の由来情報を提供するために採用されるにつれ、C2PAマニフェストはモデルおよびデータセット資産に埋め込むことが可能となり、資産タイプアサーションを用いてこれらのモデルおよびデータセット資産のタイプを指定できるようになります。

表11. アセットタイプ値

C2PAタイプ	資産のC2PAタイプに関する説明
c2pa.types.dataset	複数のAI/MLフレームワークで処理可能なAI/MLデータセット、または他の値で記述されないもの
c2pa.types.dataset.jax	JAXデータセット
c2pa.types.dataset.keras	Keras データセット
c2pa.types.dataset.ml_net	ML.NET データセット
c2pa.types.dataset.mxnet	MXNet データセット
c2pa.types.dataset.onnx	ONNX データセット
c2pa.types.dataset.openvino	OpenVINO データセット
c2pa.types.dataset.pytorch	PyTorch データセット
c2pa.types.dataset.tensorflow	TensorFlow データセット
c2pa.types.model	他のモデルタイプで記述されていないAI/MLモデル
c2pa.types.model.jax	JAXモデル
c2pa.types.model.keras	Kerasモデル
c2pa.types.model.ml_net	ML.NETモデル
c2pa.types.model.mxnet	MXNetモデル
c2pa.types.model.onnx	ONNXモデル
c2pa.types.model.openvino.parameter	OpenVINO モデルパラメータ
c2pa.types.model.openvino.topology	OpenVINOモデルトポロジー
c2pa.types.model.pytorch	PyTorchモデル
c2pa.types.model.tensorflow	TensorFlowモデル
c2pa.types.numpy	シリアル化された NumPy 形式で保存
c2pa.types.protobuf	Protocol Buffer 形式で保存
c2pa.types.pickle	Pythonのpickle形式で保存
c2pa.types.savedmodel	TensorFlow SavedModel 形式で保存

18.20.2. スキーマと例

このタイプのスキーマは、以下のCDDL定義における`asset-types`ルールによって定義されます：

```
; アセットタイプアサーションは、アセットのタイプまたはフォーマットを記述する方法を提供します。
; 具体的には、その解析やその他の処理方法に関する追加コンテキストを提供します。
; また、AI/MLモデルなどの外部参照資産や関連資産を記述するためにも使用できます。

$type-choice /= "c2pa.types.classifier"
$type-choice /= "c2pa.types.cluster"
$type-choice /= "c2pa.types.dataset"
$type-choice /= "c2pa.types.dataset.jax"
$type-choice /= "c2pa.types.dataset.keras"
$type-choice /= "c2pa.types.dataset.ml_net"
$type-choice /= "c2pa.types.dataset.mxnet"
$type-choice /= "c2pa.types.dataset.onnx"
$type-choice /= "c2pa.types.dataset.openvino"
$type-choice /= "c2pa.types.dataset.pytorch"
$type-choice /= "c2pa.types.dataset.tensorflow"
$type-choice /= "c2pa.types.format.numpy"
$type-choice /= "c2pa.types.format.protobuf"
$type-choice /= "c2pa.types.format.pickle"
$type-choice /= "c2pa.types.generator"
$type-choice /= "c2pa.types.generator.prompt"
$type-choice /= "c2pa.types.generator.seed"
$type-choice /= "c2pa.types.model"
$type-choice /= "c2pa.types.model.jax"
$type-choice /= "c2pa.types.model.keras"
$type-choice /= "c2pa.types.model.ml_net"
$type-choice /= "c2pa.types.model.mxnet"
$type-choice /= "c2pa.types.model.onnx"
$type-choice /= "c2pa.types.model.openvino"
$type-choice /= "c2pa.types.model.openvino.parameter"
$type-choice /= "c2pa.types.model.openvino.topology"
$type-choice /= "c2pa.types.model.pytorch"
$type-choice /= "c2pa.types.model.tensorflow"
$type-choice /= "c2pa.types.regressor"
$type-choice /= "c2pa.types.tensorflow.hubmodule"
$type-choice /= "c2pa.types.tensorflow.savedmodel"
$type-choice /= tstr .regexp "([\\da-zA-Z_-]+\\.)+[\\da-zA-Z_-]+"

asset-type-map = {
   "type": $type-choice, ; リストされた選択肢のいずれか、またはカスタム値
   ? "version": tstr .regexp "^(0|[1-9]\\d*)\\.(0|[1-9]\\d*)\\.(0|[1-9]\\d*)(?:-((?:0|[1-

]*))*))?(?:\\+([0-9a-zA-Z-]+(?:\\.[0-9a-zA-Z-]+)*))?$"
}

asset-types = {
   ? "dc:format": format-string, ; アセットの IANA メディアタイプ
   ? "types": [* asset-type-map], ; アセットに関連するタイプの集合
   ? "metadata": $assertion-metadata-map ; アサーションに関する追加情報
}
```

CBOR診断表記法（RFC 8949、第8条）の例を以下に示す。この例では、資産はバージョン2.11.0のTensorFlowモデルファイルであり、SavedModel形式で保存されている。

```
{
   "types":
   [
      {
         "type": "c2pa.types.model.tensorflow",
```

```
        "version": "2.11.0"
      },
      {
        "type": "c2pa.types.savedmodel","version":
        "2.11.0"
      }
    ]
}
```

18.20.3. typeの値の選択に関する詳細

資産の正確なタイプがIANAレジストリのアプリケーションタイプまたはテキストタイプ（JSON、CSV、XMLタイプを含む）で指定されている場合、この情報は資産タイプアサーションのdc:formatフィールドに含めるべきである。

例えば、資産がCSV形式のテキストファイルである場合、dc:formatフィールドはtext/csvとなります。

アセットタイプアサーションには、アセットのタイプに関する追加情報を提供するため、ダブリンコア形式とC2PA標準またはカスタムアセットタイプの両方が含まれる場合があります。他のアセットタイプと組み合わせてアセットタイプアサーションで一般的に使用される既存のダブリンコアタイプの一部は、表12「一般的なDCフォーマット」に示されています。

表12. 一般的なDC フォーマット

dc:format 値	資産のダブリンコアタイプに関する説明
application/json	JSON形式で保存
application/gzip	GZIP形式で保存
application/vnd.rar	RAR形式で保存
application/zip	ZIP形式で保存
application/octet-stream	任意のバイナリ形式で保存
text/csv	CSV形式で保存
text/plain	プレーンテキスト形式で保存
text/tab-separated-values	タブ区切り値 (TSV) テキスト形式で保存
text/xml	XML形式で保存

C2PA クレームの dc:format フィールドでは、+json や +zip などの IANA 構造化サフィックスもサポートされており、追加のタイプを指定できます。

一部の dc:format タイプは一般的に使用されていますが、IANA レジストリには指定されていません。以下の dc:format 値は、表13「追加フォーマット」に示すように、C2PA資産に対して有効です。

表13. 追加フォーマット

dc:format 値	資産のダブリンコアタイプに関する説明
application/x-hdf5	HDF5形式で保存
application/x-7z-compressed	7Z形式で保存

18.21. 深度マップ

18.21.1. 説明

深度マップアサーションは、カメラが撮影するシーンの 3D 記述を提供します。深度マップアサーションには、事前に計算された深度マップ、または後続の取り込みソフトウェアや表示ソフトウェアによって深度マップを計算するために後で使用できるデータ（例：左右のステレオ画像）が含まれる場合があります。

すべての深度マップアサーションは、`c2pa.depthmap` で始まるラベルを持ち、深度マップの種類を識別する第3セクションが続くものとします。

C2PA深度マップのアサーションは、光学的に取得されなければならない。例えば機械学習モデルを介した単一の2D画像からの推定によるものであってはならない。

18.21.2. GDepth深度マップ

GDepth深度マップアサーションは、確立されたGDepthフォーマットを活用して事前計算された深度マップをエンコードする。GDepth深度マップアサーションは`c2pa.depthmap.GDepth`のラベルを持つものとする。

このアサーションに格納されるデータのスキーマは、常に https://developers.google.com/depthmap-metadata/reference のスキーマを反映するものとする。

注記

64KBを超える場合でも、XMPのAPP1セグメントサイズ制限に対応するために設けられた制限であるため、GDepthデータを分割することには問題ありません。

18.21.3. スキーマと例

このタイプのスキーマは、以下のCDDL定義における`depthmap-gdepth-map`ルールによって定義される：

; キャプチャされたシーンのGDepth形式3D深度マップをエンコードするアサーションdepthmap-gdepth-map = {

"GDepth:Format": フォーマットタイプ, ; 深度マップデータを有効な浮動小数点深度マップに変換する方法を記述するフォーマット。現在の有効な値は 'RangeInverse' と 'RangeLinear' です

"GDepth:Near": float, ; 深度単位での深度マップの近距離値"GDepth:Far": float, ; 深度単位での深度マップの遠距離値

"GDepth:Mime": mime-type, ; 深度画像の内容を記述するbase64文字列のMIMEタイプ。例: 'image/jpeg' または 'image/png'",

"GDepth:Data": base64-string-type, ; Base64エンコードされた深度画像。developers.google.comのGDepthエンコーディングページを参照。深度マップは対応するカラー画像に合わせて伸縮される

? "GDepth:Units": 単位タイプ, ; 深度マップの単位。例: メートルの場合は 'm'、ミリメートルの場合は 'mm'

```
   ? "GDepth:MeasureType": depth-meas-type, ; 深度測定の種類。現在の有効な値は 'OpticalAxis' および 'OpticRay'
   ? "GDepth:ConfidenceMime": confidence-mime-type, ; 信頼度画像の内容を記述するBase64文字列のMIMEタイプ。例: 'image/png'。",
   ? "GDepth:Confidence": base64-string-type,                ; ベース64エンコードされた信頼度画像。詳細は
developers.google.comのGDepthエンコーディングページを参照。信頼度マップは深度マップと同じサイズであるべき
   ? "GDepth:Manufacturer": tstr .size (1..max-tstr-length), ; この深度マップを作成したデバイスの製造元
   ? "GDepth:Model": tstr .size (1..max-tstr-length), ; この深度マップを作成したデバイスのモデル
   ? "GDepth:Software": tstr .size (1..max-tstr-length), ; この深度マップを作成したソフトウェア
   ? "GDepth:ImageWidth": float, ; この深度マップに関連付けられた元のカラー画像のピクセル単位の幅。深度マップの幅ではありません。存在する場合、アプリケーションはカラー画像の拡大縮小、トリミング、回転時にこのプロパティを更新する必要があります。クライアントはこのプロパティを使用して、カラー画像に対する深度マップの整合性を検証します
   ? "GDepth:ImageHeight": float, ; この深度マップに関連付けられた元のカラー画像の高さ（ピクセル単位）。深度マップの高さではありません。存在する場合、アプリはカラー画像の拡大縮小、トリミング、回転時にこのプロパティを更新する必要があります。クライアントはこのプロパティを使用して、カラー画像に対する深度マップの整合性を検証します
   ? "metadata": $assertion-metadata-map, ; アサーションに関する追加情報
}

base64-string-type = tstr

$mime-choice /= "image/jpeg"
$mime-choice /= "image/png"

mime-type = $mime-choice .default "image/jpeg"confidence-mime-type =
$mime-choice .default "image/png"

$format-choice /= "RangeInverse"
$format-choice /= "RangeLinear"

format-type = $format-choice .default "RangeInverse"

; 単位はメートル（"m"）またはミリメートル（"mm"）で表される
$unit-choice /= "m"
$unit-choice /= "mm"
unit-type = $unit-choice .default "m"

$depth-meas-choice /= "OpticalAxis"
$depth-meas-choice /= "OpticRay"
depth-meas-type = $depth-meas-choice .default "OpticalAxis"
```

CBOR診断表記法（RFC 8949、第8条）の例を以下に示す：

```
{
   "GDepth:Far": 878.7,
   "GDepth:Data": "hoOspQQ1lFTy/4Tp8Epx670E5QW5NwkNR+2b30KFXug=","GDepth:Mime":
   "image/jpeg",
   "GDepth:Near": 29.3,
   "GDepth:Model": "CameraCompany Shooter S1","GDepth:Units": "mm",
   "GDepth:Format": "RangeInverse",
   "GDepth:Software": "Truepic Foresight Firmware for QC QRD8250 v0.01", "GDepth:Confidence":
   "acdbpQQ1lFTy/4Tp8Epx670E5QW5NwkNR+2b30KFXug=", "GDepth:ImageWidth": 32.2,
   "GDepth:ImageHeight": 43.6
```

```
  "GDepth:MeasureType": "OpticalAxis",
  "GDepth:Manufacturer": "CameraCompany",
  "GDepth:ConfidenceMime": "image/png",
}
```

GDepth仕様で定義されている通り、以下のフィールドは全てのGDepth深度マップアサーションに存在しなければならない:

- GDepth:Format;
- GDepth:Near;
- GDepth:Far;
- GDepth:Mime;
- GDepth:Data.

18.22. フォント情報

18.22.1. 説明

フォント情報アサーションは、名前、フォーマット、作成者属性、ライセンスなどの基本的なフォントメタデータが、暗号的に検証可能な方法でアセットに追加されることを保証するために使用されます。

フォント情報アサーションはラベル `font.info` を持つものとし、マニフェストごとに最大1つのフォント情報アサーションが存在するものとする。

18.22.2. スキーマと例

このタイプのスキーマは、以下のCDDL定義における`font-info-map`ルールによって定義されます：

```
; font.info アサーションのデータ。font-info-map = {
  "fullName": tstr, ; フォントの正式名称
  ; セマンティックバージョニング（semver）形式のバージョン。
  ? "version": tstr .regexp "^(0|[1-9]\\d*)\\.(0|[1-9]\\d*)\\.(0|[1-9]\\d*)(?:-((?:0|[1-
]*))*))?(?:\\+([0-9a-zA-Z-]+(?:\\.[0-9a-zA-Z-]+)*))?$",
  ? "versionUrl": ext-url-type, ; このフォントのバージョンに関連するリリースノートのURL。
  ? "releaseDate": tdate, ; このフォントバージョンのリリース日または公開日。 "familyName": tstr, ; フォントファミリー。
  "style": $font-style, ; フォントのスタイル（例：イタリック、レギュラー）。"weight": font-weight-map, ; フ
  ォントのウェイト（名前と値）。
  ; フォントの 'name' テーブルからの PostScript 名、ID 6。
  "postScriptName": tstr .regexp "^(?!.*[\\[\\]\\(\\)\\{\\}<>\\/%])[!-~]{1,63}$", ; ASCII 33-126 の文字から、以下の文字を
除いたもの：[](){}<>/%
  "format": $font-format-choice, ; このフォントのフォーマット。"copyrightNotice": tstr, ; こ
  のフォントに関連する著作権表示。
  ? "copyrightHolder": font-entity-map, ; フォントの著作権を保有するエンティティ。
  ? "copyrightYears": [1* font-copyright-year-range], ; 保有者が著作権を主張する年。
  ? "designers": [1* font-designer-map], ; フォントをデザインした個人。
 ? "designFoundry": font-entity-map, ; フォントをデザインしたファウンドリ。
```

```
   ? "sourceFoundry": font-entity-map, ; フォントを配布するファウンドリ。
   ? "identifier": tstr, ; フォントの内部識別子（フォントメーカーまたはベンダー向け）。
}

; フォントフォーマット
$font-format-choice /= "TrueType"
$font-format-choice /= "OpenType"

; 著作権年範囲
font-copyright-year-range = 1..9999

; フォントの太さ範囲
font-weight-range = 1..1000

; フォントの太さクラス記述子
$font-weight-class /= "Microline"
$font-weight-class /= "Hairline"
$font-weight-class /= "UltraThin"
$font-weight-class /= "ExtraThin"
$font-weight-class /= "Thin"
$font-weight-class /= "UltraLight"
$font-weight-class /= "ExtraLight"
$font-weight-class /= "Light"
$font-weight-class /= "セミライト"
$font-weight-class /= "Book"
$font-weight-class /= "Normal"
$font-weight-class /= "Regular"
$font-weight-class /= "Medium"
$font-weight-class /= "DemiBold"
$font-weight-class /= "SemiBold"
$font-weight-class /= "Bold"
$font-weight-class /= "ヘビー"
$font-weight-class /= "ExtraBold"
$font-weight-class /= "UltraBold"
$font-weight-class /= "SemiBlack"
$font-weight-class /= "Black"
$font-weight-class /= "ExtraBlack"
$font-weight-class /= "UltraBlack"
$font-weight-class /= "MegaBlack"

; フォントスタイル
$font-style /= "Normal"
$font-style /= "Italic"
$font-style /= "斜体"
$font-style /= "Roman"
$font-style /= "Regular"

; フォントの太さに関するデータfont-
weight-map = {
   "class": $font-weight-class, ; ウェイトクラスの説明名（例：bold、thin）。
   "value": font-weight-range, ; ウェイトの値。
}

; 名前と認証情報を持つエンティティのデータ font-entity-map = {
   "name": tstr, ; 人物またはファウンドリーの名前。
   ? "url": ext-url-type, ; この人物または鋳造所に関する追加情報の URL。
}

; フォントデザイナーのデータ font-
designer-map = {
  "person": font-entity-map, ; フォントをデザインした人物。
```

```
  ? "foundry": font-entity-map, ; フォントデザインに貢献した際にデザイナーが所属していた鋳造所の名称。
  ? "contribution": tstr, ; デザイナーがフォントにどのような貢献をしたかの説明。例: 'ラテン文字とアラビア文字の全文字'。
  ? "startDate": tdate, ; デザイナーがフォントデザインへの貢献を開始した時期。
  ? "endDate": tdate, ; デザイナーがフォントデザインへの貢献を終えた時期。
}
```

必須フィールドのみを含む、CBOR診断表記法（RFC 8949、第8条）の基本的な例を以下に示す：

```
{
     "fullName": "Example Two Italic",
     "familyName": "ExampleTwo", "style":
     "Italic",
     "weight": {
          "class": "Regular", "value": 400
     },
     "postScriptName": "Example-Two-Italic","format":
     "TrueType",
     "copyrightNotice": "Copyright 2011 The Example Two Project Authors
(https://www.example.com/lifonts/Example-Two), with Reserved Font Name 'Example Two'.",
     "copyrightHolder": {"name":
          "Fabrikam"
     },
     "designers": [
          {
               "person": {
                  "name": "John Doe",
                  "url": "https://fabrikam.example.com/jdoefonts"
               }
          }
     ]
}
```

この拡張例では、オプションフィールドも示しています:

```
{
     "fullName": "Example Font Bold Italic", "version":
     "7.0.4-beta",
     "versionUrl": "https://fabrikam.example.com/release/efbi/7.0", "familyName":
     "ExampleFont",
     "style": "Italic",
     "weight": {
          "class": "Bold",
          "value": 700
     },
     "postScriptName": "ExampleFont-BoldItalic", "format": "OpenType",
     "copyrightNotice": "© 2017 Fabrikam, Inc. All Rights Reserved.","copyrightHolder":
     {
          "name": "ファブリカム株式会社"
     },
     "copyrightYears": [ 1982,
          2017
     ],
     "designers": [
```

```
          {
               "person": {
                  "name": "John Doe",
                  "url": "https://fabrikam.example.com/browse/designers/john-doe"
               },
               "foundry": {
                   "name": "Fabrikam Fonts"
               },
               "contribution": "合字。",
          },
          {
               "person": {
                  "name": "Jane Doe"
               },
               "フォントメーカー": {
                    "name": "ファブリカム・フォント"
               },
               "contribution": "全文字。"
          }
     ],
     "designFoundry": {
          "name": "Fabrikam Fonts",
          "url": "https://fabrikam.example.com"
     },
     "sourceFoundry": {
          "name": "Fonts Direct 2 U","url":
          "https://fd2u.example.com"
     },
     "identifier": "ExampleFont Bold Italic (Fabrikam)"
}
```

第19章 特許方針

C2PAはW3Cの特許モード（2004年）を通じてオープンスタンダード特許方針を採用しています：

ライセンス供与の約束。ワーキンググループが開発したソースコードまたはデータセット以外の素材について、各ワーキンググループ参加者は、W3C特許ポリシー（http://www.w3.org/Consortium/Patent-Policy-20040205）で定義される必須クレームを、当該ワーキンググループが採択した承認済み成果物において、あたかもその承認済み成果物がW3C勧告であるかのように、W3C RFライセンス要件セクション5（http://www.w3.org/Consortium/Patent-Policy-20040205）に基づき提供することを承諾する。ワーキンググループによって開発されたソースコードは、ワーキンググループ憲章に定められたライセンスの対象となります。

除外について。ドラフト成果物が承認済み成果物として採用される前に、作業部会参加者は、本契約に基づくライセンス供与の約束から必須クレームを除外する意思を書面による通知（「除外通知」）で作業部会議長に提供することにより、当該除外を行うことができる。発行済み特許及び公開出願に関する除外通知には、本特許方針第1条に定めるライセンス供与義務から除外を希望する発行済み特許又は係属中特許出願のそれぞれについて、該当する特許番号又はタイトル及び出願番号を記載しなければならない。必須クレームを含む可能性のある発行済み特許または係属中の特許出願が除外通知に記載されていない場合、それらの必須クレームは本契約に基づくライセンス提供義務の対象であり続ける。未公開特許出願に関する除外通知には、次のいずれかを記載しなければならない：(i) 出願された出願書類の全文、または (ii) 除外対象クレームを必須クレームとするドラフト成果物の特定部分の実施内容。(ii) を選択した場合、除外の効果はドラフト成果物の特定部分に限定される。除外通知はワーキンググループ議長が公開する。

付録A: マニフェストの埋め込み

A.1. サポートされている形式

C2PAマニフェストは、そのアセットのC2PAマニフェストストアの一部としてアセットに埋め込まれます。

C2PAマニフェストストアをアセットに埋め込む際、その位置はアセットの種類や形式によって異なります。以下に代表的なファイル形式と、それぞれのC2PAマニフェストストアの位置を示します：

JPEG

詳細はセクションA.3.1「JPEGへのマニフェスト埋め込み」を参照してください。

JPEG-XL

詳細はセクションA.3.8「JPEG XLへのマニフェスト埋め込み」を参照してください。

PNG

詳細については、セクション A.3.2「PNG へのマニフェストの埋め込み」を参照してください。

SVG

詳細については、セクションA.3.3「SVGへのマニフェストの埋め込み」を参照してください。

FLAC

詳細については、セクション A.3.4「ID3 へのマニフェストの埋め込み」を参照してください。

MP3

詳細については、セクション A.3.4「ID3 へのマニフェストの埋め込み」を参照してください。

GIF

詳細については、セクション A.3.7「GIF へのマニフェストの埋め込み」を参照してください。

DNG

詳細については、セクション A.3.5「TIFF ベースのアセットへのマニフェストの埋め込み」を参照してください。

TIFF ベースのフォーマット

詳細については、セクション A.3.5「TIFF ベースのアセットへのマニフェストの埋め込み」を参照してください。

WAVおよびBWF

詳細はセクションA.3.6「RIFFベースのアセットへのマニフェストの埋め込み」を参照してください。

AVI

詳細については、セクション A.3.6「RIFF ベースの資産へのマニフェストの埋め込み」を参照してください。

WebP

詳細については、セクション A.3.6「RIFF ベースの資産へのマニフェストの埋め込み」を参照してください。

その他の RIFF ベースのフォーマット

詳細については、セクション A.3.6「RIFF ベースの資産へのマニフェストの埋め込み」を参照してください。

フォント

詳細については、セクション A.3.9「フォントへのマニフェストの埋め込み」を参照してください。

PDF

詳細については、セクションA.4「PDFへのマニフェストの埋め込み」を参照してください。

EPUB

詳細については、セクション A.6「ZIP ベースの形式へのマニフェストの埋め込み」を参照してください。

OOXML

詳細については、セクション A.6「ZIP ベースの形式へのマニフェストの埋め込み」を参照してください。

Open Document

詳細については、セクション A.6「ZIP ベースのフォーマットへのマニフェストの埋め込み」を参照してください。

OpenXPS

詳細については、セクション A.6「ZIP ベースの形式へのマニフェストの埋め込み」を参照してください。

その他の ZIP ベースのフォーマット

詳細については、セクションA.6「ZIPベースの形式へのマニフェストの埋め込み」を参照してください。

MP4

詳細については、セクション A.5「BMFF ベースの資産へのマニフェストの埋め込み」を参照してください。

MOV

詳細については、セクション A.5「BMFF ベースの資産へのマニフェストの埋め込み」を参照してください。

AAC

詳細については、セクション A.5「BMFF ベースの資産へのマニフェストの埋め込み」を参照してください。

ALAC

詳細については、セクション A.5「BMFF ベースの資産へのマニフェストの埋め込み」を参照してください。

HEIF

詳細については、セクションA.5「BMFFベースのアセットへのマニフェストの埋め込み」を参照してください。

その他のBMFFベースのフォーマット

セクション A.5「BMFF ベースの資産へのマニフェストの埋め込み」で指定されているボックス。

注記

この仕様への追加が検討されている BMFF ベース以外のオーディオフォーマットには、Ogg Vorbis およびネイティブコンテナバージョンの Free Lossless Audio Codec (ネイティブ FLAC) があります。

A.2. マルチパートアセットへのマニフェストの埋め込み

マルチパート資産（「マルチアセット」）にC2PAマニフェストを埋め込む場合、資産のプライマリパート（アクティブマニフェストを含む）にC2PAマニフェストストアを埋め込む必要がある。ただし、追加のパートも独自のC2PAマニフェストストアを含むことが可能である。プライマリ部分のアクティブマニフェストには、資産内の各部分の場所とハッシュを記述するマルチアセットハッシュアサーションを含める必要があり、マルチパート資産全体の来歴を記述すべきである。

A.3. 非BMFFベースの資産へのマニフェスト埋め込み

A.3.1. JPEGへのマニフェスト埋め込み

C2PA マニフェストストアは、JPEG XT（ISO/IEC 18477-3）で定義されている **APP11** マーカーセグメントに含まれるデータとして埋め込まれるものとする。

JPEG 1における単一のマーカーセグメントは64Kバイトを超えることができないため、複数の**APP11**セグメントが必要となる可能性が高い。これらはJPEG 1規格およびISO 19566-5:2023 D.2に従って構築されなければならない。複数のセグメントを書き込む場合、それらは順番に書き込まれ、連続している必要があります（つまり、1 つのセグメントが次のセグメントに直ちに続く）。

A.3.2. PNGへのマニフェストの埋め込み

C2PAマニフェストストアは、補助的な非公開のコピー不可チャンクタイプ「`caBX`」（PNG 4.7.2に準拠）を使用して埋め込まれるものとする。caBXチャンクはIDATチャンクに先行することを推奨する。

注記

PNGはこれをサポートしているが、『`IDAT`』の後ろかつ『`IEND`』の前にデータブロックを配置することは好ましくない形式とみなされる（アニメーションPNGブロックは例外）。

A.3.3. SVGへのマニフェスト埋め込み

SVGはXMLベースのフォーマットであり、単独で存在することも、HTMLなどの他のテキストベースのフォーマットに埋め込むことも可能です。そのため、埋め込みを行うにはバイナリ形式のC2PAマニフェストストアをBase64エンコードする必要があります。本節ではその方法を説明しますが、外部マニフェストの使用が推奨されます。

C2PAマニフェストストアは、SVGのメタデータ要素内の`c2pa:manifest`要素のBase64エンコード値として埋め込む必要があります。XML、特にSVGでは名前空間の使用前に宣言を強く推奨するため、svg要素に`xmlns:c2pa = "http://c2pa.org/manifest"`属性の宣言を追加する必要があります。

例17. SVG内のC2PA マニフェストストアの例

SVG内のC2PAマニフェストストアの例（実際のC2PAマニフェストデータは省略）。

```
<?xml version="1.0" standalone="yes"?>
<svg width="4in" height="3in" version="1.1" xmlns =
     "http://www.w3.org/2000/svg" xmlns:c2pa =
     "http://c2pa.org/manifest">
     <metadata>
          <c2pa:manifest>...Base64データがここに記述される...</c2pa:manifest>
     </metadata>
</svg>
```

A.3.4. ID3へのマニフェスト埋め込み

C2PAマニフェストストアは、https://id3.org/id3v2.3.0で定義される一般カプセル化オブジェクト（GEOB）のカプセル化オブジェクトデータとして、ID3v2互換の圧縮オーディオファイル（例：MP3またはFLAC）に埋め込まれるものとする。GEOBのMIMEタイプフィールドは存在し、セクション11.4「外部マニフェスト」で説明されているJUMBFのメディアタイプ値を使用しなければならない。

A.3.5. TIFFベースのアセットへのマニフェストの埋め込み

デジタルネガティブ（DNG）フォーマットは、カメラメーカーが自社のカメラRAWフォーマットを標準化された方法で提供することを可能にします。DNGはTIFF/EP（それ自体がTIFFに基づいている）に基づいています。

C2PAマニフェストストアは、ID 52545（10進数）または0xCD41（16進数）のタグデータとして、タグタイプ7でTIFF互換ファイル（TIFF/EP、DNG、その他のTIFFベースRAW形式）に埋め込まれるものとする。

TIFFは複数ページやレイヤーの概念（複数のIFDを介して）をサポートしていますが、アセット全体に対してC2PAマニフェストストアは1つだけ存在し、IFDごとに複数存在してはなりません。したがって、C2PAマニフェストストアは、ファイルの末尾に直前に位置する最後のIFD内に存在する唯一のボックスでなければなりません。

注記 この仕様の以前のバージョンではIFD 0の使用が要求されていましたが、これによりTIFFベースのRAWフォーマットでの使用が制限されることが認識されました。

A.3.6. RIFFベースのアセットへのマニフェスト埋め込み

RIFF（リソース交換ファイル形式）は、タグ付きチャンクでデータを保存するための汎用コンテナ形式を提供する。主に画像、音声、動画などのマルチメディアを保存するために使用される。WAV、BWF、Broadcast Wave、AVI、WebPのコンテナ形式として機能する。

注記 RIFFはIFFと呼ばれる古い形式に基づいています。

C2PAマニフェストストアは、RIFF互換ファイル（WAV、AVI、WebPなど）内に、識別子C2PAを持つチャンクのデータとして埋め込まれるものとする。互換性確保のため、このC2PAチャンクはRIFFチャンクの末尾に配置されるものとする。

A.3.7. GIFへのマニフェスト埋め込み

C2PAマニフェストストアは、255バイト以下のサイズに分割され、以下に規定するC2PA専用アプリケーション拡張ブロック（GIF 26に準拠）内の連続したデータサブブロック（GIF 15に準拠）に埋め込まれるものとする。

注記

このC2PAアプリケーション拡張ブロックでは、アプリケーション認証コードは以下に使用されない。ブロックを生成したアプリケーションを認証するものではありません。代わりにブロックバージョンとして使用され、初期値はメジャーバージョン1、マイナーバージョン0に設定され、以下に規定される方法でエンコードされます。

拡張導入部：0x21 アプリケーション拡張ラベル：0xFF
ブロックサイズ：0xB
アプリケーション識別子：0x43, 0x32, 0x50, 0x41, 0x5F, 0x47, 0x49, 0x46（「C2PA_GIF」）アプリケーション認証コード：0x010000（0x[メジャーバージョン][マイナーバージョン]00） アプリケーションデータ：C2PAマニフェストストア。一連のデータサブブロックとしてエンコードされ、各ブロックは1バイトのサイズ情報に最大255バイトのデータが続く
ブロック終端記号：0x00（C2PA マニフェストストアの最後のデータサブブロックの後に追加） 数量：1

このブロックは、ヘッダーの後、最初の画像記述子ボックスの前に埋め込まれるものとする。

A.3.8. JPEG XL へのマニフェストの埋め込み

ISO/IEC 18181-2:2024の第4条に記載されている通り、JPEG XLはデータに対して2種類の異なるフォーマットをサポートする。JPEG 2000およびJPEG XSと互換性のあるボックス構造を使用する場合もあれば、ボックス構造を持たない直接的なJPEG XLコードストリームである場合もある。ボックス構造を使用するJPEG XLファイルは、最大1つのJUMBF（ジャンブ）スーパーボックス（ISO/IEC 18181-2:2024、条項9.3）を含み、その中にセクション11.1.4.2「マニフェストストア」で説明されるC2PAマニフェストを含むC2PAマニフェストJUMBFボックスを含まなければならない。コードストリームのみのJPEG XLファイルは、埋め込みC2PAマニフェストを含めることができない。

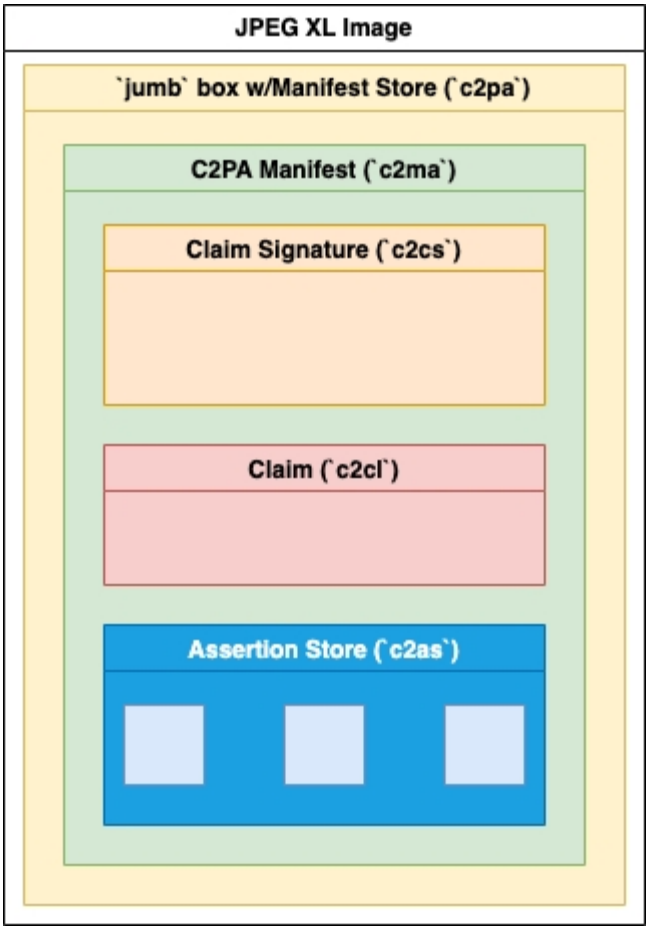

図19. JPEG XL画像に埋め込まれたC2PA マニフェスト

A.3.9. マニフェストのフォントへの埋め込み

Open Font Format または OpenType 仕様のいずれかに準拠するフォントは、`C2PA` テーブルを含めることができます。このテーブルが存在する場合、埋め込みマニフェスト、リモートマニフェスト URI、またはその両方が含まれる可能性があります。

`C2PA`テーブルの形式は、Open Font FormatおよびOpenType仕様のいずれにおいてもまだ定義されていません。以下の定義は暫定的なものです：

A.3.9.1. テーブルタグ

C2PAテーブルレコードは、以下のテーブルタグで識別されます：`C2PA`。

A.3.9.2. テーブルレコード

C2PAテーブルは、マニフェストストアが埋め込み型、リモート型、またはその両方となることを完全にサポートします。テーブルレコードは以下のように定義されます：

表14. C2PA テーブルレコード

タイプ	名前	説明
uint16	majorVersion	C2PAフォントテーブルのメジャーバージョンを指定します。
uint16	minorVersion	C2PA フォントテーブルのマイナーバージョンを指定します。
activeManifestUriOffset	activeManifestUriOffset	C2PA フォントテーブルの先頭から、アクティブなマニフェストへの URI を含むセクションまでのオフセット。URI が指定されていない場合は、NULL オフセット = 0x0000 を使用する必要があります。
uint16	activeManifestUriLength	URI の長さ（バイト単位）。
uint16	reserved	将来の使用のために予約されています。
Offset32	manifestStoreOffset	C2PA フォントテーブルの先頭から、C2PA マニフェストストアを含むセクションまでのオフセット。マニフェストストアが提供されない場合は、NULL オフセット = 0x0000を使用すること。
uint32	manifestStoreLength	C2PAマニフェストストアデータのバイト単位の長さ。

非埋め込みC2PAマニフェストは、リモートまたは同一ストレージシステム上のローカルに存在し得る。参照がJUMBF URIである場合、C2PAマニフェストストア内で有効な参照である必要がある。

A.4. PDFへのマニフェスト埋め込み

A.4.1. 一般

すべての C2PA マニフェストストアは、埋め込みファイルストリーム（ISO 32000、7.11.4）を使用して埋め込むものとします。埋め込みファイル仕様辞書は、値が `application/c2pa` のサブタイプキーと、値が `C2PA_Manifest` の `AFRelationship` キー（ISO 32000、7.11.3）を持つものとする。C2PA マニフェストストアが暗号化された PDF に埋め込まれる場合、埋め込みファイルストリームは `Identity` crypt フィルターを使用するものとする。

A.4.2. ドキュメントレベルのマニフェスト

A.4.2.1. PDFへのマニフェスト追加

PDF全体にC2PAマニフェストを追加する場合、ドキュメントカタログ辞書には、アクティブなマニフェストを含む埋め込みファイル仕様への間接参照を値とするAFエントリが含まれるものとする。その埋め込みファイル仕様は、間接オブジェクトを介して、`EmbeddedFiles` NameTree

（/Catalog/Names/EmbeddedFiles）またはFileAttachmentアノテーションのいずれかからも参照されるものとする。注釈アプローチは、既存のPDF認証署名が存在するPDFにC2PAマニフェストストアを追加する場合に使用され、そのDocMDP制限が無効化されるのを回避する。

注記 DocMDP辞書のPフィールドの値が*1または2の場合*、この種の変更は許可されない。値が3の場合のみ許可される。

他のほとんどの形式では、資産のすべてのC2PAマニフェストを含む単一のC2PAマニフェストストアのみが存在します。しかし、PDFの「増分更新」機能のため、代わりに単一のPDF内で複数のマニフェストをサポートする必要があります。このシナリオでは、ベースPDF内にあるC2PAマニフェストストアを初期マニフェストと見なし、最新の更新にあるものをアクティブマニフェストと見なします。C2PAマニフェストコンシューマーは、すべてのC2PAマニフェストストア内のC2PAマニフェストを、単一のC2PAマニフェストストアに含まれるものとして処理しなければならない。

注記 JUMBF URIは常に完全なURIであるため、特定のC2PAマニフェストから始まり、すべてのC2PAマニフェストは単一のC2PAマニフェストストアに含まれるものとみなされるため、PDF内でC2PAマニフェストストアを跨いで`parentOf`要素を参照するためにこのようなURIを使用することは許容される。

A.4.2.2. PDF署名との互換性

新しいC2PAマニフェストストアを追加する際には、PDF署名（認証または承認）も適用されるかどうかを把握する必要があります。PDF署名はC2PAマニフェスト署名後にPDFデータを変更するため、PDF署名辞書のContentsキーのサイズと位置はC2PA署名前に決定されるべきです。そのバイト範囲は、`c2pa.hash.data`アサーションの除外リストに追加する必要があります。これにより、PDF署名の追加によってC2PA署名が無効化されることを防ぎます。PDF署名は、関連するC2PAマニフェストストアを含むPDF全体に対して適用されるものとします。

注記 C2PAのクレーム署名に加えてPDF署名を追加することで、既存のPDFエコシステムとの互換性が向上する。

A.4.3. オブジェクトレベルのマニフェスト

PDF文書自体に対して文書レベルのマニフェストを介して出所情報を提供できることに加え、文書内の個々のオブジェクトにも関連付けられたC2PAマニフェストストアを設定できます。これは、オブジェクトのストリームまたは辞書に`AF`エントリを追加することで実現されます。AFエントリの値は、前述の方法で埋め込まれたC2PAマニフェストストアを含む埋め込みファイル仕様への間接参照でなければなりません。

この機能の最も一般的な用途は、埋め込み画像（ImageまたはForm XObjectとして）およびフォントの出所情報を提供することです。また、ISO 32000-2（14.13.1）の関連ファイル条項で説明されているように、オブジェクト（プロパティリスト経由）または構造要素にAFエントリを追加することで、特定のコンテンツ部分の出所情報を提供するためにも使用できます。

追加されるオブジェクトレベルのマニフェストは、アクティブなマニフェストから `componentOf` 要素として参照されることが推奨されます。これにより、C2PA マニフェストコンシューマーがアセットの出所情報の全チェーンを容易に追跡できるようになります。

一般的に、実際の情報リソースを表すストリームまたは辞書であれば、いかなるPDFストリームまたは辞書にもC2PAマニフェストを添付できる。どのストリームまたは辞書に`AF`エントリを付与すべきか不明確な場合、マニフェストは記述されたデータリソースを実際に格納するオブジェクトに可能な限り近接して添付されるものとする。

注記 ラスター画像を記述するC2PAマニフェストは、Image XObjectストリームに添付される。また、埋め込みフォントファイル用のマニフェストは、フォントファイルストリームに添付され、

フォント辞書には添付されない。

A.4.4. 例

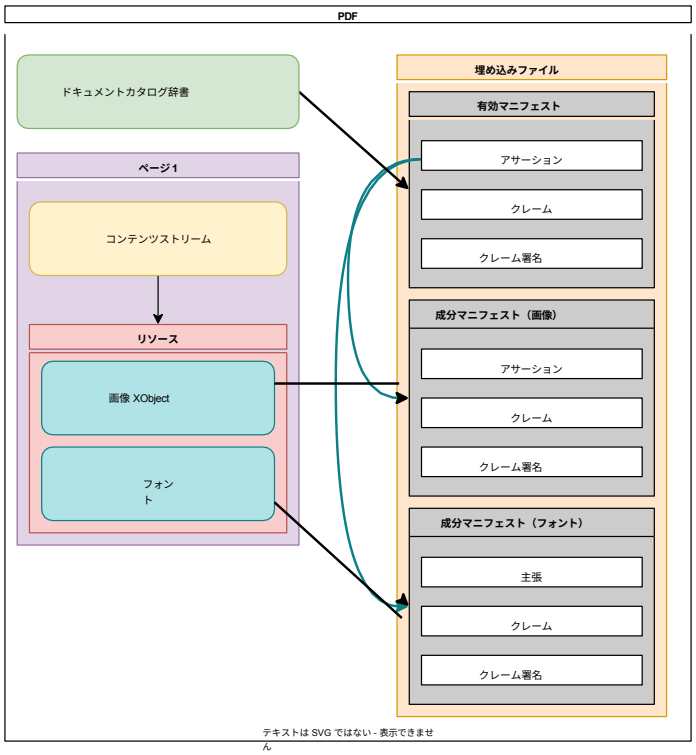

図20. 複数の成分マニフェストを含むPDFの例

A.5. マニフェストのBMFFベースのアセットへの埋め込み

A.5.1. C2PA用「uuid」ボックス

すべてのBMFFベースのC2PA資産（音声トラックの有無にかかわらず動画、静止画、ライブ写真やアニメーション写真などの混合メディアを含む）は、以下に定義する構文と意味論に準拠した「uuid」ボックスを使用しなければならない。

注記

「c2pa」ボックスではなく「uuid」ボックスが使用されている理由は、Chromiumベースのブラウザが未知のトップレベルボックスを検出すると即座に再生を失敗するためです。

この方法でサポートされるBMFFベースのファイル形式には以下が含まれます：

- MPEG-4コードポイント（完全版(.mp4)または断片化版(.m4s)）、ダウンロード可能なオーディオファイル(.m4a)、
- HEIF (.heif, .heic)；
- AVIF (.avif)

A.5.1.1. 定義

```
Box Type: 'uuid'
```

拡張ボックスタイプ: 0xD8, 0xFE, 0xC3, 0xD6, 0x1B, 0x0E, 0x48, 0x3C, 0x92, 0x97, 0x58, 0x28, 0x87, 0x7E, 0xC4, 0x81
コンテナ: ファイル 必須: いいえ
数量: ゼロ以上

C2PAの「uuid」ボックスはBMFFに由来情報を埋め込みます。1つのボックスにはC2PAマニフェストストアが含まれ、検証に必要な追加情報を含む補助ボックスが1つ以上存在する場合があります。

A.5.1.2. 構文

```
aligned(8) クラス ContentProvenanceBox は FullBox('uuid',extended_type = 0xD8 0xFE 0xC3 0xD6 0x1B 0x0E 0x48 0x3C 0x92
0x97 0x58 0x28 0x87 0x7E 0xC4 0x81, version = 0, 0) {
     string box_purpose;
     bit(8) data[];
}
```

A.5.1.3. 一意のIDについて

断片化されたMP4（fMP4）など、アセットの一部（例：'tkhd'ボックスのtrack_idフィールド）のIDが、アセット全体に対してグローバルに一意ではなく、アセットの一部に対してのみローカルに一意である場合があります。

ハッシュ化する対象を特定するためにグローバルに一意なIDが必要であるため、一意のIDが含まれます。この一意のIDは元の資産のいかなる値とも一致せず、代わりに各値はマニフェスト作成時に定義されます。その後、この一意のIDは関連するローカルIDと組み合わされ、資産全体に対してグローバルに一意なIDを形成します。

A.5.2. セマンティクス

各ボックス（box_purpose）の目的と、それに依存するフィールド（data）について、ボックスごとに以下で説明します。

A.5.3. マニフェストを含むボックス

C2PAマニフェストストアを含むボックスは、ファイル内の最初の'mdat'ボックスの前かつファイル内のいずれの'moov'ボックスの前にも配置されるものとする。major_brandおよびcompatible_brandの検証に対応するため、'ftyp'ボックスの後に配置されるものとする。アセットのアクティブマニフェストが更新マニフェストである場合、以前の標準C2PAマニフェストストアは上記のように配置され、box_purposeはoriginalに変更される。更新されたC2PAマニフェストストアは、ファイルの最後のボックスとして存在し、box_purposeはupdateに設定される。

上記で説明した対応するボックス内のフィールドは、以下のように設定する。

box_purpose

C2PA マニフェストストアの場合、この値は manifest、original または update とする。

data

box_purpose が manifest の場合、'data' 内の最初の 8 バイトは、box_purpose が merkle である最初の補助的な 'uuid' C2PA ボックスへの

絶対ファイルバイトオフセットとする。このファイルにそのようなボックスが含まれていない場合、それらの 8 バイトは

ゼロでなければならない。この8バイトの後に、生のC2PAマニフェストストアバイトが続き、その後ろにゼロ個以上の未使用パディングバイトが続く。box_purposeがoriginalの場合、box_purpose値がupdateに設定された別のC2PAボックスが存在することを示す。このoriginalボックス内の'data'は変更されない。box_purposeがupdateの場合、C2PAマニフェストストアには更新マニフェストのみが含まれる。

NOTE

タイプが manifest または original の 'uuid' ボックス内の 'data' フィールドには、絶対ファイルパスが含まれる。バイトオフセット、マニフェスト、およびパディングバイトを含む。originalボックスとmanifestボックスはbox_purposeの値を除き同一であるため、ハッシュバインディングは変更されない。'update'ボックスの追加によってハッシュ化されたデータが移動されることはない。

パディングバイトは、独自のmp4ボックス（例：'free'ボックスなどの独自のmp4ボックスに含まれている場合を除き、'uuid'ボックスの外側ではパディングバイトは許可されません。

フラグメント化された MP4 (fMP4) ファイルの場合、各初期化セグメントには、タイプがマニフェストの同一の 'uuid' C2PA ボックスが存在しなければなりません。C2PA マニフェストストアは同一でなければなりません。

A.5.4. 大規模およびフラグメント化ファイル用の補助 'c2pa' ボックス

A.5.4.1. 一般

一部のファイルには、1つ以上の非常に大きな「mdat」ボックス（例：プログレッシブにダウンロードおよびレンダリングされる可能性のある大きな動画や画像ファイル）または多数の独立した「mdat」ボックス（例：各フラグメントを個別にダウンロードできるfMP4）が含まれる場合があります。

このような場合、クライアントがアセットの任意の部分を検証する前に全ての 'mdat' ボックスを完全にダウンロードすることを要求するのは非合理的である。この必要性を回避する解決策として、複数のハッシュを使用する。

各大きな「mdat」ボックスには、個別に検証可能な個別のハッシュを持つサブセットが存在します。これらのサブセットの決定方法は以下に規定されています。各「mdat」ボックスが個別にダウンロード可能なfMP4コンテンツの場合、各フラグメントは独自の個別のハッシュを持ちます。

最も単純なケースでは、これらのハッシュはすべてアクティブマニフェストに格納されます。各サブセットには補助的な「uuid」C2PAボックスがあり、アクティブマニフェスト内でそのハッシュを特定する方法を宣言します。この仕組みの背景については、上記の一意のIDに関する注記を参照してください。

ただし、十分に大きな資産の場合、マニフェスト自体にすべての部分集合のハッシュを含めると、C2PAマニフェストストアのサイズが1メガバイト以上に増加します。

大きな資産に対してこのような大きな C2PA マニフェストストアを回避するには、1 つ以上のマークルツリーを使用します。

- 単一の大きなファイルに 1 つ以上の「mdat」ボックスを含む、断片化されていない大きなアセットの場合、各「mdat」ボックスに 1 つのマークルツリーが使用されます。

- 単一のトラックに対して複数のファイルに分散する可能性がある一連の「`mdat`」ボックスを含む、大きな断片化されたアセットの場合、トラックごとに 1 つのマークルツリーが使用されます。

いずれの場合も：

- 任意のマークル木の各葉ノードは、その部分集合のハッシュである。
- マニフェストは各マークルツリーの1行を格納します。
- 各部分集合に対して存在する補助的な「uuid」C2PAボックスは、アクティブなマニフェスト内のどのマークルツリー行を必要とし、どの葉ノードを表すかを示します。また、アクティブなマニフェストのマークルツリー行内のハッシュを導出するために必要な、マークルツリーからの追加のハッシュも含まれます。

マニフェストに保存するマークルツリーの行の選択は、資産内のサイズトレードオフを生じさせる。具体的には、マニフェストにマークルツリーごとに単一のハッシュを保存するとマニフェストのサイズが最小化されるが、各サブセット固有のボックスにlog2(サブセット数)を保存する必要がある。マニフェストに保存されるマークルツリーのハッシュ数が倍増するたびに（マークルツリーの行を「下」に移動することで）、各サブセット固有のボックスに保存されるハッシュ数は1つ減少します。したがって、マニフェストのサイズを大きくするとアセット全体のサイズが縮小し、その逆も同様である。また、マニフェストで指定されたハッシュを導出するために個々のサブセットのハッシュがサブセット間で複製される必要があるため、このトレードオフは1対1ではない。

このサイズのトレードオフは、マニフェストを作成する実装に委ねられる。本仕様は、特定のマークルツリーの行をマニフェストに格納することを義務付けも推奨もしない。ただし、すべての部分集合ハッシュをマニフェストに格納する最も単純なケースは、葉ノードがマニフェストに格納されるマークルツリーを使用することと同等であるため、いずれの場合も複数のハッシュに対して同じマークルツリー構造が用いられる。その構造は以下のように定義される。

マニフェストのBMFFハッシュを含む部分には、マークルフィールドを含めること。詳細はセクション9.2.3「BMFF形式アセットのハッシュ化」を参照のこと。

A.5.4.1.1. 部分的に検証可能な非断片化資産

マニフェストにマークルツリーの非リーフロウが含まれる場合、以下に記述する通りbox_purposeが'merkle'に設定された補助的な'uuid' C2PAボックスを2つ以上ファイルに含める必要がある。マニフェストにマークルツリーのリーフロウが含まれる場合、これらのボックスを含める必要はない。存在する場合は、ファイル内の最後の'mdat'ボックスに続く位置に配置しなければならない。

マークルツリー内の特定のリーフノードに使用されるハッシュは、'mdat'のペイロードのサブセットから計算されるものとする。'mdat'は、'fixedBlockSize'またはbmff-merkle-mapに定義される'variableBlockSizes'の配列によって定義されるサイズに分割され、'variableBlockSizes'の合計は'mdat'ペイロードのサイズと等しくなければならない。

このような補助的な「uuid」C2PAボックスはすべて、以下の要件を満たすものとする。

- それらは、「variableBlockSizes」フィールドで指定されるように、それらがハッシュするサブセットと同じ順序でなければならない。フィールドで指定される、ハッシュ処理対象のサブセットと同じ順序でなければならない。
- 単一のマークルツリーの補助的な「uuid」C2PA ボックスは、間に他のボックスが介入することなく連続しているようにグループ化される。
- 最初のボックスの位置値は0に設定され、2番目のボックスは1に設定され、

以降順次増加させる。

A.5.4.1.2. 断片化されたアセット

複数のファイルに分割されたfMP4資産の場合：

- 各フラグメントファイルには、下記の通り `box_purpose` が `'merkle'` に設定された補助的な `'uuid'` C2PA ボックスを 1 つ含め、これは `'moof'` ボックスの直前に配置されるものとする。
- マークルツリー内の特定のリーフノードに対して使用されるハッシュは、除外リストによって除外されたデータを除き、そのリーフノードを含む単一フラグメントファイル内の全データに対して生成されるものとする。

注記

本仕様は、個々のフラグメントファイルが複数の「`moof`」ボックスまたは「`mdat`」ボックス、あるいはその両方を含む、複数ファイルに分割された fMP4 アセットのサポートを可能とするものではない。

fMP4アセットは、単一のフラットMP4ファイルとして保存され、すべてのトラックに対して単一の「`moov`」と、各フラグメントごとに1組の「`moof`」と「/'mdat」ペアで構成されます。
各フラグメントごとに1組の`'mdat'`ペアを持つものについて：

- 各「`moof`」ボックスの直前に、以下で説明する通り`box_purpose`が「`merkle`」に設定された補助的な「`uuid`」C2PAボックスを1つ含めること。
- Merkleツリー内の特定のリーフノードに使用されるハッシュは、その'moof'ボックスと、次の'moof'ボックスまでの全データ（それ以降の`'moof'`ボックスが存在しない場合はファイル末尾までの全データ）に対して計算されるものとする。ハッシュは除外リストで除外されたデータを含んではならない。

重要な

複数のファイルに分割されたC2PA準拠のfMP4アセット（つまり、タイプが「`manifest`」および「`merkle`」の「`c2pa`」ボックスを持つもの）を、個々のファイルを単純に連結しても、C2PA準拠の単一ファイルは生成されません（逆も同様です）。これは、どの
各「`merkle`」ハッシュに含まれるボックスの種類が両ケースで異なるためです。両形式が必要とされる場合、後者の形式では前者を「材料」と見なし、新規マニフェストには親関係を示す`parentOf`属性を伴う材料アサーションと、`c2pa.repackaged`タイプのアクションを含むアクションアサーションの両方を含める必要があります。

A.5.4.1.3. マークル補助情報を含むボックス

アセットの構造にかかわらず、上記で説明した対応するボックスのフィールドは以下のように設定される。

box_purpose

補助的な「`uuid`」C2PAボックスの場合、この値は`merkle`とする。

data

`box_purpose` が `merkle` の場合、この値には、以下に定義される資産の一部を検証する方法を示す生の CBOR バイトが含まれるものとする。単一ファイル内の特定の Merkle ツリーに対して `box_purpose` が `merkle` の補助的な `'uuid'` C2PA ボックスが複数存在する場合は、各ボックスの後に十分なパディングバイト（ゼロ個以上）を付加し、その Merkle ツリーに対するすべての補助的な `'uuid'` C2PA ボック

スが固定サイズとなるようにするものとする。

注　　単一ファイル内にこれらのボックスが複数存在する場合、すなわち大規模な

> 'mdat'ファイルが部分的に検証されるため、クライアントが検証に必要なボックスのみをダウンロードできるよう、固定サイズが必要です。これにより、クライアントはMerkleツリー全体ではなく必要なボックスのみを段階的にダウンロードできます。このようなクライアントは、アクティブなマニフェスト内の絶対ファイルバイトオフセットに基づいて、最初のボックスを十分量ダウンロードし、そのuniqueIdとlocalIdが検証対象の'mdat'と一致するか判断できます。一致する場合、サブセット番号にそのサイズを乗じることで検証が必要なボックスへの絶対ファイルバイトオフセットを特定し、そのボックスのみをダウンロードできる。一致しない場合、固定サイズに現在のマークルツリーの葉ノード総数を乗じることで次のマークルツリーの先頭への絶対ファイルバイトオフセットを特定し、必要なボックスが見つかるまでこのプロセスを繰り返す。このサブセットのボックス群の総ダウンロードサイズは、単一サブセットのサイズに比べ極めて小さい。

A.5.4.2. スキーマと例

このタイプのスキーマは、以下のCDDL定義におけるbmff-merkle-mapルールによって定義されます：

```
; 単一の 'mdat' ボックスまたは

; マークルツリー使用時に'mdat'ボックスの一部を検証するのに十分な情報を格納するデータ構
造", bmff-merkle-map = {
   "uniqueId": int, ; ローカルIDを区別するための一意の整数"localId": int, ; マークルツリーを示すロ
   ーカルID
   "location": int, ; この 'mdat' ボックスまたはその一部に対応する、最下位マークルツリー行へのゼロベースのインデックス
   ? "hashes": [1* bstr], ; マニフェストで指定されたマークルツリー内のハッシュに到達するために必要な追加ハッシュの集合を表す順序付き配列。葉ノード（このノードの同位ノード）からルートノード（マニフェスト内のノードの子）への順序で記述される。この配列は存在しない場合があることに注意してください。例えば、マニフェスト自体がマークルツリーの最下位行を含む場合などです。ヌルハッシュはこの配列に含まれません。使用されるアルゴリズムは、BMFFハッシュ構造内の`merkle`フィールド配列に対応するエントリの`alg`フィールドを使用して決定されます。
}
```

CBOR診断表記法（RFC 8949、第8条）による例を以下に示す：

```
{
   "hashes": [
     b64'TWVub3JhaA=='
   ],
   "localId": 4402,
   "location": 2203,
   "uniqueId": 1339
}
```

非断片化アセットの場合、bmff-merkle-map内のlocalIdフィールドは'mdat'ボックスを示すものとする。これはファイル内の'mdat'の順序を示すゼロベースのインデックスである。断片化アセットの場合、bmff-merkle-map内のlocalIdフィールドは、ハッシュ対象の'mdat'に関連する'tkhd'ボックスのtrack_idフィールドに設定するものとする。

A.5.5. 動的ストリーム生成

多くの適応ビットレートストリーミング（ABR）実装では、単一バージョンのアセットを保存します。例えば、フラットMP4として、または

別の中間フォーマットとして保存し、消費時に様々なコーデックやビットレートなどを用いて個別のアセットストリームを生成する。その結果、当該サーバーは、コンテンツが消費されるたびに当該ストリームをハッシュ化しC2PAマニフェストを作成するか、生成が確定的である場合にはハッシュ値とC2PAマニフェストを一度作成してキャッシュし、消費時にそれらを埋め込むこととなる。

A.5.6. 除外リスト要件

すべての`c2pa.hash.bmff.v2`（非推奨）および`c2pa.hash.bmff.v3`アサーションにおいて、例18「常に除外されるボックス」のエントリは常に除外リストに記載されなければならない。その他のエントリは許可されるが必須ではない。

`'uuid'` C2PAボックス全体を除外する。（`'data'`フィールドは他の`'uuid'`ボックスが除外されないことを保証する。）

例18. 常に除外されるボックス

```
xpath = "/uuid"
data = [ { offset = 8, data = b64'2P7D1hsOSDySl1goh37EgQ==' } ]
```

`'ftyp'` および `'mfra'` ボックス全体を除外する。

```
xpath = "/ftyp"
```

```
xpath = "/mfra"
```

注記

この仕様の以前のバージョンでは追加の必須除外項目が含まれていましたが、それらを除外することは安全でないことが判明しました。

bmff-hash-mapがハッシュフィールドとマークルフィールドの両方を含むすべての`c2pa.hash.bmff.v2`（非推奨）および`c2pa.hash.bmff.v3`アサーションにおいて、例19「常に除外される追加ボックス」のエントリは除外リストに表示されるものとする。

例19. 追加の常に除外されるボックス

```
xpath = "/mdat"
サブセット = { { 16, 0 } }
```

注記

上記のCDDL定義で示されているように、`c2pa.hash.`bmffアサーションは`'mdat'`ボックス全体を除外しますが、これを除外することは安全でないことが判明しました。

上記のCDDL定義で示されているように、ボックスの長さを超える相対バイトオフセットまたは相対バイトオフセットと長さの合計は許可される。ボックスの末尾を超えるバイトは決してハッシュ化されない。例えば、`mdat`ボックスの長さがわずか12バイトの場合、その全てがハッシュ化され、前述の必須除外エントリは

依然として必須である。

A.5.7. 音声でも映像でもないタイムドメディアストリーム

音声でも映像でもないタイムドメディアストリーム（例：字幕用のテキストストリーム）で、クレーム生成者が改ざん防止を望むものは、音声・映像ストリームと同様に扱う。

A.5.8. 外部参照

BMFFボックス内（例：`'dref'`、`'url'`、'urn'ボックス）で宣言された外部参照コンテンツで、クレーム生成者が改ざん防止を望むものは、参照元ボックスを除外せず、ハッシュ化対象となる各外部参照ごとに個別のクラウドデータアサーションを含めるものとする。

A.5.9. サイズ要件

BMFFベースのアセットが任意のボックス（例：'stco'ボックス）で32ビットサイズまたはオフセットを使用し、例：`'stco'`ボックス。本仕様に準拠するためボックスを追加するとファイルサイズが4ギガバイトを超える場合、マニフェスト作成者はマニフェスト作成前にファイルを編集し適切なサイズとオフセットを使用する責任がある。例：'stco'ボックスを'co64'ボックスに置き換える。

A.6. ZIPベースのフォーマットへのマニフェストの埋め込み

A.6.1. 一般事項

長寿命であり、公開仕様であることから、多くのコマンドファイル形式は実際にはZIPアーカイブですが、内容ファイルが特定の構成で整理されています。これにはEPUB、Office Open XML、Open Document、OpenXPSなどの形式が含まれます。

A.6.2. ハッシュ処理

A.6.2.1. ファイルのハッシュ化

ZIPベースのファイル形式は、コレクションデータハッシュを用いてハッシュ化されるものとする。この際、ZIPに含まれる各ファイル（C2PAマニフェスト本体を除く）を含めること。コレクション内の各ファイルのハッシュは、ファイルのローカルファイルヘッダーに続いて圧縮および／または暗号化されたコンテンツ、ならびに存在する場合のデータ記述に対して計算される。使用するハッシュアルゴリズムは、コレクションデータハッシュ構造の`alg`フィールドで指定するものとする。

注記	圧縮/暗号化済みコンテンツに対してハッシュを計算する理由は、検証を可能にするためである。 解凍する必要性、または復号化キーの必要性。これはEPUBのように暗号化可能な形式において重要です。

A.6.2.2. ZIP中央ディレクトリのハッシュ化

ZIP AppNoteの4.3.12で説明されているように、中央ディレクトリは中央ディレクトリヘッダーの配列であり、ZIPアーカイブ内のファイルごとに1つずつ存在します。これはZIPアーカイブの末尾に格納され、ZIPアーカイブ内のファイルの位置を特定し、それらに関する必要な

情報/メタデータの検索に使用されます。その直後には、ZIPアーカイブ自体に関する情報を含む中央ディレクトリ終了レコード（ZIP AppNote、4.3.16）が続きます。

ZIP中央ディレクトリへの改ざん（新規ファイルの追加や既存ファイル情報の変更など）を防ぐため、ZIP中央ディレクトリ内の各「中央ディレクトリヘッダー」および「中央ディレクトリ終了レコード」はハッシュ化される。ハッシュは、コレクションデータハッシュ構造の `alg` フィールドで指定されたハッシュアルゴリズムを使用して、「中央ディレクトリヘッダー」の最初のバイトから「中央ディレクトリ終了レコード」の最後のバイトまでの範囲のバイトに対して計算される。

注 「中央ディレクトリヘッダ」は連続して格納され、その直後に「中央ディレクトリ終了レコード」が続く。

結果のハッシュ値は、コレクションデータハッシュ構造の `zip_central_directory_hash` フィールドに格納される。

注記 ファイル一覧に特別に命名されたファイルを使用する案が検討されたが、以下に説明する2パスシナリオのため採用されなかった。

```
; URI とそれに対応するハッシュの配列
$collection-data-hash-map /= { "uris": [1* uri-
   hashed-data-map],
   "alg": tstr .size (1..max-tstr-length), ; `uris`配列の各エントリのハッシュ計算に使用される暗号ハッシュアルゴリズムを識別する文字列。C2PAハッシュアルゴリズム識別子リストから取得。
     ? "zip_central_directory_hash" : bstr,
}

; URIとそのハッシュへの参照を格納するために使用されるデータ構造。
$uri-hashed-data-map /= {
   "uri": 相対URL型, ; 相対URI参照"hash": bstr, ; ハッシュ値を含むバイト列
   ? "size": size-type, ; データのバイト数
   ? "dc:format": format-string, ; データの IANA メディアタイプ
   ? "data_types": [1* $asset-type-map], ; データタイプに関する追加情報
}

; CBOR ヘッダー (#) およびテール ($) は正規表現で導入されるため、明示的に指定する必要はありません。relative-url-type /= tstr
.regexp "[-a-zA-Z0-9@:%._\\+~#=]{2,256}\\.[a-z]{2,6}\\b[-a-zA-Z0-9@:%_\\+.~#?&//=]*"
```

ZIPファイルはC2PAマニフェストの完成前に完了させる必要があるため、JPEG、BMFF、PDFで説明したように2段階アプローチを採用する。第1パスでは、内容がゼロ埋めされたcontent_credential.c2paファイルを含むZIPファイルを作成し、ZIP中央ディレクトリのハッシュを計算する。第2パスでは、zip_central_directory_hashフィールドの値を埋めることを含め、C2PAマニフェストを完成させる。

この2段階アプローチの一例として、以下の実装が考えられる：

- ゼロ埋めされたC2PAマニフェストストアファイル（置換可能な十分な容量）を含むZIPを作成する；
- ZIP中央ディレクトリのハッシュを計算する；
- ハッシュ値をコレクションデータハッシュマップのzip_central_directory_hashフィールドに追加する；

- マニフェストを完成させる；
- ゼロ埋めされた `content_credential.c2pa` ファイルを、完成したマニフェストデータで上書きします。

ZIPアーカイブ内で`content_credential.`c2paファイルを作成する際、圧縮方式0で保存し暗号化しないこと。汎用ビットフラグとCRC-32フィールドは0に設定する。日付時刻フィールドはZIPアーカイブ作成時刻、または0を設定可能。ファイルコメントを付加できる。

A.6.3. マニフェストストアの配置

C2PAマニフェストストアは、ZIPアーカイブのMETA-INFディレクトリ内に、ファイル名`content_credential.c2pa`、外部マニフェストに推奨されるメディアタイプで保存される。ファイルは保存（圧縮方式0）され、暗号化されない。

A.6.4. ZIPベース形式のデジタル署名

A.6.4.1. EPUB

EPUBのデジタル署名はW3C XML DigSig Coreに基づきます。署名対象の各ファイルは`<Signature>`要素の`<Manifest>`要素にリストされる。さらに、ZIP中央ディレクトリの署名サポートは存在しない。したがって、EPUBネイティブ署名はC2PAマニフェスト導入前に実施される。

A.6.4.2. Office Open XML

OOXMLのデジタル署名はW3C XML DigSig Coreに基づいています。署名される各ファイルは、`<Signature>`要素の`<Manifest>`要素内に`<Reference>`要素としてリストされます。さらに、ZIP中央ディレクトリの署名に対するサポートは存在しません。したがって、OOXMLネイティブ署名はC2PAマニフェストの導入前に実施される必要があります。

注記 OpenXPSはOOXMLと同じOpen Packaging Convention（OPC）標準に基づいているため、同様のアプローチが適用されます。

付録B: 実装の詳細

c2pa.metadata

c2pa.metadataアサーションには、以下に記述するスキーマとそのフィールドのサブセットのみを含めるものとする。ただし、カスタムメタデータアサーションは、これらまたは他のスキーマからの任意の値を含むことができる。

注記 有効な全スキーマとそのフィールドの機械可読リストは、C2PA仕様ウェブサイトで確認できる。

c2pa.metadataアサーションに存在する値は、メタデータアサーション固有のものでも、アセットフォーマットの標準「メタデータブロック」から取得されたものでもよい。いずれの場合も、ここで説明されているXMPのJSON-LDシリアライゼーション規則に従ってシリアライズされなければならない。

B.1. 完全にサポートされているスキーマ

表15「完全にサポートされているスキーマ」に記載されている以下のスキーマ/名前空間は、すべての署名者によって完全にサポートされています:

表15. 完全にサポートされているスキーマ

名前	名前空間
XMP Basic	http://ns.adobe.com/xap/1.0/
XMP メディア管理	http://ns.adobe.com/xap/1.0/mm/
XMP Paged-Text	http://ns.adobe.com/xap/1.0/t/pg/
Camera Raw	http://ns.adobe.com/camera-raw-settings/1.0/
PDF	http://ns.adobe.com/pdf/1.3/

B.2. 部分的にサポートされているスキーマ

表16「部分的にサポートされているスキーマ」に記載されている以下のスキーマ/名前空間は、一部のみサポートされています。

表16. 部分的にサポートされているスキーマ

名前	名前空間
ダブリンコア (DC)	http://purl.org/dc/elements/1.1/
IPTC Core	http://iptc.org/std/Iptc4xmpCore/1.0/xmlns/
IPTC Extension	http://iptc.org/std/Iptc4xmpExt/2008-02-29/
Exif	http://ns.adobe.com/exif/1.0/

ExifEx	http://cipa.jp/exif/1.0/exifEX
名前	**名前空間**
Photoshop	http://ns.adobe.com/photoshop/1.0/
TIFF	http://ns.adobe.com/tiff/1.0/
XMP Dynamic Media	http://ns.adobe.com/xmp/1.0/DynamicMedia/
PLUS	http://ns.useplus.org/ldf/xmp/1.0/

B.2.1. ダブリンコア (DC)

以下のダブリンコア（`dc`）プロパティのみがサポートされています：

- `dc:coverage`
- `dc:date`
- `dc:format`
- `dc:identifier`
- `dc:language`
- `dc:relation`
- `dc:type`

B.2.2. IPTC Core

以下の IPTC Core (`Iptc4xmpCore`) プロパティのみがサポートされています:

- `Iptc4xmpCore:Scene`

注記 一部の IPTC コアプロパティは、IPTC 拡張スキーマの新しいバージョンによって置き換えられています。

B.2.3. IPTC Extension

以下の IPTC Extension (`Iptc4xmpExt`) プロパティのみがサポートされています:

- `Iptc4xmpExt:DigImageGUID`
- `Iptc4xmpExt:DigitalSourceType`
- `Iptc4xmpExt:EventId`
- `Iptc4xmpExt:ジャンル`
- `Iptc4xmpExt:ImageRating`

- Iptc4xmpExt:画像領域
- Iptc4xmpExt:登録ID
- Iptc4xmpExt:LocationCreated

- `Iptc4xmpExt:LocationShown`
- `Iptc4xmpExt:MaxAvailHeight`
- `Iptc4xmpExt:MaxAvailWidth`

For 詳細 情報 について これらについての 参照 参照 https://www.iptc.org/std/photometadata/specification/IPTC-PhotoMetadata#xmp-namespaces-and-identifiers-2 を参照してください。

B.2.4. Exif

以下の表17「サポート対象のExifプロパティ」に記載されているExifプロパティのみがサポートされます：

表17. サポートされるExifプロパティ

- exif:ApertureValue
- exif:BrightnessValue
- exif:CFAPattern
- exif:ColorSpace
- exif:圧縮ビット/ピクセル
- exif:コントラスト
- exif:カスタムレンダリング
- exif:DateTimeDigitized
- exif:DateTimeOriginal
- exif:DeviceSettingDescr iption
- exif:デジタルズーム倍率
- exif:ExifVersion
- exif:露出補正値
- exif:露出指数
- exif:露出モード
- exif:露出プログラム
- exif:露出時間
- exif:ファイルソース
- exif:フラッシュ
- exif:フラッシュエネルギー
- exif:フラッシュピクバージョン
- exif:Fナンバー
- exif:焦点距離
- exif:焦点距離（35mmフィルム換算）
- exif:焦点面解像度単位
- exif:焦点面X解像度
- exif:焦点平面Y解像度
- exif:ゲイン制御
- exif:画像固有ID
- exif:ISO感度
- exif:LightSource
- exif:最大絞り値
- exif:測光モード
- exif:OECF
- exif:オフセット時間オリジナル
- exif:PixelYDimension
- exif:PixelYDimension
- exif:関連サウンドファイル
- exif:Saturation
- exif:SceneCaptureType
- exif:SceneType
- exif:SensingMethod
- exif:シャープネス
- exif:シャッタースピード値
- exif:空間周波数応答
- exif:スペクトル感度
- exif:被写体領域
- exif:被写体距離
- exif:被写体距離範囲
- exif:被写体位置
- exif:ホワイトバランス
- exif:GPS高度
- exif:GPS高度参照
- exif:GPS日付スタンプ
- exif:GPSDestBearing
- exif:GPSDestBearingRef
- exif:GPS目的地距離
- exif:GPS目的地距離参照
- exif:GPSDestLatitude
- exif:GPSDestLongitude
- exif:GPSDifferential
- exif:GPSDOP
- exif:GPSHPositioningErr または
- exif:GPSImgDirection
- exif:GPSImgDirectionRef
- exif:GPSLatitude
- exif:GPS経度
- exif:GPSMapDatum
- exif:GPSMeasureMode
- exif:GPSProcessingMethod
- exif:GPSSatellites
- exif:GPS速度
- exif:GPS速度基準
- exif:GPSStatus
- exif:GPSTimeStamp
- exif:GPSTrack
- exif:GPS速度
- exif:GPSVersionID

B.2.5. ExifEx

以下の ExifEx プロパティのみがサポートされています:

- exifEX:BodySerialNumber
- exifEX:Gamma
- exifEX:InteroperabilityIndex
- exifEX:ISOSpeed
- exifEX:ISOSpeedLatitudeyyy
- exifEX:ISO感度範囲
- exifEX:レンズメーカー
- exifEX:レンズモデル
- exifEX:レンズシリアル番号
- exifEX:レンズ仕様
- exifEX:感度
- exifEX:推奨露出指数
- exifEX:感度タイプ
- exifEX:標準出力感度

これらの詳細については、https://www.cipa.jp/std/documents/download_e.html?DC-010-2020_E を参照してください。

B.2.6. Photoshop

以下の Photoshop プロパティのみがサポートされています:

- photoshop:Category
- photoshop:City
- photoshop:ColorMode
- photoshop:Country
- photoshop:DateCreated
- photoshop:ドキュメントの祖先
- Photoshop:履歴
- photoshop:ICCプロファイル
- photoshop:状態
- photoshop:補足カテゴリ

- photoshop:テキストレイヤー
- photoshop:TransmissionReference
- photoshop:緊急度

B.2.7. TIFF

以下の TIFF プロパティのみがサポートされています:

- tiff:BitsPerSample
- tiff:Compression
- tiff:DateTime
- tiff:ImageLength
- tiff:ImageWidth
- tiff:Make
- tiff:Model
- tiff:方向
- tiff:測光解釈
- tiff:平面構成
- tiff:PrimaryChromaticities
- tiff:基準黒白
- tiff:解像度単位
- tiff:サンプル数/ピクセル
- tiff:Software
- tiff:転送関数
- tiff:白点
- tiff:X解像度
- tiff:YResolution
- tiff:YCbCr係数
- tiff:YCbCrPositioning
- tiff:YCbCrSubSampling

B.2.8. XMP Dynamic Media

以下のXMP Dynamic Media（xmpDM）プロパティのみがサポートされます（表18「XMP Dynamic Mediaプロパティ」参照）：

表 18. XMP Dynamic Media プロパティ

- xmpDM:absPeakAudioFileP ath
- xmpDM:album
- xmpDM:altTapeName
- xmpDM:altTimecode
- xmpDM:audioChannelType
- xmpDM:audioCompressor
- xmpDM:audioSampleRate
- xmpDM:オーディオサンプルタイプ
- xmpDM:beatSpliceParams
- xmpDM:cameraAngle
- xmpDM:cameraLabel
- xmpDM:カメラモデル
- xmpDM:カメラ移動
- xmpDM:contributedMedia
- xmpDM:contributedMedia
- xmpDM:duration
- xmpDM:ファイルデータレート
- xmpDM:genre
- xmpDM:good
- xmpDM:楽器
- xmpDM:introTime
- xmpDM:key
- xmpDM:logComment
- xmpDM:loop
- xmpDM:numberOfBeats
- xmpDM:マーカー
- xmpDM:outCue
- xmpDM:プロジェクト名
- xmpDM:プロジェクト参照
- xmpDM:pullDown
- xmpDM:相対ピークオーディオファイルパス
- xmpDM:相対タイムスタンプ
- xmpDM:releaseDate
- xmpDM:resampleParams
- xmpDM:scaleType
- xmpDM:scene
- xmpDM:撮影日
- xmpDM:撮影曜日
- xmpDM:撮影場所
- xmpDM:撮影名
- xmpDM:撮影番号
- xmpDM:撮影サイズ
- xmpDM:スピーカー配置
- xmpDM:開始タイムコード
- xmpDM:ストレッチモード
- xmpDM:テイク番号
- xmpDM:テープ名
- xmpDM:tempo
- xmpDM:時間スケールパラメータ
- xmpDM:拍子記号
- xmpDM:trackNumber
- xmpDM:トラック
- xmpDM:ビデオアルファモード
- xmpDM:videoAlphaPremultipleCol or
- xmpDM:videoAlphaUnityIs Transparent
- xmpDM:ビデオカラースペース
- xmpDM:ビデオ圧縮
- xmpDM:videoFieldOrder
- xmpDM:ビデオフレームレート
- xmpDM:ビデオフレームサイズ
- xmpDM:ビデオピクセルアスペクト比
- xmpDM:ビデオピクセル深度
- xmpDM:partOfCompilation
- xmpDM:歌詞
- xmpDM:discNumber

B.2.9. PLUS

以下の PLUS プロパティのみがサポートされています:

- plus:FileNameAsDelivered
- plus:初回公開日

- plus:ImageFileFormatAsDelivered
- plus:ImageFileSizeAsDelivered
- plus:ImageType
- plus:バージョン

これらの詳細については、http://ns.useplus.org/LDF/ldf-XMPSpecification を参照してください。

付録 C: 非推奨化に関する考慮事項

C.1. 構文要素のステータス

以下の表は、本仕様の進化に伴いステータスが変更された構文を一覧表示します。使用されるステータス値は

以下の通りです：

非推奨

Constructは非推奨です（クレーム生成器はこれを生成しないことが求められます；検証器はこれを受け入れることが推奨されます）。

UNDEFINED

構文が未定義です（バリデータはこれを無視する必要があります）。

<blank>

この構文は完全にサポートされています（バリデータはこれを受け入れる必要があります）。

表19. 構文要素のステータス

構文	タイプ	v1.3	v1.4	v2.0	v2.1	v2.2
タイムスタンプマニフェスト	マニフェスト	未定義	未定義	未定義		未定義
`urn:uuid` 名前空間	ラベル				廃止予定	廃止予定
`urn:c2pa` 名前空間	ラベル	未定義	UNDEFINED	未定義		
`c2pa.data` (データボックス)	ラベル					廃止予定
`c2pa.databoxes` (データボックスストア)	ラベル					廃止予定
`sigTst` timestamp	タイムスタンプ				廃止予定	非推奨
`sigTst2` タイムスタンプ	タイムスタンプ	未定義	未定義	UNDEFINED		
`c2pa.claim`	Assertion			非推奨	非推奨	非推奨
`c2pa.claim.v2`	アサーション	未定義	UNDEFINED			

c2pa.actio ns	アサート					
構文	**タイプ**	**v1.3**	**v1.4**	**v2.0**	**v2.1**	**v2.2**
c2pa.actions.v 2	アサート					
c2pa.asset -type	アサーション					非推奨
c2pa.asset -type.v2	アサーション	未定義	UNDEFINED	未定義	UNDEFINED	
c2pa.certific ate-status	Assertion	UNDEFINED	未定義	UNDEFINED	UNDEFINED	
c2pa.embed ded-data	Assertion	UNDEFINED	未定義	UNDEFINED	未定義	
c2pa.font.info	アサート	未定義		非推奨	非推奨	非推奨
c2pa.hash.bmff	アサーション	廃止予定	廃止予定	UNDEFINED	UNDEFINED	UNDEFINED
c2pa.hash.bmff .v2	アサーション				非推奨	非推奨
c2pa.hash.bmff .v3	アサーション	未定義	UNDEFINED	UNDEFINED		
c2pa.hash. collection .data	アサーション	UNDEFINED				
c2pa.hash. マルチアセット	アサーション	未定義	未定義	UNDEFINED	UNDEFINED	
c2pa.ingre dient	アサーション			非推奨	非推奨	非推奨
c2pa.ingre dient.v2	アサーション				廃止予定	廃止予定
c2pa.ingre dient.v3	アサーション	未定義	UNDEFINED	UNDEFINED		
stds.metad ata	アサート	未定義		非推奨	非推奨	非推奨
c2pa.metadata	アサーション	未定義	UNDEFINED			
c2pa.thumbnail .claim	Assertion	UNDEFINED	UNDEFINED	UNDEFINED	未定義	
c2pa.thumbnail .claim .*	アサーション					非推奨
c2pa.thumb nail.ingre dient	アサーション	未定義	UNDEFINED	UNDEFINED	UNDEFINED	

c2pa.thumbnail.ingredient.*	アサーション					非推奨
構文	**タイプ**	**v1.3**	**v1.4**	**v2.0**	**v2.1**	**v2.2**
c2pa.タイムスタンプ	アサーション	未定義	未定義	未定義	未定義	
font.info	アサート	UNDEFINED	未定義			
stds.iptc	アサーション		非推奨	廃止予定	非推奨	非推奨
stds.exif	アサーション		非推奨	廃止予定	非推奨	非推奨
stds.schema-org	アサート		非推奨	非推奨	非推奨	非推奨
role in 地域マップ	フィールド				廃止予定	非推奨
action-items-map-v2 のアター	フィールド			廃止予定	廃止予定	廃止予定
softwareAgents in actions-map-v2	フィールド	未定義	未定義	未定義		
softwareAgentIndex in action-common-map-v2	フィールド	未定義	UNDEFINED			
action-items-map-v2 で変更	フィールド				廃止予定	廃止予定
action-items-map-v2 の変更	フィールド	未定義	未定義	未定義		
instanceID in パラメータ-map-v2	フィールド				非推奨	非推奨
sourceLang 言語 パラメータ-map-v2	フィールド	未定義	未定義	未定義		

targetLanguage in parameters-map-v2	フィールド	未定義	未定義	未定義		
c2pa.trainedAlgorithmicData	DigitalSourceType					非推奨
Construct	**Type**	**v1.3**	**v1.4**	**v2.0**	**v2.1**	**v2.2**
http://c2pa.org/digitalsourcetype/trainedAlgorithmicData	DigitalSourceType	未定義	未定義	未定義	未定義	
http://c2pa.org/digitalsourcetype/empty	DigitalSourceType	未定義	未定義	未定義	未定義	