

AINET Genesis：署名付きObject交換の設計と単一ホスト評価

Two Brain

技術報告 TB-AINET-TR-001・改訂3・2026年10月31日・未査読

要旨

AINET Genesisは、既存のインターネット上で署名付き情報オブジェクトを交換するための実験的なアプリケーションプロトコルである。本報告は、バージョン0.1の設計と、単一Linuxホストでの探索的評価を記述する。設計では、内容、作者の鍵に基づく識別、出典に関する申告、公開・非公開の条件を不変のObject IDに結び付け、作者とコピーの提供ノードを区別する。取得時間、同時要求、手動再公開、プロセス強制終了、公開制御、Go・Python間のObject検証を評価した。6種類の入力はいずれも20回の測定ですべて取得でき、手動伝播では39区間にわたり署名Recordが一致した。一方、8並列の要求は96件中48件が失敗した。事後追加した待ち行列実験では、2・4ワーカーで各96件すべてに成功し、8ワーカーでは96件中12件にとどまった。目録アプリとローカルHTTP対照は同じ算術結果を返した。これらは限定した実装挙動の確認であり、Webに対する優位性、世界規模の耐障害性、AI間の意味の相互運用を示さない。今後の実証課題として、要求受け入れの制御、独立実装、複数ホストでの評価を挙げる。

1 序論

複数の情報源を利用するソフトウェアでは、接続先の識別、受信したバイト列の同一性、その情報についての主張を区別する必要がある。通信相手を認証しても、その相手が提供するコピーの元の作者が確認できるとは限らない。反対に、作者の署名は元のホストが停止しても検証できるが、それだけでは利用可能な別の提供元を発見できない。自動化されたクライアントの情報交換を設計する際、これらを別の問題として扱うことが重要である。

AINET Genesisは、不変の署名付きObject、明示的なローカル公開、認証済みノード間の直接交換を組み合わせた設計を検討する。将来の利用者としてAIシステムを想定するが、0.1にモデル推論は含まれない。本研究では、コピーを経てもObjectの識別と作者検証を維持できるか、取得の条件と失敗を観測可能にできるかを問う。さらに、必須の中央登録サービスを置かず、資料を選択・取得・引用するアプリケーションを構成できるかを調べる。

既存のインターネットとWebは、機械可読データ、アプリケーションプロトコル、署名、ミラー、分散サービスをすでに扱っている。Web全体に単一の中央サーバーがあるわけでもない。本研究はこれらの不足や機能的な不可能性を前提としない。AINETはIP上のアプリケーション層の試作であり、IPや物理回線の置換ではない。現段階の貢献は、小さな実行可能な設計と、記録を確認できるローカル実験をそろえた点にある。新規性や別の標準を設ける必要性の判断には、より広い先行研究の調査と、保証条件をそろえた比較が必要である。

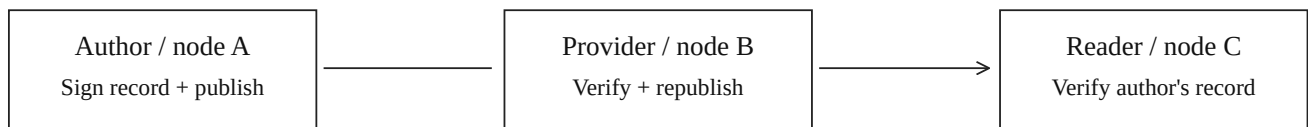
2 システムモデルと設計

2.1 識別、Object、提供元

各ノードはEd25519鍵ペアを永続化し、公開鍵からAINET IDを導出する。同じ鍵を用いて対応するlibp2pのPeer IDを構成する。接続相手は手動登録し、申告されたAINET IDと、通信で認証された鍵との対応を検証する。これは鍵の保有を認証するものであり、人名、組織への所属、利用権限の法的な妥当性を証明するものではない。

Objectは本文と署名付きRecordから成る。RecordはManifest、Object ID、Ed25519署名を含む。Manifestには本文のSHA-256、メディア型、正確なバイト数、作者IDと公開鍵、作成時刻、出典に関する申告、参照、公開条件を記録する。Object IDは用途を区別する接頭辞と決定的符号化されたManifestをハッシュして得る。このため本文だけのハッシュとは異なり、同じ本文でも作者・時刻・公開条件が変われば別のIDになる。署名は別の用途識別接頭辞を使い、作者の鍵とObject IDを結び付ける。接頭辞と符号化の厳密な定義は付属ワイヤ仕様に記載する。

検証時には作者IDと鍵の対応、Manifestの再符号化、IDの再計算、署名、本文のサイズ・ハッシュ・形式を確認する。実装は文章・PNG・JPEGを扱い、性能実験は文章とPNGを対象とした。現在の接続で認証された提供ノードと、Recordに記載された作者は異なってもよい。したがって、元の作者が接続中でなくてもコピーを検証できる。ただし、作者が記述した内容の真偽を証明する性質ではない。



Same signed record; different authenticated providers. Transfers are manually selected.

図1：作者と提供ノードの分離。コピーは手動で指定・再公開する。図は設計を表し、自動探索を示さない。

2.2 通信と公開制御

ノードはgo-libp2pのTCP、Noise、Yamuxを利用する。制御フレームは長さを制限した決定的CBORのEnvelopeを含み、Object本文はRecordの後に生のバイト列で続く。Base64には包まない。CBORの許可型を制限し、重複キー、非標準表現、未知のフィールドを拒否する。新たな暗号プリミティブは作らず、既存ライブラリを使用する。接続確認、Object取得、構造化メッセージを実装し、自動発見と分散問い合わせは今回の対象外とした。

新しく保存した資料は初期状態で非公開である。公開操作では別のIDを持つ署名付き公開版を作成する。受信したコピーは自動では配信せず、受信ノードでの明示的な再公開を要する。公開停止はそのノードからの新しい取得を拒否するが、他のノードに渡ったコピーを回収できない。署名された公開条件と、提供ノードの現在の配信判断は区別される。この2状態の方針は汎用の認可システムではなく、受信者別の暗号化や失効を実装していない。

保存にはノード別のローカルディレクトリを用いる。受信データを一時保存し、検証・正式保存を経て成功を応答する。鍵の保護はOSのファイル権限に依存し、同じOSユーザーと管理者を信頼する。メッセージ・本文のサイズ、通信時間、同時処理数には上限を設ける。これらは実装上の制御であり、悪意ある相手に対する包括的な安全性の実証ではない。任意コードの遠隔実行や、受信文章をシステム命令として解釈する仕組みは含まれない。

3 実験方法

実験は2026年10月31日、Intel Xeon Platinum 8573Cを搭載した単一のLinux x86-64環境で行った。プロセスから利用可能なCPUは5個と報告され、cgroupの実行時間上限は4コア相当、メモリ上限は16 GiBであった。各ノードを独立プロセスとして起動し、鍵と保存領域を分け、127.0.0.1上で通信した。これはプロセス単位の評価であり、独立した機械や地域を用いた試験ではない。元の研究と追加実験では同じGo実行ファイルを使用した。合成入力のseedは20261031である。ノードの鍵は別途生成し、配布成果物には残さなかった。

事前計画ではR1～R7を定義した。R1は3種類の文章（1 KiB、64 KiB、1 MiB）と、1辺512・1024・2048ピクセルの乱数RGB PNGを扱う。入力ごとに初回1回、準備2回、測定20回を実行し、反復内の入力順を乱数化した。R2は同じ受信ノードに1・2・4・8件を同時要求し、1 MiB文章と約3 MiB PNGについて条件ごとに6バッチ実行した。R1の初回・準備を含む要求とR2の要求は合計318行である。R2のバッチ終了後の単発復帰確認は別に記録した。

R3は64 KiB文章を、手動接続した3・5・8ノードの連鎖で各3回伝播させた。スクリプトが取得・検証・再公開した後、直前の提供元を停止した。R4は保存・公開完了後に5回、受信中の一時ファイルを観測した後に5回、SIGKILLを実行した。R5は20個の合成文章に6項目ずつの公開制御・変異検査を行った。R6はGoとPython間のObject検証および不正入力を評価した。R7は署名付き目録と決定的な集計アプリを作り、ローカルHTTPを対照とした。R2で過負荷時の拒否を観測した後、R8としてクライアント側の待ち行列を追加した。1バッチ32件、2・4・8ワーカー、各3バッチの計288要求であり、Go本体の変更や失敗時の再試行は行わなかった。R8は事後追加である。

取得時間にはCLI起動、ローカルIPC、HELLO、提供側の検証、転送、受信側の検証・保存を含む。回線の転送時間だけではない。同じObjectとデーモンを再利用し、OSキャッシュは消去していない。初回とは受信側にObjectがない状態を指し、ディスクやOSキャッシュが冷えた条件を意味しない。中央値、四分位、補間した観測p95は記録した試行群の記述であり、信頼区間や母集団への保証は導かない。同じバッチの要求や同一ホストの反復は、独立した配備環境の標本ではない。手順確認用のpilotは集計から除外した。

4 結果

4.1 取得時間と資源観測

R1の測定120要求はすべて成功した。表1に分布を示す。中央値は1 KiB文章の11.6 msから最大PNGの375.0 msまで増加した。512ピクセルPNGでは中央値39.4 msに対して最大391.6 msを観測した。この値は除外しておらず、原因も特定していない。乱数RGB PNGは圧縮しにくい合成資料であり、通常の写真や図を代表しない。また形式検証には画像復号が含まれるため、文章と画像の時間差をサイズだけで説明できない。

表1：R1の取得時間（ms）。各入力20測定、すべて成功。

入力	Bytes	中央値	Q1-Q3	p95	最大
Text 1 KiB	1,024	11.6	10.4–15.1	17.1	25.8
Text 64 KiB	65,536	13.2	12.5–14.2	19.5	20.0
Text 1 MiB	1,048,576	19.5	17.4–21.3	28.0	28.2
PNG 512 x 512	788,087	39.4	33.5–44.4	77.7	391.6
PNG 1024 x 1024	3,151,227	118.9	113.3–131.8	142.2	154.6
PNG 2048 x 2048	12,598,409	375.0	366.0–414.2	492.7	565.3

CPUカウンタは2つのノードデーモンを対象とし、CLIとPython測定器を含めない。小さな文章の中央値が0 msと記録されたのはカウンタの分解能によるもので、計算が不要という意味ではない。2ノードのRSS合計を約10 msごとに観測し、最大PNGの試行で最大197.7 MiBを記録した。デーモンを再利用しているため、過去の入力と保持された割り当てが後続試行に影響する。RSSを各入力だけに必要な増分メモリと解釈できず、観測間の短いピークを逃す可能性もある。関数別のプロファイルは取得しておらず、復号、ハッシュ、保存への寄与率は未確定である。

4.2 同時要求と待ち行列

R2では同時要求1件で12/12、2件で24/24、4件で48/48に成功し、8件では96要求中48件に成功した。失敗48件のうち47件はbusy、1件はHELLO authentication/versionエラーであった。後者は複数の失敗経路を同じ文言にまとめており、鍵の不正や正確な根本原因を特定できない。各バッチ後の単発取得48回はすべて成功した。実装は関連する送受信処理で4つの枠を共有するため、追加要求の拒否はこの受け入れ機構と整合する。

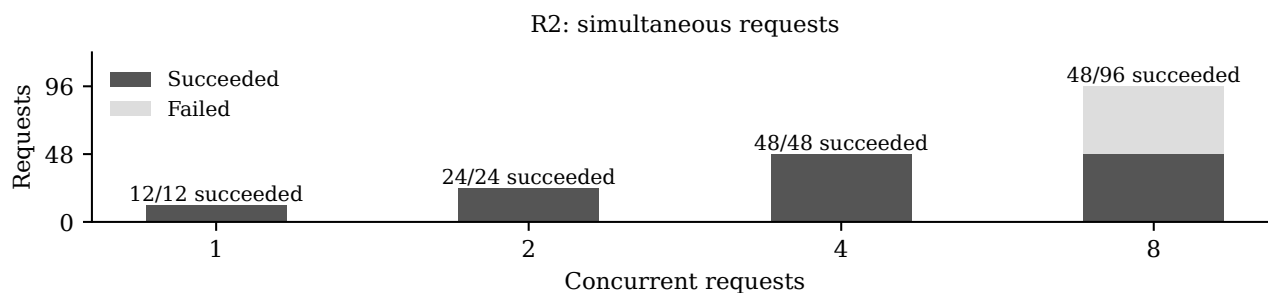


図2：R2の成功と失敗。2入力・各条件6バッチを集計。要求数は同時実行数によって異なる。

R8では2・4ワーカーが条件ごとに96件すべてを完了し、8ワーカーは12/96であった。実行中の仕事が処理枠を占有している間、拒否によって空いたクライアントが次の仕事を投入し、それも拒否され得る。この挙動は拒否が続く観測と整合するが、サーバーが仕事を高速に完了したことを意味しない。表2は、成功した仕事について、バッチ投入からの待ち時間を含む完了時間を示す。8ワーカーの中央値0.292秒は少数の成功分に条件付けられ、速度上の利点として扱えない。条件ごとに3バッチであり、2・4ワーカーのいずれかを普遍的に最適と結論付けるには不十分である。

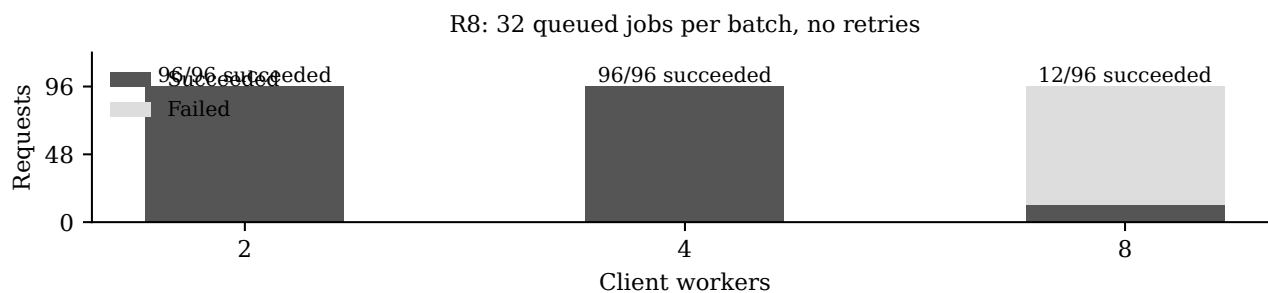


図3：R8の成功と失敗。約3 MiB PNG、各32件の3バッチ、再試行なし。図2とは別の負荷であり、改善比較ではない。

表2：R8の成功数と、成功分の待ち時間込み完了時間（秒）。

ワーカー	成功/要求	中央値	p95
2	96/96	1.385	2.489
4	96/96	1.429	2.431
8	12/96	0.292	0.385

4.3 コピー、復帰、検証

R3では3・5・8ノード構成について6・12・21区間、合計39区間で署名Record全体がバイト単位で一致した。9回すべてで、前の提供元が停止した後も最後のノードでObjectを検証できた。停止済み提供元への直接要求9回は想定どおり失敗した。これは手動で指定したコピーの検証であり、自動複製探索、経路切替、ホスト自体の障害への耐性を実証していない。

R4の完了応答後5試行は同じIDで復帰し、保存Objectを検証し、公開設定を維持した。受信中の5試行では停止前に一時ファイルを観測し、再起動前に正式Recordがないことを確認した。再起動による一時データの除去と再取得は成功した。ただし観測と停止は同時ではなく、最後に実行した命令は不明である。SIGKILL中もカーネルとファイルシステムは動作しており、電源断、媒体故障、すべての保存境界、ディスク満杯を試験したものではない。本文の正式配置後からメタデータ保存前までに停止すると孤立blobが残り得るため、後続の回収機構が必要である。

表3：主な機能観測。行ごとに集計単位が異なり、合計しない。

実験	観測数	確認内容
R3	39/39	署名Recordの一致
R4	5 + 5	完了後／受信中の停止から復帰
R5	120/120	公開条件・ローカル変異チェック
R6	6 + 20	Go→Python / Python→Go
R6	64 + 64 + 6	本文変異・署名変異・不正CBORを拒否

R5の120チェックは、非公開・不存在に対する同じ一般エラー、明示公開後の取得、受信コピーの自動公開禁止、公開停止後の拒否、ローカル本文変異と署名変異の拒否を反復した。同じエラー文言は時間差や通信量による漏洩がない証明ではない。変異は管理下の保存領域に加えた故障注入であり、遠隔攻撃の実証ではない。

R6のPython検証器は別の最小CBOR実装と既存のEd25519ライブラリを使用した。Go作成の6入力をPythonで、Python作成の20文章をGoで検証できた。Python側は本文変異64件、署名変異64件、不正CBOR6件を拒否した。Python作成資料にはRFC 8032の公開試験鍵を使用した。この鍵は試験専用であり、秘密の身元には使えない。相互運用の確認はObject層に限られ、PythonによるNoise・libp2pを含む完全な第二実装ではない。同じ開発支援者が両実装に関与したため、独立した作成・レビューの代替にもならない。元の研究全体では412/412チェックに成功したが、反復チェックと測定要求は別の集計単位である。

4.4 目録処理とHTTP対照

R7では文章Object内のJSONとして目録を表現した。合成温度18・20・22・24℃の4件を選択し、各Objectを取得して型と単位を確認し、平均21を根拠Object IDとともに構造化回答として返した。非公開の合成資料は目録から除外した。目録のリンクはアプリケーションJSONのデータであり、Manifest参照の新しいAPIや、プロトコルのDISCOVER・QUERYを実装したものではない。最初の目録IDと提供元は既知であった。

ローカルHTTP対照も同じ目録と値を配信し、平均21を返した。18を19に更新すると、対照は同じパスの表現を置換でき、AINETは新しいIDを作り、古いObjectには元の値を残した。ただしWebにも不変・版付きURL、ハッシュ、署名付き表現を設計できる。したがってこの対照はアプリケーション挙動の説明であり、Webの本質的な制約を示さない。HTTPS、署名、キャッシュ、永続化の条件は一致させておらず、性能の優劣も付けない。処理は固定の算術であり、モデル生成回答や意味理解の評価ではない。

5 考察と既存技術との関係

実験は、特定Objectの検証、提供元の可用性、内容の意味についての合意という3つの性質を分ける。R3は、手動選択された提供元をまたいでも最初の性質が成立するという限定的な証拠である。全体の可用性や意味の合意は扱っていない。R7では回答に根拠の識別子を含められたが、正しい参照があるだけでAIによる正しい解釈が示されるわけではない。今後の評価でも暗号層とアプリケーション層を区別する必要がある。

HTTPでも、署名付き表現、構造化データ、アクセス制御、版付き参照、複製をアプリケーション層に実装できる[1]。決定的CBOR[2]とEd25519[3]は既存の構成要素であり、それらの組合せやコピー後の作者署名の維持自体を新しい安全性の原理として扱わない。R3はこの実装の結合試験である。現時点で独自プロトコルの必要性は確立していない。

IPFSは内容に基づく識別子を用い[4]、IPNSは配信経路から独立して検証できる署名付きの更新可能な参照を提供する[7]。NDNはデータへの信頼と転送経路への信頼を分ける[8]。Nostr NIP-01は、作者が署名したイベントをリレーで配布し、置換可能・アドレス可能なイベントの規則も定める[9]。これらは作者と提供元を分ける点でAINETと重なる。AINETはそれらの探索・名前解決・更新機構を実装しておらず、鍵方式や符号化の差だけでは実用上の優位を説明できない。

COSE Hash EnvelopeはCOSEに本文ハッシュを示す標準方式であり、独自署名包装を置き換える候補となる[10]。同等のプロファイルには、メタデータ、公開判断、受領手順を別途定める必要がある。AINETの整数キーCBORとtag拒否は、文字列キーとCIDリンクを扱うDAG-CBORと異なり[11]、Object IDもCID互換ではない。当初の小さな固定スキーマには相互運用上の費用があるが、今回は測定していない。C2PAはメディアの署名付き来歴に関するより広い仕様であり[12]、AINETの自由文Provenanceは署名された申告にとどまり、独立して確認された編集履歴ではない。RDFも関係の表現を扱うもので、これらの信頼問題を一括して解決する方式ではない[5]。

次の比較では署名・検証・保存・キャッシュ・信頼の入力・資源条件をそろえる。同じObjectをAINETとHTTPSで運ぶ比較と、同等のメタデータを独自包装とCOSEで表す比較を分離する。標準方式で要件を満たし、独自方式の利点が認められなければ標準への移行を優先する。付属の研究002計画は草案であり、実施済みの実験や外部事前登録ではない。

最も明確な実装上の示唆は、処理数を制限したサービスに対し、成功した要求だけの時間を指標とすると不十分なことである。受け入れ制御が仕事を待機させず拒否する場合、サーバーコードを変えなくてもクライアントの待ち行列で完了率が変わる。次の評価では、完了した有効な仕事量、待ち時間分布、取り消し、上限付き再試行、相手ごとの公平性を測定する。2ワーカーは次のアダプターの保守的な初期候補であり、任意の環境での最適値という発見ではない。

妥当性には複数の制約が残る。全ノードはホスト、スケジューラー、メモリ上限、ファイルシステム環境を共有した。反復数の少なさ、キャッシュ再利用、実行履歴、事後追加実験は因果的・統計的な主張を制限する。合成文章と乱数画像は実運用の負荷を代表しない。公開制御はローカル管理者への信頼を要し、遠隔コピーを失効させない。外部監査、Mac実機、別マシン間の転送、世界規模の発見、独立した全プロトコル実装、AIモデルの評価は行っていない。

6 結論と再現性

0.1では明示的に公開した情報について、検証付き取得、手動再公開、根拠を伴うアプリケーション処理までの流れをローカルに確認した。同時に、並行要求の拒否と、可用性・信頼に関する主張の限界も観測した。次の課題は管理下での複数ホスト試験、別の作成者による互換ノード、待ち行列付き取得、保存容量管理である。AI接続は0.2に位置付け、明示的な入力許可、引用の正しさ、課題ごとの意味の一致を、定義した正解に対して評価すべきである。

保存した証拠はCSV・JSONLの要求記録、チェック結果、イベント、環境情報、入力ハッシュ、計画、Object試験ベクトルである。CSVとJSONLは同じ要求を記録し、二重に数えてはならない。元の318要求と追加288要求は計606件で、準備や失敗を含む。606/606成功という結果ではない。測定した成果物はソースと実行ファイルのハッシュで特定する。再ビルドや別環境での実行には、新たな測定と考察が必要であり、本報告の結論をそのまま流用しない。英語版と日本語版は同じ実験記録を扱い、今回の編集改訂では実装の実験を再実行していない。

改訂3は研究001の測定を置き換えず、解釈と仕様の説明を整理した。測定時のソースは `research/security/baseline-source/` に保存した。別途実施したセキュリティ研究TB-AINET-SEC-001では受け入れ制御、受信ロック、キャンセルを修正しており、その結果を本報告の実行ファイルに帰属させない。プレビューでは文書と測定を配布するが、完全なソース公開と配布ライセンスは未確定である。ローカル計画のハッシュや自己署名タグは独立した時刻証明ではなく、人間による独立レビューも主張しない。

研究記録。Two Brain、AINTERNET Genesis、技術報告TB-AINET-TR-001、改訂3。探索的技術報告であり未査読。実装、実験、分析、原稿作成にCodexを使用した。独立監査や第三者による再現は主張しない。測定した実行ファイルのSHA-256は 872cbb9fa7698e0cefa94cb70f9d7ba6d66457392a14dae179a345477c194a36 である。

参考文献

- [1] R. Fielding, M. Nottingham, and J. Reschke. *HTTP Semantics*. RFC 9110, 2022. <https://www.rfc-editor.org/rfc/rfc9110>.
- [2] C. Bormann and P. Hoffman. *Concise Binary Object Representation (CBOR)*. RFC 8949, 2020, § 4.2.1. <https://www.rfc-editor.org/rfc/rfc8949>.
- [3] S. Josefsson and I. Liusvaara. *Edwards-Curve Digital Signature Algorithm (EdDSA)*. RFC 8032, 2017. <https://www.rfc-editor.org/rfc/rfc8032>.
- [4] IPFS Documentation. *Content Identifiers (CIDs)*. <https://docs.ipfs.tech/concepts/content-addressing/>. 2026年10月31日参照。
- [5] R. Cyganiak, D. Wood, and M. Lanthaler, editors. *RDF 1.1 Concepts and Abstract Syntax*. W3C Recommendation, 2014. <https://www.w3.org/TR/rdf11-concepts/>.
- [6] Two Brain. *AINET Language / ID / Object* ワイヤ仕様1、実験計画および記録、AINTERNET Genesis 0.1、2026。付属成果物：spec/wire.md、research/PLAN.md、research/FOLLOWUP.md、research/results/。
- [7] IPFS Documentation. *InterPlanetary Name System (IPNS)*. <https://docs.ipfs.tech/concepts/ipns/>.
- [8] Named Data Networking Project. *Architectural Overview*. <https://named-data.net/project/archoverview/>.
- [9] Nostr. *NIP-01: Basic protocol flow description*. <https://github.com/nostr-protocol/nips/blob/master/01.md>.
- [10] O. Steele, S. Lasker, and H. Birkholz. *COSE Hash Envelope*. RFC 9995, 2026. <https://www.rfc-editor.org/rfc/rfc9995.html>.
- [11] IPLD. *DAG-CBOR Specification*. <https://ipld.io/specs/codecs/dag-cbor/spec/>.
- [12] C2PA. *Technical Specification*, version 2.2. https://spec.c2pa.org/specifications/specifications/2.2/specs/C2PA_Specification.html. Accessed 31 October 2026.