Revision-Native Media

Persistent Identity, Continuous Mastering, and the Architecture of Audiovisual Publication

Flyxion
Independent Researcher

November 2026
Draft Specification, Version 0.8

Abstract

Conventional video publishing treats a recording as finished at the moment of upload. Corrections require new uploads, references expressed as timestamps decay when the material changes, alternative edits duplicate their sources, separately published streams cannot be played together, and the company that hosts a work controls what is placed before, inside, and around it. This monograph specifies a publishing architecture in which a work has a durable identity, its states are immutable and signed compositions of independently addressed source materials, and every playable file is a derived realization of one such state. Four properties are kept distinct: who committed to material, whether it has changed, whether a rendition realizes its composition, and whether the viewer permitted the particular use being made of it. The specification defines passage lineage and a procedure for resolving old references that reports alteration, ambiguity, and withdrawal explicitly. It defines temporal correspondences that permit separately hosted recordings, soundtracks, transcripts, and camera views to be synchronized, and a proposal mechanism through which other people can contribute repairs, translations, and alternative cuts. Resistance to inserted advertising is treated as a structural requirement: a conforming player emits only sequences of presentation events that the viewer's selections and actions authorize, the document formats contain no means of compelling attention or reporting behaviour, and discovery services may not sell position. The limits of these guarantees are stated and, where possible, proved. Arrangements are described for meeting the costs of storage, transmission, processing, moderation, and preservation without charging for capabilities already present on the viewer's device. No technical rule can determine whether an author was paid to say something, and no slogan can bind an institution. The architecture therefore combines verification, disclosure, exit, independent implementation, and governance constraints that outsiders can check.

Keywords: version control; digital publishing; video; provenance; content addressing; synchronization; collaborative editing; advertising; protocol governance

Contents

1 Purpose and Scope **5**
- 1.1 The Condition Being Addressed . . . 5
- 1.2 Design Goals . . . 5
- 1.3 Matters Outside the Specification . . . 6
- 1.4 Status and Conventions . . . 6

2 Terms and System Model **7**
- 2.1 Participants . . . 7
- 2.2 Objects . . . 7
- 2.3 Time . . . 8
- 2.4 Canonical Records and Signatures . . . 8

3 Compositions **10**
- 3.1 Clips, Occurrences, and Tracks . . . 10
- 3.2 Passages . . . 10
- 3.3 Three Textual Layers . . . 10
- 3.4 Transformations and Derivations . . . 11
- 3.5 Continuous Mastering . . . 11
- 3.6 Multiple Observations of One Event . . . 12
- 3.7 Audio Stems . . . 12

4 Revision, Lineage, and Reference **13**
- 4.1 Revision Records . . . 13
- 4.2 Lineage . . . 13
- 4.3 References . . . 14
- 4.4 Three Kinds of Continuity . . . 14
- 4.5 The Evidence Set . . . 14
- 4.6 Paired Lineage . . . 15
- 4.7 Point Resolution . . . 15
- 4.8 Coordinates and Intervals . . . 17
- 4.9 Annotation Migration . . . 18
- 4.10 Branching Histories . . . 19
- 4.11 Describing Changes . . . 19
- 4.12 Equivocation . . . 19
- 4.13 Head Transitions . . . 20
- 4.14 Latest State and Freshness . . . 20
- 4.15 Withdrawal and Deletion . . . 21

5 Rendering, Storage, and Delivery **22**
5.1 Dependency-Aware Rendering . . . 22
5.2 Segmented Delivery . . . 22
5.3 Verification Status of a Rendition . . . 23
5.4 Storage . . . 23
5.5 Listening and Offline Use . . . 23

6 Resistance to Implanted Advertising **25**
6.1 The Problem . . . 25
6.2 Definitions . . . 25
6.3 Four Distinct Properties . . . 26
6.4 The Presentation Boundary . . . 26
6.5 Viewer-Specific Editions . . . 28
6.6 Capabilities the Formats Lack . . . 28
6.7 Paid Material Placed by Authors . . . 29
6.8 Discovery . . . 30
6.9 Clients . . . 31
6.10 Summary of Defences . . . 31
6.11 Payment Without Attention . . . 31

7 Presentation Semantics **32**
7.1 Session State . . . 32
7.2 Grants, Restrictions, and Precedence . . . 32
7.3 Order of Operations . . . 32
7.4 Updates . . . 33
7.5 Transitions . . . 33
7.6 Revocation . . . 34
7.7 Preservation Results . . . 34

8 Contribution and Derivative Editions **36**
8.1 Proposals . . . 36
8.2 Structural Operations . . . 36
8.3 Merging . . . 36
8.4 Relations Between Editions . . . 37
8.5 Rights, Consent, and Misuse . . . 38

9 Synchronization **39**
9.1 The Aim . . . 39
9.2 Correspondences . . . 39
9.3 Companion Behaviour . . . 40
9.4 Sessions . . . 40
9.5 Clocks and Error . . . 41
9.6 Distributed Presentations . . . 41

10 Discovery, Federation, and Stewardship **42**
10.1 Separable Services . . . 42
10.2 Moderation . . . 42
10.3 Comments and Annotation . . . 42
10.4 Accessibility . . . 43

11 Economic Architecture **44**
11.1 The Obligation to Account for Cost . . . 44
11.2 Kinds of Cost . . . 44
11.3 Arrangements for Payment . . . 44
11.4 Preservation and Departure . . . 45

12 Governance Against Capture **46**
12.1 Why a Motto Is Insufficient . . . 46
12.2 The Invariants . . . 46
12.3 What Licensing Can and Cannot Do . . . 47
12.4 The Conformance Mark . . . 47
12.5 Amendment . . . 48

13 Security and Threat Model **49**
13.1 Assumptions . . . 49
13.2 Threats . . . 49

14 Conformance **51**
14.1 Profiles . . . 51
14.2 Claims and Their Scope . . . 51
14.3 Test Cases . . . 51
14.4 Differential Fixtures . . . 53

15 Implementation Order and Open Decisions **55**
15.1 Order of Construction . . . 55
15.2 Decisions Not Yet Made . . . 55
15.3 Conclusion . . . 55

A Record Sketches **57**
A.1 Revision . . . 57
A.2 Composition Fragment . . . 57
A.3 Lineage . . . 58
A.4 Correspondence . . . 58
A.5 Head, Grant, and Claim . . . 58
A.6 Gain Envelope . . . 59

Chapter 1

Purpose and Scope

1.1 The Condition Being Addressed

A published video is ordinarily a single encoded object attached to a single page. The recording, the edit, the soundtrack, the captions, and the surrounding presentation are fused at the point of release, and the fusion is administered by whoever operates the page. An author who discovers a false statement at the fourteenth minute may add a note beneath the video, publish a second video explaining the first, or remove the whole publication and lose the address through which others cited it. A viewer who wants the same lecture with the music removed, with the camera held on the demonstrator’s hands, or with a transcript on a second screen has no general means of asking for it. A person who marked a position in the recording holds a number of seconds that will point somewhere else as soon as anything earlier in the file is changed. A person who listens while working holds a license, in practice, to hear whatever the host chooses to splice into the stream.

These defects share a structure. Each follows from treating the playable file as the authoritative object, so that identity, history, composition, and presentation have nowhere else to reside. Software development addressed an analogous problem by distinguishing a repository, which records sources and their history, from a build, which is produced from a particular recorded state [8]. Typeset scholarship makes a similar distinction between source files and a compiled document. The present specification applies the distinction to audiovisual publication and then examines what must be added because recordings have properties that program text lacks: they extend in time, they are expensive to transform, their meaning is poorly captured by line differences, and they are consumed continuously by people who are not looking at the editing tools.

Video is the most demanding case and supplies many of the examples, but the architecture concerns any publication that unfolds in time. A podcast or audiobook is a composition with audio tracks and text. A documentary archive needs durable identity and honest history for material that may be annotated for decades. An educational repository depends on repair, translation, and stable citation. A recorded performance may offer several views of one event, and collaborative scholarship may join recordings, manuscripts, and commentary held by different institutions. None of these requires a video platform, and an implementation limited to sound can satisfy the listening profile of Chapter 14 in full.

1.2 Design Goals

The architecture has six goals. A work keeps one identity while its contents change. Every released state remains distinguishable and, subject to lawful and ethical removal, recoverable.

A reference made against one state either resolves correctly against another or says why it cannot. Recordings, soundtracks, text, and views are separable resources that may be recombined and synchronized, including across independent hosts. People other than the original author can contribute work without obtaining control over the publication. Finally, no party standing between author and viewer can add material to a presentation, demand attention as a condition of playback, or turn the act of watching into a supply of behavioural records.

1.3 Matters Outside the Specification

The specification does not describe a hosting company, a social network, a recommendation product, or a means of paying creators at scale. It does not require generative reconstruction of any medium. It does not guarantee that published statements are true, that authors disclose their interests honestly, or that a work will remain available when nobody is willing to store it. It does not attempt to prevent a viewer from recording what a device plays. Where the text identifies a problem that verification cannot solve, the problem is recorded as a limit rather than assigned to a future mechanism.

1.4 Status and Conventions

This is a design specification. No conforming implementation exists, and the numerical bounds and formats given here have not been tested in operation. The words MUST, MUST NOT, SHOULD, and MAY in numbered requirements carry the meanings assigned by the standard requirement keywords [20, 25]. An *invariant* is a requirement that no later version of the protocol may weaken while retaining its name; the invariants are collected in Chapter 12. Statements labelled as propositions are proved under the assumptions written with them, and their force extends no further than those assumptions.

Chapter 2

Terms and System Model

2.1 Participants

Definition 2.1 (Participants)**.** *An* author *is a holder of signing keys authorized to advance the official history of a work. A* contributor *submits proposed changes. A* host *stores and transmits data. An* index *answers queries about works. A* client *is software that retrieves, verifies, and presents works on behalf of a* viewer*, who is the person watching, listening, or reading.*

One person or organization may occupy several of these positions, but the protocol assigns powers to positions. A host that is also an author may change a work only through the author's signed history, and its possession of the stored bytes confers no editorial authority. An index may order or omit results according to its published method, but it cannot alter the works it lists. The viewer is the only participant entitled to decide, for a given session, which permitted material is presented.

2.2 Objects

Definition 2.2 (Asset)**.** *An* asset *is a finite immutable sequence of bytes together with a declared media type. Its identifier is a cryptographic digest of the bytes.*

The initial digest function is SHA-256 [14], and every identifier carries the name of its function so that a later one can be introduced without reinterpreting old records. An asset may hold a recording, an image, a font, a transcript, a subtitle track, or the source of an animation. Because the identifier is derived from the content, any holder can serve an asset and any recipient can check it, which separates the question of whether bytes are correct from the question of who supplied them. Content addressing of this kind underlies distributed version control and peer storage systems [8, 4].

Definition 2.3 (Work)**.** *A* work *is a publication identity I, established by a signed founding record that names the keys authorized to extend its history.*

The identifier I is the digest of the founding record. It therefore survives changes of host, domain name, title, duration, and contents. Human-readable addresses are aliases resolved to I; an alias may be lost or reassigned without altering the identity it once named, and a client MUST display a change in what an alias resolves to.

Definition 2.4 (Composition and revision)**.** *A* composition *C is a document specifying which intervals of which assets are presented, on which tracks, at which times, and with*

which transformations. A revision *r is a signed record binding a composition C_r to a work, to the revision or revisions it succeeds, and to the lineage information described in Chapter 4.*

Definition 2.5 (Rendition)**.** *Given a rendering profile p, which fixes codec, resolution, channel layout, language selection, and related parameters, a* rendition *is a playable asset*

$$V_{r,p} = \mathrm{Render}(C_r, p). \tag{2.1}$$

A work can accordingly be summarized as $W = (I, G, \mathcal{R}, h)$, where G is the directed graph of dependencies among assets and compositions, $\mathcal{R}$ is the set of revisions ordered by succession, and $h \in \mathcal{R}$ is the head that the author currently designates for general viewing. The graph G is the union of the dependency closures of all revisions. Two encodings of one composition are two renditions of one edition. Two compositions that sound alike may nevertheless assert different things and are then different editions, whatever their acoustic similarity.

2.3 Time

Floating-point seconds are unsuitable as the stored measure of position, because repeated conversion accumulates error and because different software rounds differently. A position within an audio asset is an integer sample index at the asset's declared sampling rate. A position within a video asset is an integer count in a rational time base declared by the asset. A *presentation time* on a composition's timeline is a rational number of seconds. All intervals are half open, written $[a, b)$, so that adjacent intervals share no instant.

Sample positions are only meaningful relative to a definite decoding. Lossy formats carry encoder delay and padding that decoders have not always handled alike, so an interval of a compressed recording may denote slightly different audio under different software. A recording used for exact reference SHOULD therefore be ingested by decoding it once into a lossless representation, such as FLAC [29], whose decoded samples are then the asset to which positions refer. The compressed original is retained as a separate asset with a derivation record connecting the two. For audio distributed to listeners, efficient lossy codecs such as Opus [22] remain appropriate, since a listener's rendition need not be the object against which references are defined.

2.4 Canonical Records and Signatures

Requirement 2.6 (Canonical serialization)**.** Every signed record MUST be serialized by a deterministic canonical form before hashing, and the initial form is the JSON Canonicalization Scheme [27]. A record MUST NOT contain duplicate keys, non-finite numbers, or fields not defined by its declared schema version.

Requirement 2.7 (Numeric normal form)**.** Integers in signed records MUST lie in $[-(2^{53} - 1), 2^{53} - 1]$, the range that interchange formats based on double precision represent exactly; a later schema needing more MUST encode them as canonical decimal strings under a new version. A rational number MUST be a pair $[n, q]$ of such integers with $q > 0$ and $\gcd(|n|, q) = 1$, with zero written $[0, 1]$. Records in any other form are rejected, so that $[2, 4]$ cannot denote a coordinate distinct from $[1, 2]$.

Signatures initially use Ed25519 [24]. A founding record names one or more authorized keys and the threshold required to change that set. Key rotation is itself a record in the history. Loss of all keys ends the official history without destroying it; others may continue

from the last revision only as a derivative work. Compromise of a key permits forged revisions until it is revoked, and revisions signed after the declared moment of compromise MUST be flagged by clients that learn of the revocation. Neither signature nor digest establishes who a person is. The binding of a key to a human or institution rests on evidence outside the protocol, which provenance standards such as the specification of the Coalition for Content Provenance and Authenticity treat through certificate-based trust lists [6]. This architecture permits such attestations as optional attachments and does not require an author to disclose a legal identity.

Chapter 3

Compositions

3.1 Clips, Occurrences, and Tracks

Definition 3.1 (Clip occurrence)**.** *A* clip occurrence *is a tuple* $o = (k, a, [m, n), \tau, \rho, \chi, \eta)$*, where* k *is an identifier unique within the work,* a *is an asset,* $[m, n)$ *is a source interval,* τ *is the presentation time at which the interval begins,* $\rho > 0$ *is a rational playback rate,* χ *is a track, and* η *is a finite ordered list of declared transformations.*

With source rate f samples per second, the occurrence covers presentation interval $[\tau,\ \tau + (n - m)/(f\rho))$, and in the absence of transformations that alter time, source sample $j \in [m, n)$ is presented at

$$t_o(j) = \tau + \frac{j - m}{f\rho}. \tag{3.1}$$

The identifier k names this particular use of the material. The same source interval may appear twice in one composition, as when a demonstration is replayed, and the two uses then carry different identifiers. An identifier is retained in a successor revision only while the occurrence refers to the same asset and its source interval overlaps the earlier one; any other change creates a new identifier connected to the old through lineage.

3.2 Passages

Definition 3.2 (Passage)**.** *A* passage *is a named set of occurrence intervals that an author treats as a unit of discourse or demonstration, together with optional text, references, and descriptive attributes.*

A passage might be a sentence with its narration and illustration, a step in a repair, or a movement in a performance. Passages may nest into sections and chapters. They are editorial objects: nothing in a waveform determines where one ends. Automatic transcription and caption cues, for instance in WebVTT [36], supply a provisional segmentation that the author may merge, divide, or move.

3.3 Three Textual Layers

Spoken material requires the system to keep apart three texts that are easily confused. The *observed transcript* records what is judged to be audible in a recording, with the method of judgment declared. The *edited transcript* is a reading version in which hesitations, errors, and spelling may be corrected. The *authorized script* states what the author now wishes

the passage to say, including wording that no recording yet contains. Each is a separate asset aligned to occurrences.

Requirement 3.3 (Textual honesty)**.** A client MUST NOT present an edited transcript or authorized script as a record of audible speech. Where the authorized script differs from the observed transcript of the presented audio, the difference MUST be available to the viewer. A change to text MUST NOT be recorded as a change to a recording.

An author may therefore correct a sentence in text immediately and record the new narration later. In the interval, a reader sees the correction and a listener can be told that the audio has not yet caught up. If a synthetic voice supplies the new narration, the derived asset MUST declare that it is synthetic and name the method class, since a listener may reasonably care whether a person spoke the words.

3.4 Transformations and Derivations

Definition 3.4 (Derivation)**.** *A* derivation *is a signed record* $d = (a, a', \phi, \delta)$ *stating that asset* a' *was produced from asset* a *by a described process* ϕ*, with a* time map δ *relating positions of* a *to positions of* a'*.*

Noise removal, equalization, colour correction, background replacement, and restoration all yield new assets. Without a derivation record, each would sever every exact reference into the material even when no word or action had changed. The time map is a finite list of segments, each pairing a source interval with a target interval under an affine relation, with source or target gaps recorded explicitly. A pure filter has a single segment with unit slope and zero offset, apart from any declared latency. A record of this kind is an assertion by its signer. A verifier MAY test it, for example by cross-correlating short windows of the two assets and rejecting a map whose implied offsets disagree with the observed ones beyond a declared tolerance, but passing such a test shows temporal correspondence only and not equivalence of content.

Transformations listed in η are applied at render time and MUST be drawn from a registry whose entries fix their parameters and behaviour. A transformation that cannot be reproduced from its declaration, including most operations performed by learned models, MUST be materialized as a derived asset, so that the composition refers to its actual output and does not depend on a model that may change or disappear. Systems for reproducible software builds provide the relevant discipline: an output is reproducible only when inputs, tools, and parameters are all fixed [19].

3.5 Continuous Mastering

The separation of sources from renditions allows a publication to improve after release. An author may publish a plain recording on the day it is made, then later repair damaged audio, correct lighting, remove a visible microphone where the picture permits it, add diagrams, replace the visual track while keeping the spoken argument, or record a passage again. These operations are ordinary revisions. None obliges the author to treat the recording room as the permanent setting of the work. Restoration has limits that the record should acknowledge: clipped peaks, occluded objects, and unrecorded sounds can be estimated or replaced, and an estimate is a new creation whose derivation declares it as such.

3.6 Multiple Observations of One Event

A tutorial may be recorded by ten cameras, several microphones, and a screen capture. These are independent observations of one event, each with its own clock. An *event timeline* is a reference timeline to which each observation is aligned by a correspondence as defined in Chapter 9. A *cut* is then a function from event time to a layout, where a layout selects one or more views and audio sources and arranges them. The author's official production is one cut. A viewer with permission to access the sources may hold the view on the demonstrator's hands, show an overhead camera beside a screen capture, or follow another contributor's cut. Because a cut contains only selections and layout, publishing a second one costs a small document and no duplicate footage.

Requirement 3.5 (Source access)**.** Publication of a rendition MUST NOT be taken to grant access to its sources. Each asset carries an access and licensing declaration, and a composition that depends on an inaccessible asset MUST report the dependency rather than substitute other material.

3.7 Audio Stems

Definition 3.6 (Stem set)**.** *A* stem set *is a collection of time-aligned audio assets $x_1, \dots, x_n$, labelled by function, with a declared mixing rule that produces the author's mix.*

Typical labels are dialogue, narration, music, ambience, mechanical sound, performed effects, and sudden loud events. Where the mixing rule is linear, a personal mix is

$$y(t) = \sum_{i=1}^{n} g_i(t)\, x_i(t), \tag{3.2}$$

with viewer-controlled gain functions g_i, and the author's mix is recovered by the author's gains. Where the finished soundtrack was produced by a nonlinear mastering chain M, such as compression or limiting applied to the sum, the composition MUST declare whether M is to be applied after the viewer's gains. Applying it preserves loudness behaviour but changes how the remaining stems respond when one is removed; omitting it yields a mix that differs from the released one even at the author's gains. Either choice is legitimate when stated.

A viewer who works all day to recorded speech, shares a room, or is distressed by sudden noise needs to lower explosions or gunfire without abandoning the work. Where stems exist this is a gain setting. Where they do not, two weaker instruments remain. A *gain envelope* is a small document listing intervals and attenuations relative to an identified rendition; it carries no audio and can be exchanged independently. A reconstructed mix, made by source separation or manual editing, is a derivative asset that MUST disclose its origin, since separation can leave artifacts and can remove wanted sound. Clients SHOULD also offer a local dynamic range limiter that requires no document from anyone. Loudness measurement and normalization SHOULD follow an established broadcast recommendation [12], so that two editions can be compared on a common scale.

Chapter 4

Revision, Lineage, and Reference

4.1 Revision Records

A revision is immutable. Correcting a revision means publishing a successor. For revision r with canonical body D_r, the revision identifier is

$$H_r = \mathrm{Hash}(D_r), \tag{4.1}$$

where D_r contains the work identifier, the identifiers of all predecessor revisions, the digest of C_r, the lineage record, the author's description, a claimed time of publication, and the signing key. Since predecessor identifiers are inside the hashed body, each identifier commits to the whole ancestry. Branches are expressed by two revisions naming one predecessor, and merges by one revision naming several.

Definition 4.1 (Head record)**.** *A* head record *is a signed statement* (I, H_r, s) *designating* H_r *as the current revision of work* I*, with a strictly increasing sequence number* s*.*

The address of a work resolves to the revision named by its latest head record. An address that includes H_r resolves to that revision or to an explicit statement of its unavailability. A client MUST NOT accept a head record whose sequence number is lower than one it has already accepted for the same work, which prevents a host from silently reverting a viewer to an older state.

4.2 Lineage

Persistent passage names do not by themselves establish continuity. A name can be kept on material that has been wholly replaced, and material can survive while its name is discarded. Continuity must be stated by a relation and, for exact positions, grounded in the source material.

Definition 4.2 (Lineage record)**.** *Let* P_r *and* P_s *be the passage sets of a predecessor* r *and successor* s*. A* lineage record *is a tuple* (E, X, N, F) *with typed edges* $E \subseteq P_r \times P_s \times K$*, a deleted set* $X \subseteq P_r$*, an inserted set* $N \subseteq P_s$*, and* occurrence edges F *from occurrences of* r *to occurrences of* s*, where* K *is the set of edge types. An occurrence edge is valid only between passages joined by an edge in* E *and between source intervals that overlap in the same asset or in assets joined by derivation.*

Requirement 4.3 (Total accounting)**.** Every $u \in P_r$ MUST either be the source of at least one edge or belong to X, and not both. Every $v \in P_s$ MUST either be the target of at least one edge or belong to N, and not both.

The requirement turns omission into an error that software can detect. Cardinality alone does not describe what happened, since preservation, replacement, and reordering all connect one passage to one passage. The edge types therefore distinguish them.

Edge type	Shape	Condition checked by a verifier
`preserved`	one to one	Identical assets, source intervals, and transformations
`moved`	one to one	As preserved, with a different relative order
`derived`	one to one	Assets connected by derivation records covering the intervals
`replaced`	one to one	No material condition; the edge is an editorial assertion
`split`	one to many	Two or more edges from one source
`joined`	many to one	Two or more edges into one target
`recomposed`	many to many	No material condition

Table 4.1: Lineage edge types. A type with a material condition may be rejected when the condition fails.

Lineage is distinct from equivalence. An author may replace an argument by its denial and correctly record that the new passage succeeds the old one. Textual identity, acoustic identity, compositional identity, and claimed sameness of meaning are separate relations, and only the first three can be checked mechanically.

4.3 References

Definition 4.4 (Reference)**.** *A* reference *is a tuple* $b = (I, H_r, u, k, a, j, e)$ *naming a work, a revision, a passage, a clip occurrence, a source asset, a source position* j*, and an optional end position* e*.*

Comments, citations, bookmarks, annotations, and playback positions are all stored as references. The occurrence identifier records which use of repeated material was meant. The revision identifier records the context in which the reference was made, which remains part of its meaning even after it has been migrated.

4.4 Three Kinds of Continuity

Finding the same stored sample in a later revision does not show that a reference continues to mean what it meant. An interviewee's words can be cut from an answer that rejected a proposal and reused in a montage that appears to support it. The bytes survive and the object of the citation does not. Source identity, editorial descent, and surrounding context are therefore established and reported separately, as attributes of candidates, and none of them is a stage whose success ends the search for others.

4.5 The Evidence Set

Two resolvers can agree only if they reason from the same records.

Definition 4.5 (Evidence set)**.** *For a reference made at revision* r *and a target* s*, the* evidence set $\Xi_{r,s}$ *consists of the revisions, compositions, and lineage records on every path from* r *to* s*, together with every derivation record named by those revision records and signed by a key authorized for the work. Where no path joins them,* $\Xi_{r,s}$ *consists of* r *and* s*, their compositions, and the derivation records those two revisions name; it contains no lineage, and so supports no witness.*

Derivations published by others may be added by the viewer; a result then identifies the enlarged set. If any member of $\Xi_{r,s}$ cannot be obtained, the result is marked *incomplete* and names what is missing. A work-independent target is Unrelated unless a declared derivative relation connects it to the source.

4.6 Paired Lineage

Passage descent and occurrence descent established separately, perhaps on different branches, do not add up to one history. Both are therefore followed together.

Definition 4.6 (Paired step and witness)**.** *Let r' be an immediate successor of r_0 with lineage record (E, X, N, F) relative to r_0. A* paired step $(u, k, j) \to (v, \ell, j^*)$ *exists if k belongs to passage u and ℓ to passage v; $(u, v, \kappa) \in E$ for some type κ; either $\ell = k$ with the retention condition of Chapter 3 or $(k, \ell) \in F$; and position j^* in the source interval of ℓ equals j where the asset is unchanged, or is the image of j under the composed time maps of a derivation path in $\Xi_{r,s}$ that leads from the old asset to the new one and repeats no asset. A* witness *is a chain of paired steps along a single revision path, recording at each step the revision, passage, occurrence, position, and derivation path used.*

A step may therefore carry several derivation records, as when a recording is cleaned and then equalized between two published revisions. Requiring an intermediate revision for each record would make continuity depend on how often an author happened to publish.

4.7 Point Resolution

Definition 4.7 (Candidate)**.** *A* candidate *is a tuple $c = (o, j', \gamma, \omega, \varepsilon, A)$: a target occurrence o, a target position j', a derivation path γ (empty for the original asset), a witness ω (possibly absent), an uncertainty ε, and an attribute set A.*

The resolver MUST always enumerate both kinds of candidate. The first kind takes every occurrence in C_s that draws from asset a and contains j. The second takes every derivation path in $\Xi_{r,s}$ from a that repeats no asset, and every occurrence of its final asset containing the image of j. Every witness ending at a candidate's occurrence and position is attached to it. Candidates may be consolidated only when occurrence and exact rational position coincide, and the consolidated entry retains all paths and witnesses. Two paths that reach different positions within one occurrence yield two candidates.

The attributes are independent: *source identity*, when γ is empty; *derived correspondence*, when it is not; *occurrence continuity*, when a witness is present whose derivation paths, taken in order, concatenate to γ; and *contextual continuity*, defined below, which presupposes occurrence continuity. The result also lists every descendant passage of u.

Definition 4.8 (Structural summary)**.** *The summary of a complete result is the first that applies. With exactly one candidate having occurrence continuity:* Continued *if it has source identity and contextual continuity,* Processed *if it has derived correspondence, and otherwise* Recontextualized*. With several having occurrence continuity:* Ambiguous*. With none, but candidates present:* Rediscovered *if there is one and* Ambiguous *if several. With no candidates:* Altered *if u has a descendant in s, and* Withdrawn *if it has none.*

A copy of original bytes in an appendix therefore cannot hide a restored passage: both are candidates, and only the restored one carries a witness. Where no path leads from r to s, no witness exists and source attributes alone are available. Resource limits on path enumeration are permitted, but a truncated enumeration MUST be reported as incomplete and MUST NOT receive a summary asserting uniqueness.

Derivation Evidence

A derivation record is *declared*; it is additionally *supported* or *contradicted* with respect to a named test, parameters, and verifier. The candidate set is computed from declarations alone, so that optional testing cannot change what exists. Eligibility for automatic navigation or annotation migration is a second stage, in which the viewer's policy states which tests are required and how contradiction is treated. The default excludes any candidate resting on a contradicted record and displays it with that reason.

Context

Neighbouring passages are not well defined when passages nest, overlap, or run on several tracks. Context is defined from the timeline.

Definition 4.9 (Contextual projection)**.** *Let t be the presentation time of the referenced position in r, and let $[h_0, h_1)$ be the least interval containing the referenced passage. For a margin $\mu \geq 0$, put $\alpha = t - h_0 + \mu$ and $\beta = h_1 - t + \mu$. The* window *is the half-open interval $[t - \alpha, t + \beta)$. A* portion *is the intersection of the window with the half-open presentation interval of an occurrence, on any track, when that intersection has positive length. It records asset, source interval, start and end relative to t, rate, track, role, transformations, layout, gain, applicable mastering chain, text layers, and declarations. Two portions are* joinable *if the first ends where the second begins in both presentation time and source position and every other recorded attribute is equal. Portions are taken in canonical order, and each is joined to the first earlier chain to which it is joinable. The* projection $\Gamma_r(t, \alpha, \beta)$ *is the canonically sorted list of the resulting chains, together with the durations by which the window extends before the start of the programme and after the latest end of any occurrence. Identifiers and absolute programme time are omitted.*

A candidate has contextual continuity if it has occurrence continuity and source identity, its occurrence has the same rate as the referenced occurrence, and $\Gamma_s(t', \alpha, \beta) = \Gamma_r(t, \alpha, \beta)$, where t' is its presentation time. The margin defaults to ten seconds, a provisional value, and a reference MAY record another or a named wider section, whose bounds then replace $[h_0, h_1)$. Because every presentation attribute outside the stated omissions is compared, a changed playback rate or a move to a background track is detected. Dividing an occurrence into contiguous parts that are otherwise identical leaves the projection unchanged, in agreement with the comparison of traces in Chapter 7; a division that also separates text layers or declarations is detected. This remains a statement about the composition and does not establish that meaning is unchanged.

Proposition 4.10 (Determinacy)**.** *Two resolvers that implement these definitions without truncation, given the same reference, target, margin, and evidence set, produce the same candidates, attributes, and summary.*

Proof. The occurrences of C_s are finite. Derivation paths repeating no asset in a finite record set are finite, as are revision paths in a finite acyclic graph and chains of paired steps along them. Positions are computed in exact rational arithmetic. Each attribute is a predicate on these objects and the summary is a function of the attributed set. No step depends on search order or on an optional test. □

Proposition 4.11 (Source coordinate preservation)**.** *Suppose digests are collision resistant and the client has verified the assets and composition of s. If a candidate has source identity, then its occurrence draws from the asset designated by the reference and its source interval contains j.*

Proof. Source identity requires an empty derivation path, so the occurrence names the identifier a and $j' = j$ lies in its source interval. Under collision resistance, verified bytes bearing that identifier are the bytes designated when the reference was made. □

Proposition 4.12 (Sample correspondence under identity rendering)**.** *Suppose in addition that the occurrence has an empty transformation list, rate one, no overlapping occurrence on any audio track, no gain or mastering chain, output at the asset's sampling rate without resampling, and a decoder that seeks sample accurately in a lossless asset. Then the sample the client submits for output at presentation time $t_o(j)$ equals stored sample j of a.*

Proof. Under the stated conditions the render function over the neighbourhood of $t_o(j)$ is the identity on the stored samples with the index translation given by t_o. Evaluating it at $t_o(j)$ yields sample j. □

Resampling, time stretching, cross fading, mixing, filtering, and approximate seeking each break the hypothesis, and the audio device lies beyond it.

4.8 Coordinates and Intervals

Resolution keeps positions as exact rationals, using arithmetic of whatever precision composition requires and serializing large results as decimal strings. A map such as $\delta(j) = \frac{3}{2}j + \frac{1}{4}$ therefore needs no rounding. Containment in half-open intervals is tested on the rational value. Only a rendering or seeking operation rounds, by taking the greatest integer sample not exceeding the coordinate, and the rounding is reported.

A reference with end position e designates an ordered list $J = (J_1, \ldots, J_n)$ of source intervals, each belonging to a named passage and occurrence. The resolver partitions it at every position corresponding to a boundary of an occurrence or of a time-map segment, whether that boundary lies in the target, in the revision of the reference, or in any other revision of $\Xi_{r,s}$, pulled back along every derivation path. Intermediate revisions cannot be ignored. Material trimmed in one revision and restored in the next has the same candidates on both sides of the cut and different witnesses.

Resolution is defined here for time maps that increase on every segment; a record with any other segment is reported as unsupported. For each part of the partition the resolver enumerates candidates at its first position, which is also the position at which context is evaluated. Two correspondences are consolidated only if they reach the same occurrence by the same map over the whole part, so that maps which merely cross at one position remain distinct.

Definition 4.13 (Coherent selection)**.** *A* coherent selection *chooses one candidate for every part such that either no member has a witness, or every member has one and all lie on a common revision path; one derivation path can be chosen for each member so that parts from the same original asset use identical records wherever the domains of those records meet; original order is preserved; and parts adjacent in the original presentation remain adjacent on the same track.*

A selection that joined a part linked by lineage to a part merely found again would present two kinds of evidence as one, and is therefore not coherent. Call a coherent selection *witnessed* if its members have witnesses.

Definition 4.14 (Interval summary)**.** *The summary of a complete interval result is the first that applies. With exactly one witnessed coherent selection:* Continued *if every member*

has source identity and contextual continuity, Processed *if some member has derived correspondence, and otherwise* Recontextualized. *With several witnessed:* Ambiguous. *With none witnessed, but coherent selections present:* Rediscovered *if there is one and* Ambiguous *if several. With no coherent selection but at least one part having a candidate:* Fragmented. *With no candidates:* Altered *if a referenced passage has a descendant in s, and* Withdrawn *otherwise.*

Proposition 4.15 (Agreement of points and intervals)**.** *Let an interval reference consist of a single part that the partition does not divide, and let j be a position in it at which no two distinct candidate maps coincide. Then the interval summary equals the point summary at j.*

Proof. Between boundaries, containment in every occurrence and time-map segment is constant, so the occurrences reached, their derivation paths, and the existence of witnesses are constant. Context requires equal rates and source identity; the presentation times t and t' then advance together, and every quantity compared by the two projections is either absolute in source position or relative to those times, over windows whose bounds are fixed relative to the composition. A coherent selection of one part is a single candidate, and the clauses of the interval summary correspond one for one to those of the point summary. □

Local uniqueness of every part is not sufficient for continuity. The report states the numbers of witnessed and unwitnessed coherent selections and gives two coverage figures over the original reference: *sample coverage*, the fraction of original samples with a candidate, and *duration coverage*, the corresponding fraction of original source time. A reference made of one second at 48 kHz that survives and one second at 8 kHz that does not has sample coverage 6/7 and duration coverage 1/2. It also reports multiplicity, order changes, and discontinuities. A point result MUST NOT be presented as an interval result.

4.9 Annotation Migration

Structural facts do not settle which annotations should follow them. A warning about a loud sound survives a change of argument; a statement that a passage is promotional may not, as when the same recording is later examined as an example of misleading advertising.

Annotation basis	Default condition for automatic application
Acoustic property	Source identity, or derived correspondence with a supporting test of the relevant property
Textual correction	Source identity and unchanged relevant text layer
Rhetorical or commercial function	Occurrence and contextual continuity
Citation	Contextual continuity
Personal bookmark	Policy chosen by the viewer

Table 4.2: Migration bases. An annotation with no declared basis requires contextual continuity.

Every annotation declares its basis, and the viewer MAY substitute a stricter or looser condition. Interval annotations are applied automatically only to a selection satisfying the condition on every part that is the sole witnessed coherent selection, or, where the condition requires no witness, the sole coherent selection. Declarations carried by the target revision itself apply independently of any migration.

4.10 Branching Histories

A merge revision MUST carry a separate lineage record for each predecessor, each satisfying total accounting. The descendant set $\Delta_{r,s}(u)$ contains the passages $v \in P_s$ for which some path from r to s admits a chain of passage edges from u to v. If it is nonempty while on some other path every chain ends before s, the result is marked Contested and identifies the revisions where chains ended. If it is empty, Withdrawn reports all such revisions. Descendant sets describe editorial history; only witnesses establish continuity of a position.

Requirement 4.16 (Visible migration)**.** A client MUST NOT present any result as continuing a reference unless its summary is Continued. It MUST state the summary and attributes, and where the original revision remains accessible MUST offer it.

Example 4.17. A recording has passages A, B, C, and D, each longer than the margin. Its successor adds an introduction before A, keeps A, records B again, divides C into C_1 and C_2 by cutting a pause, and removes D. A mark in A is Recontextualized, since the projection now includes the introduction and the new B. A mark in B is Altered. A mark in the retained part of C with a declared occurrence edge is Recontextualized, and without one Rediscovered. A mark in the removed pause is Altered. A mark in D is Withdrawn, and becomes Rediscovered if its audio is reused in a montage without lineage.

4.11 Describing Changes

An author may describe a revision as technical, presentational, linguistic, substantive, evidentiary, or a retraction. These labels help readers and are nothing more than assertions. Removing the word *not* is a very small textual edit that reverses a proposition, and it can be labelled a correction of wording. A client MUST make the underlying differences in text, assets, and composition available independently of any label. Automated analysis MAY mark changes that appear consequential, including added or removed negations, quantities, and names, and MUST NOT certify that meaning is unchanged on the strength of a similarity score. Third parties MAY publish their own signed descriptions of a change, and a viewer chooses whose descriptions to display.

An author is entitled to change an opinion and to keep an error out of the current edition, so that later viewers need not sit through it. The entitlement does not extend to representing the current wording as what was said before. Revision identifiers make the distinction practical, since a citation to H_r continues to denote r.

4.12 Equivocation

A signed ancestry shows that a key endorsed each record. It cannot show that the author published only one history: two viewers may be given two different head records with the same sequence number, each internally valid. Detecting this requires comparison. Transparency logs, developed for web certificates, maintain an append-only Merkle tree for which a log can prove that an entry is included and that a later state extends an earlier one [28, 9].

Requirement 4.18 (Logged heads)**.** A head record MUST be entered in public append-only logs, to the threshold stated below, before a conforming client treats it as official. Clients MUST verify an inclusion proof, MUST retain log checkpoints and verify that later checkpoints extend them, and SHOULD exchange checkpoints with independent observers. Two valid

head records for one work with equal sequence numbers and different revisions constitute proof of equivocation, which a client MUST report.

This arrangement detects equivocation among parties who compare what they have seen. It does not detect it among parties who never communicate, and a log operator colluding with an author can delay discovery until a comparison occurs.

4.13 Head Transitions

A higher sequence number shows that a head record is newer and says nothing about the direction of the change. A client classifies each accepted head against the last one it accepted by the first rule that applies: *retraction*, if the record designates no replacement and accompanies a withdrawal of the current revision, or designates a replacement while withdrawing it; *renewal*, if the revision is unchanged; *exact composition reinstatement*, if the composition digest differs from the current one and equals that of any ancestor or earlier head; *advance*, if the new revision is a descendant of the current one; and *switch* otherwise. Moving to an ancestor with a composition identical to the current one is classified as a switch and shown as moving backwards. Retractions, reinstatements, and switches MUST be displayed. The reinstatement test detects identical compositions only. A restoration with renamed identifiers, or a partial reversion, appears as an advance, and passage differences remain the means of finding it.

4.14 Latest State and Freshness

An inclusion proof shows that a head was logged. A recent checkpoint that also contains a later head does nothing for a client shown only the earlier proof. What is required is evidence that no greater sequence number exists in the relevant log at that checkpoint.

Requirement 4.19 (Latest head evidence)**.** A conforming log MUST publish, with each checkpoint, the root of an authenticated map from work identifiers to the greatest head record entered for each, and MUST answer queries with a proof of the mapped value against that root. Independent monitors MUST be able to recompute the map from the log. Alternatively a client MAY enumerate and check every entry through the checkpoint. A founding record, as amended by later signed records, names n logs and a write threshold w; a head is official only when entered in at least w of them. To report a head as current, a client MUST obtain latest-head evidence from at least $n - w + 1$ of the named logs.

Proposition 4.20 (Latest head under quorum)**.** *Suppose every official head has been entered in at least w of n named logs before the queried checkpoints, every queried log's map is correct, and the client takes the greatest valid sequence number returned by at least $n - w + 1$ of them. Then no official head so entered has a greater sequence number.*

Proof. A set of w logs and a set of $n - w + 1$ logs drawn from n have sizes summing to $n + 1$ and therefore share a member. The shared log contains any given official head and its correct map reports that head's sequence number or a greater one. □

Each head also states a validity period, renewed by a further record. Status is *current as of* the oldest checkpoint used, if within the period and the evidence above is complete; otherwise *expired* or *unknown*. A first contact without such evidence reports unknown freshness. The guarantee depends on correct maps and says nothing of heads entered after the checkpoints. Named logs are a dependency that mirrors and successor records can

relieve but not remove: when too few respond, playback of verified material continues and currency is unknown.

4.15 Withdrawal and Deletion

Definition 4.21 (Withdrawal record)**.** *A* withdrawal record *is a signed statement that identified revisions, passages, or assets are no longer offered, with a reason class that may be left unspecified.*

Revision integrity cannot mean that every recording is retained or displayed forever. Privacy, safety, consent, and law can require removal, sometimes of the descriptive record as well as the media. The ordinary case removes the media and keeps the signed fact of withdrawal. A plain digest of small or guessable material, such as a short transcript containing a name, can confirm a guess after the material is gone. Sensitive assets SHOULD therefore be identified under a separate scheme, written with its own prefix, as Hash("salted" $\|$ σ $\|$ a) for a random value σ of at least 256 bits stored beside the asset. A recipient retrieves both σ and a and verifies the pair. Equal contents under different salts have different identifiers, so such assets are not deduplicated across works. Destroying the salt and asset protects against later confirmation only for those who never held them; anyone who retrieved both retains the ability to verify, and the protocol cannot revoke it. A host subject to a lawful order may have to delete records that this specification would prefer to keep. Its obligation is to report unavailability and never to supply a replacement under the old identifier.

Chapter 5

Rendering, Storage, and Delivery

5.1 Dependency-Aware Rendering

Let G be the rendering dependency graph, whose edges run from inputs to products, and let S be the set of inputs changed between two revisions. The products that require regeneration lie in

$$\mathcal{A}(S) = S \cup \mathrm{Desc}_G(S). \tag{5.1}$$

If a localized edit affects duration ΔT of a work of duration T, the ratio $\Delta T/T$ describes the directly affected share of the timeline and nothing more. Normalizing loudness over a whole programme, grading colour globally, cross fades, motion-compensated compression, and captions burned into pictures can each extend $\mathcal{A}(S)$ far beyond the edit.

Requirement 5.1 (Cache soundness)**.** A cached product MAY be reused only if the digest of its full build description, comprising input identifiers, operation identifiers and versions, parameters, and rendering profile, equals that of the product required.

Rendering procedures that vary between machines, as threaded and hardware encoders may, produce different bytes from identical descriptions. The cache rule remains sound because it never asserts equality of outputs; it reuses one output recorded for one description. Where independent verification of a rendition is wanted, the profile MUST name a deterministic toolchain, or the comparison MUST be made on decoded media within a declared tolerance and reported as such. FFmpeg is a suitable initial engine [13], with the exact version and configuration included in the build description.

5.2 Segmented Delivery

Adaptive streaming divides a rendition into short segments listed in a playlist or manifest, as in HTTP Live Streaming [26] and Dynamic Adaptive Streaming over HTTP [10]. Segments unchanged between revisions can be shared. For video this requires compatible encoder settings, segment boundaries at random access points, consistent timing information, and, where these cannot be matched, declared discontinuities. Audio and video MUST be offered as separate streams, both because a change to one then leaves the other intact and because a listener MUST be able to obtain sound without transferring or displaying pictures.

Definition 5.2 (Segment commitment)**.** *For a rendition divided into segments $g_1, \dots, g_m$, the* segment commitment *is the root of a Merkle tree [15] over the digests of the segments in order, together with their count, durations, and the rendering profile.*

A player holding a trusted root can verify each segment as it arrives from any host, using a short proof of its position. Leaf and interior hashes MUST be domain separated, as in certificate transparency [28], and each leaf MUST bind its index, duration, track, and digest. The commitment also covers initialization segments, track identifiers, codec configuration, edit lists, discontinuity declarations, and every other piece of container metadata capable of changing what is played or when.

5.3 Verification Status of a Rendition

Definition 5.3 (Rendition status)**.** *A rendition is* authenticated *if an authorized signer committed to its bytes and metadata;* composition bound *if that signer further states that it is a rendition of an identified composition under an identified profile; and* render verified *if a named verifier has established the correspondence by reproducing it bit for bit with a deterministic toolchain, or by comparing decoded media under a declared procedure and tolerance.*

The first two are cryptographic facts about what was signed. Neither prevents an author from signing as a rendition a file containing a promotional segment absent from the composition. Only the third examines the relation, and a tolerant comparison is limited to what its procedure can detect. A client MUST display the status and MUST NOT call a rendition verified on the strength of a signature. Where sources are private or discarded, a rendition may remain playable while verification is impossible, and the record says so. A client MAY offer a mode that renders locally from sources or accepts only render-verified editions.

A transcode produced and attested only by a host is a third-party edition. A client MUST NOT play it in place of an official rendition unless the viewer explicitly grants it, and MUST then identify the edition as that party's. An author may make a transcode official by signing its commitment and composition binding, whether after rendering locally or after checking a deterministic render.

5.4 Storage

If $\mathcal{U}$ is the set of distinct stored chunks, $\ell(c)$ the size of chunk c, $q(c)$ its number of replicas, and π a price per unit of storage and time, the recurring storage cost is approximately

$$\Pi_{\text{store}} = \pi \sum_{c \in \mathcal{U}} \ell(c)\, q(c), \tag{5.2}$$

to which transmission and computation must be added. Sharing unchanged material reduces $|\mathcal{U}|$, while retaining raw multi-camera sources can enlarge it by an order of magnitude or more over a single rendition. Authors MAY declare retention classes that distinguish material required to reconstruct published revisions from working sources that may be discarded. A revision whose sources have been discarded but whose rendition survives is *playable but not reconstructible*, and records MUST say which condition holds.

5.5 Listening and Offline Use

Sound is an independently useful publication. A client MUST be able to play audio with the screen off or without any visual component, store permitted works for offline use, and keep references and playback history locally. Playback SHOULD offer fine control of tempo

with pitch controlled separately, configurable pauses between sentences where alignment exists, repetition of marked intervals, and export of annotations in an open form such as the Web Annotation Data Model [3]. Where a publication's license permits downloading, the download MUST be an ordinary file readable by common software, accompanied by its records. A person's listening ledger belongs to that person and may combine activity from many sources; no host receives it by virtue of having supplied some of the material.

Chapter 6

Resistance to Implanted Advertising

6.1 The Problem

Brin and Page observed in the appendix to their 1998 description of the Google prototype that financing a search engine by advertising creates incentives at odds with serving its users, and they argued for a transparent engine outside the commercial sphere [5]. The company that grew from the prototype later placed the injunction against being evil in its code of conduct, and in 2018 removed it from the preface of that code while leaving a mention at its close [37]. The episode need not be read as a verdict on any individual to yield a design lesson. A declaration of intent binds only as long as those who control its interpretation choose to be bound. A structural constraint should make the unwanted act impossible, detectable, or costly for the party that would otherwise perform it, and should leave everyone else able to leave.

Advertising reaches viewers of video through several distinct routes. It may be requested by the player and shown between portions of the work. It may be joined to the media stream on the server, a method that YouTube was reported in 2024 to be testing and that displaced the timestamps relied upon by a community tool for skipping sponsored segments [16]. It may occupy the page around the picture, determine which works are recommended, or be spoken by the author as part of the recording. Each route calls for a different defence, and at least one admits no technical defence at all.

6.2 Definitions

Definition 6.1 (Consideration and paid material)**.** Consideration *is money, goods, services, preferential treatment, or any other benefit. Material is* paid material *if its inclusion, wording, placement, or prominence was conditioned on consideration from a party other than the viewer who receives it.*

Definition 6.2 (Implanted material)**.** *Material presented during a session is* implanted *if it is neither committed to by the signed records of a publication the viewer selected, nor selected by the viewer, nor part of the client's own controls.*

The definitions separate two properties that popular usage joins. Implanted material need not be paid: a host's notice spliced into a recording is implanted. Paid material need not be implanted: an author's sponsored sentence is inside the signed composition. Implantation

concerns who put the material there and can be tested against records. Payment concerns why it is there and generally cannot.

6.3 Four Distinct Properties

Property	Question	Established by
Authenticity	Who committed to this material?	Signature under an authorized key
Integrity	Has the committed material changed?	Digest or segment commitment
Fidelity	Does the rendition realize its composition?	Reproduction or declared comparison
Authorization	Did the viewer permit this use?	Membership in the session's authorized events

Table 6.1: Properties none of which implies all the others.

An authentic and unaltered segment from an author's commercial announcement is still an insertion if a host plays it in the middle of that author's documentary. So is an authentic segment of the documentary played twice, out of order, or on the wrong track. Verification must attach to the use of material.

6.4 The Presentation Boundary

Definition 6.3 (Presentation event)**.** *A* presentation event *is a tuple* $\epsilon = (g, a, J, \chi, \tau, \eta, \lambda)$*, where* g *identifies the granting selection,* a *an asset,* J *a source interval,* χ *a presentation role and track,* τ *a placement on the plan's timeline,* η *a list of transformations, and* λ *a spatial or acoustic layout.*

Definition 6.4 (Grant and presentation plan)**.** *A* grant *is a viewer's selection of one identified resource, at a fixed revision or under stated update classes, conferring stated capabilities. The* presentation plan $\mathcal{P}$ *of a session is computed by the client from its grants, the signed records they name, its active restrictions, and the viewer's local settings. Chapter 7 defines which sequences of events the plan permits.*

Invariant 1 (Closed presentation)**.** The trace of presentation events emitted by a conforming client in a session shall be a prefix of a trace permitted by the session's authenticated records and the viewer's grants, exclusions, and actions, as defined in Chapter 7. The only other occupation of the presentation surface or audio output shall be controls the viewer invokes, accessibility functions the viewer enables, notices necessary to report errors, verification status, or changes in what is being presented, and dismissible information about the current session. No client shall use this exception for paid material, recommendations the viewer did not request, or promotion of its own or any other party's services.

Requirement 6.5 (Verified delivery and schedule)**.** Before performing an event, the client MUST verify the asset or segment against a commitment in a signed record named by the grant, and MUST verify that its position, order, count, duration, track, profile, and container metadata agree with that record. The client MUST construct its schedule from authenticated records. A host playlist MAY indicate where bytes can be retrieved and MUST NOT determine what is presented, how often, or when. Verification failure MUST halt the affected event and be reported. The client MUST NOT accept executable code, replacement manifests, or substitute segments from a host.

Delegated Authority

Publication authority defines an official work. *Dependency authority* supplies a resource that a selected composition names. *Companion authority* supplies an independently identified supplement. *Presentation authority*, which determines what the viewer experiences, belongs to the viewer and is delegated only through grants.

Requirement 6.6 (Confined grants)**.** A grant MUST name resource, capabilities, revision rule, and update classes. A *capability* is a triple of presentation role, operation, and output region or channel. Nothing referenced by a granted resource, including a recommended companion, index entry, or later synchronization manifest, receives a grant by reference unless it is a constitutive dependency as defined below. A subtitle companion may emit text in its region and no audio; amplification is a capability distinct from attenuation; a correspondence may align granted resources and introduce none.

Definition 6.7 (Constitutive dependency)**.** *A dependency edge (x, y) is* constitutive *if the signed record of x names y by digest, marks the edge as required to realize an operation of x, and y cannot be selected for presentation except through x under this grant.*

Requirement 6.8 (Attenuation)**.** A constitutive dependency receives capabilities satisfying $\mathrm{Cap}(y \mid x) \subseteq \mathrm{Cap}(x)$, restricted to those the parent's operation needs. Any other dependency requires its own grant. By transitivity of inclusion, no chain of dependencies exceeds the capabilities of its granted root.

A font named by an animation is constitutive and may shape the animation's text. A commentary track is independent. Sound inside an animation is presented only if the animation itself was granted an audio capability.

Interpretation

A closed record schema leaves open what a decoder may do with an asset. A vector image, font, subtitle file, animation, or container may carry scripts or external references that no manifest mentions.

Requirement 6.9 (Closed interpretation)**.** Each supported format and transformation MUST have a registered profile stating which of its features are interpreted. Decoders and renderers MUST run without ambient network access, filesystem authority, or access to keys, history, or playback state beyond the inputs and outputs passed to them. Every external dependency MUST be a declared, content-addressed asset that is either separately granted or constitutive. Unsupported active features and external references MUST be refused.

The theorem that an untrusted intermediary cannot enlarge what is presented depends on how a session changes over time and is stated in Chapter 7.

Unavailability and Fallback

Requirement 6.10 (No coerced substitution)**.** A viewer's policy states the set of verification evidence accepted, since bit-for-bit reproduction, tolerant comparison under a particular procedure, and a signer's composition binding are different kinds of evidence and not points on one scale. A client MAY substitute one rendition for another without asking only if both are bound by the same signer to the same composition, carry identical paid declarations, differ solely in declared delivery parameters, and the substitute's evidence is in the accepted set. The actual status MUST remain visible, including any change. A substitute outside the accepted set, or one that introduces material or changes events, language, track selection, or sponsorship, requires explicit authorization. Unavailability never widens a grant.

A publisher may still make an undesirable variant the only one that can be retrieved. The viewer is then offered a visible choice between that variant and none, and a composition binding that has not been render verified remains the signer’s assertion.

6.5 Viewer-Specific Editions

An author, or a platform that has obtained an author’s keys, could sign a different composition for each viewer and thereby satisfy every verification. The logged head requirement confines official status to records visible in a public log, which makes the multiplication of heads observable.

Requirement 6.11 (Uniform publication)**.** A head record MUST NOT vary with the identity, location history, or behaviour of the requester. A host or index MUST NOT require an identifying credential as a parameter of manifest retrieval for public works. Variants offered for language, accessibility, or bandwidth MUST all be enumerated in the one signed revision, and the client MUST choose among them locally.

Uniform publication prevents covert tailoring from being conforming and makes it detectable when viewers compare logs. It does not stop an author who is willing to break the rule from doing so until comparison reveals it. For works restricted to subscribers, access control MAY determine whether a person receives the revision but MUST NOT determine what the revision contains.

6.6 Capabilities the Formats Lack

A rule against abuse is weaker than the absence of any means to commit it. Many obligations that advertising systems impose on viewers are expressed through player instructions, and the document formats here simply provide none.

Invariant 2 (No compulsion)**.** No record format shall express a prohibition on seeking, skipping, pausing, muting, changing rate, or closing; a requirement that particular material be watched before other material; an automatic action that opens an address or application; or a condition on playback that depends on observing the viewer.

Invariant 3 (No reporting)**.** No record format shall express a callback, measurement address, tracking identifier, or event to be transmitted. A conforming client shall transmit nothing about viewing behaviour unless the viewer performs a specific act of disclosure.

Requirement 6.12 (Closed schemas)**.** A client MUST reject any record containing an undefined field. Extensions MUST be registered with a published definition, and no registered extension may grant a capability excluded by an invariant. An unregistered or unsupported extension MUST NOT be partly interpreted.

Closed schemas answer a familiar route of capture, in which a dominant implementation tolerates extra fields, gives them meanings, and then presents other software as defective for ignoring them. Hyperlinks to references remain permitted, are inert until the viewer activates them, and MUST show their destination first. A client MUST NOT rewrite them to pass through an intermediary.

Requirement 6.13 (Declared disclosure)**.** Each network operation a client performs MUST have a documented purpose and a statement of the data it necessarily discloses. Request selection, order, repetition, and timing SHOULD follow published policies that do not depend

on the viewer beyond what retrieval requires, and optional values MUST NOT vary with behaviour.

Removing tracking fields does not remove covert channels. A dishonest client can encode information in the order or timing of legitimate requests or in repeated log queries, and a monitor that checks destinations will not notice. No general demonstration of noninterference is claimed. These measures do not make observation impossible. Any host that serves a request learns a network address, a time, and which bytes were sought, and a sequence of segment requests approximates a viewing record. Downloading whole works in advance, obtaining them from several hosts or peers, caching through a trusted relay, and prohibitions on retention in host conformance terms reduce the exposure without removing it.

6.7 Paid Material Placed by Authors

The next route runs through the author. A sponsored passage is signed like any other, and no property of its bytes records the arrangement.

Proposition 6.14 (Non-identifiability of consideration from published records)**.** *No procedure whose inputs are only the published records and media can correctly determine for every publication whether it contains paid material.*

Proof. Consider two situations in which an author records the same sentence praising a product, producing identical media and identical records. In the first the author was paid to include it and in the second was not. The procedure receives identical inputs and must return identical answers, whereas the correct answers differ. It is therefore wrong in at least one situation. □

Detection tools, including classifiers, may produce useful suspicions, and they cannot overturn the proposition, since the relevant fact lies in an agreement outside the data. The architecture consequently relies on declaration, independent report, and viewer choice.

Requirement 6.15 (Declared consideration)**.** Every passage, occurrence, asset, overlay, track, text layer, transformation, or metadata item containing paid material MUST carry a declaration naming the providing party and the form of consideration, attached at the narrowest compositional object to which it applies, and every revision MUST carry a work-level statement that it contains declared paid passages, declares none, or makes no declaration. Self-promotion by which the author solicits purchases or payments MUST be marked with its own distinct attribute.

Invariant 4 (Viewer refusal)**.** Every conforming client shall allow the viewer to omit declared paid and self-promotional objects automatically at the narrowest declared boundary, to exclude works that contain them or make no declaration, and to apply independent annotations in the same way. No record, host, or index shall penalize the viewer for doing so.

A sponsored overlay, subtitle line, or music track can thus be removed while the tutorial beneath it continues. Material already composited into one picture or mix cannot be separated, and its declaration applies to the whole inseparable object. Where omission would break the sense of the surrounding material, the interruption is a fact about the work, and the viewer may prefer another edition. Independent annotators MAY publish signed segment reports against revision identifiers, giving others a basis for skipping undeclared promotion. Crowdsourced skipping of sponsored segments already demonstrates the practice

[31]. Because the reports are anchored to source positions rather than programme time, the resolution procedure of Chapter 4 carries them across later edits and reports where it cannot. Reports are claims by their signers; a client MUST let the viewer select whose reports to apply and MUST allow disputes to be recorded beside them.

An author found to have declared falsely cannot be compelled by the protocol to do anything. The false statement is, however, signed, dated, and attached to an identifiable revision, which makes it available as evidence to whatever legal or social process applies.

A stricter policy is available, in which the conforming network refuses all paid passages. It is simple to state and, by Proposition 6.14, cannot be enforced against concealment; its likely effect would be to move declared sponsorship into undeclared speech. This draft therefore selects declaration with an unconditional right of refusal, and records the stricter policy among the open decisions. Either policy leaves Invariant 1 untouched: nobody but the author can place anything in the author's work.

6.8 Discovery

Paid position in search or recommendation influences what is seen without changing any work. An index is not trusted merely because it describes itself as neutral.

Definition 6.16 (Reproducible ranking)**.** *A query response is* reproducible *if it identifies a published ranking program, a snapshot of its public inputs by digest, and the viewer-supplied parameters, such that running the program on those inputs and parameters yields the response.*

Invariant 5 (Unsold position)**.** No conforming index or client shall let consideration affect inclusion, exclusion, rank, or rendered prominence, including size, preview, animation, repetition, and placement outside the result list, or mix paid entries with results. A conforming index shall make its responses reproducible.

Proposition 6.17 (Detectable departure)**.** *If a signed query response claims reproducibility and its named inputs are available, any difference between the response and the output of the named program on those inputs is demonstrable to a third party.*

Proof. The response is signed and names the program, input snapshot, and parameters. A third party verifies the digests, executes the deterministic program, and compares. A differing result, presented with the signed response and the inputs, can be checked by anyone repeating the computation. □

Reproducibility reveals departure from the declared method. It does not reveal payment that influenced the choice of method, the contents of the input snapshot, or an omission made before the snapshot was formed. Comparison between independent indices, public records of exclusions, and the viewer's freedom to replace an index entirely are therefore indispensable. A reproducible ordering can be displayed with unequal prominence, so a client MUST publish its layout rules and compute emphasis only from those rules, the response, and the viewer's settings. As with ranking, publication of a rule does not reveal payment behind its design. Subscriptions in chronological order, direct retrieval by identifier, and local filtering MUST work with no index at all. Ranking that depends on a person's history SHOULD be computed on that person's device.

6.9 Clients

Every guarantee above is enforced at the client, so control over the client is the decisive point of capture. The protocol MUST be fully specified in public, the verification and resolution procedures MUST have a shared test suite, and at least two independently written clients SHOULD exist before any version is declared stable. A web page operated by a host that plays media through code the host can change at each visit is not a conforming client for the purposes of these invariants, however faithful it is on a particular day. A browser client can conform when its code is distributed as a versioned release whose digest the viewer's software can check and pin. Operating systems, display devices, and hardware vendors can place material over any application. The protocol cannot prevent this, and conformance claims MUST NOT imply that it can.

6.10 Summary of Defences

Route	Mechanism	Residual exposure
Stream insertion by host or network	Verified schedule and trace admissibility	Refusal of service; nonconforming clients
Tailored signed editions	Uniform publication and public logs	Delay before viewers compare records
Forced or unskippable playback	Formats lack the instruction	Modified clients
Behavioural measurement	Formats lack callbacks; local state	Host request records
Sponsored speech	Declaration, reports, viewer refusal	Concealed arrangements
Sold rank	Unsold position and reproducibility	Manipulated inputs or method
Hardware or system overlays	Outside the protocol	Entire

Table 6.2: Routes by which advertising reaches a viewer and the limit of each defence.

6.11 Payment Without Attention

A system that excludes implanted advertising must say how its costs are met, or the omission will be filled later by whoever offers to pay. Authors may host their own works; viewers may pay authors or hosts directly; libraries, schools, cooperatives, and public bodies may store collections they value; and participants may contribute bandwidth and storage as peers, as PeerTube instances can with peer-assisted delivery [18]. A record MAY name addresses for voluntary or required payment. Payment MUST NOT purchase rank, MUST NOT be solicited by an interruption that the viewer cannot dismiss, and MUST NOT require disclosure of viewing history. None of these arrangements guarantees an income to creators or the survival of unpopular material, and the specification does not claim otherwise. Chapter 11 treats the costs and charging rules in detail.

Chapter 7

Presentation Semantics

7.1 Session State

Membership of each event in an authorized set cannot exclude playing an authorized event three times, or in the wrong order. Authorization is a property of sequences produced by permitted transitions.

Definition 7.1 (Session state)**.** *A* session state *is* $\Sigma = (R, G, D, Q, U, n)$*: accepted records* R*, active grants* G*, active restrictions* D*, queued events* Q*, the history* U *of authorized actions, and an* authorization epoch n*.*

7.2 Grants, Restrictions, and Precedence

A *grant* confers capabilities over identified material. A *restriction* removes or weakens them, and includes standing exclusions of paid material, gain envelopes that attenuate, and applied segment reports. With $\mathcal{A}(g)$ the events a grant permits and $\mathcal{D}(d)$ the events a restriction forbids, the effective authorization is

$$\mathcal{A}_{\text{eff}} = \Big(\bigcup_{g \in G} \mathcal{A}(g)\Big) \setminus \Big(\bigcup_{d \in D} \mathcal{D}(d)\Big). \tag{7.1}$$

Restrictions prevail. Selecting a resource, even one whose declared contents include excluded material, does not lift an exclusion. The client reports the conflict, and the viewer may override that particular restriction for a stated scope, which is an action recorded in U. Classifying attenuating resources as restrictions matters: were a gain envelope a grant, withdrawing it would silently permit the louder original under the lecture's own grant. As a restriction, its removal is visibly an act that widens authorization.

7.3 Order of Operations

Requirement 7.2 (Canonical pipeline)**.** Operations MUST apply in this order: selection of the revision and variant; exclusion of restricted objects; source-coordinate operations, including gain envelopes and filters anchored to asset positions; composition transformations in their listed order; rate and temporal placement; correspondence placement of companions; mixing and layout; declared mastering; session rate and viewer device adjustments. Within a stage, operations that do not commute MUST be ordered by an explicit dependency in a signed record or a viewer setting, and are otherwise rejected. A restriction anchored to another edition's timeline MUST be carried to source positions through an identified correspondence before use, and fails visibly if it cannot be.

An envelope that quietens an explosion thus follows the explosion's samples wherever synchronization places them. A grant to perform an operation does not by itself permit combining it with another whose joint effect needs a further capability.

7.4 Updates

A subscription must admit a corrected word without admitting anything whatever.

Definition 7.3 (Update predicate)**.** *Let* $\mathrm{Diff}(r, s)$ *be the canonical difference between two revisions, listing every added, removed, and changed occurrence, asset, role, capability, dependency, companion, interpreter feature, language, and declaration.* $\mathrm{PermitUpdate}(g, r, s)$ *holds if* s *is an advance from* r *under the grant's authority and every entry of the difference falls in an update class that* g *names.*

Every difference belongs to exactly one class, and a difference that no class describes fails the predicate.

Class	Differences	Default
Media	Replacement, removal, reordering, and insertion within granted roles	Permitted
Treatment	Rate, gain, transformations, layout, and mastering chain within granted capabilities	Permitted
Text	Text layers in an already granted language	Permitted
Constitutive	Constitutive dependencies under attenuation	Permitted
Capability	A new role, or a pairing of role with track or output region not previously used	Asks
Companion	A new companion or external resource	Asks
Interpreter	A new interpreter feature or language	Asks
Declaration	Any declaration added, altered, or removed, including every paid declaration	Asks

Table 7.1: Update classes. Reinstatements and switches always require the viewer.

Treatment is permitted because a replacement asset could already produce any effect that a change of treatment produces, and refusing it would stop ordinary remastering while protecting nothing. The viewer's restrictions apply to the result in either case. Removal of a paid declaration asks even when the declared material is removed with it, since the two cannot be told apart from material that stays under a new identifier. Failure of the predicate leaves the accepted revision in place and asks. A viewer MAY choose to approve every change. The default is a real delegation: a permitted replacement recording can contain anything the author says, including undeclared promotion. Confinement limits what kind of thing an update may do and does not judge its content.

7.5 Transitions

The authenticated compositions, effective authorization, and action history determine a *scheduler*, which at each state names the events that may be emitted next. The set of sequences it can produce is the permitted language $\mathcal{L}$. Implementations may divide output into blocks differently, so traces are compared after merging adjacent blocks that are contiguous in both source and presentation coordinates. Interruption and refusal of service shorten a trace and remain within the prefix closure $\mathrm{Pref}(\mathcal{L})$. A viewer pressing repeat extends the language; a host sending a segment twice does not.

Transition	May change	Condition
Delivery	R	Verified under a named authority; head ordering respected
Viewer action	G, D, U, Q, n	Performed by the viewer
Update	Revision realized by a grant, Q, n	PermitUpdate
Remote command	U, Q, transport state	Controller permitted and locally authorized
Scheduling	Q	Next events of the plan, tagged with n
Emission	Trace	Head of Q, epoch current, in $\mathcal{A}_{\text{eff}}$

Table 7.2: Admissible transitions. No other transition exists.

7.6 Revocation

Retrieval, decoding, and presentation are distinct permissions, and having retrieved or decoded material confers no right to present it later.

Requirement 7.4 (Bounded revocation)**.** Any action that narrows effective authorization MUST increment the epoch and remove affected queued events before they cross the *output boundary*, the last point at which the client can withhold them. A client MUST declare the maximum duration λ of material that may lie beyond that boundary on each output path, including estimated device buffers, and MUST report paths for which no bound is known. For another device in the session, the bound runs from receipt of the new epoch; session permissions on remote devices MUST be leases that lapse without renewal.

7.7 Preservation Results

Proposition 7.5 (Authority non-expansion)**.** *A transition whose inputs are not signed by an authority named in a grant and not performed by the viewer cannot enlarge any grant's capabilities, remove a restriction, or change the revision a grant realizes.*

Proof. By the transition table, G and D change only through viewer actions. A realized revision changes only through an update, which requires PermitUpdate and therefore a revision under the granted authority. A remote command is intersected with local authorization. Delivery changes R only with verified records, which by themselves confer no capability. □

Proposition 7.6 (Revocation preservation)**.** *Removing a grant, with restrictions unchanged, does not enlarge $\mathcal{A}_{eff}$.*

Proof. The union over a subset of grants is contained in the union over all of them, and removing a fixed set from both preserves the inclusion. □

Proposition 7.7 (Trace admissibility)**.** *Assume collision and second-preimage resistant digests, unforgeable signatures, uncompromised keys named by grants, and a client that implements the transitions, verification, and interpreter isolation above. Then, apart from material already beyond the output boundary within the declared bound when authorization narrows, the emitted trace belongs to* $\mathrm{Pref}(\mathcal{L})$ *for the session's history. An untrusted host or network intermediary cannot cause an event outside it except with negligible probability.*

Proof. Proceed by induction on transitions. The empty trace is a prefix of every permitted trace. Only emission extends the trace, and it emits the scheduled head of Q under the

current epoch and within $\mathcal{A}_{\text{eff}}$, which is by definition a permitted continuation. Other transitions preserve earlier output as permitted under the history in force when it was emitted. An intermediary supplies inputs only to delivery. To alter the schedule it must have an uncommitted record or object accepted, which requires forging a signature or finding a second input matching a committed digest or tree root. □

These results follow largely from the transitions a conforming client is permitted to take. Their use lies in stating exactly which actions can widen authority, so that an implementation or a test can look for any other. They do not address fidelity, truthfulness, concealed sponsorship, or a viewer’s understanding, and a host can still refuse service.

Chapter 8

Contribution and Derivative Editions

8.1 Proposals

A viewer who notices that thirty seconds of a lecture are nearly inaudible should be able to submit the repair and not only a complaint. The mechanism resembles the pull request of collaborative software development, generalized to media.

Definition 8.1 (Proposal)**.** *A* proposal *is a signed record* $q = (I, H_b, \Omega, A, L, \Lambda, m)$ *naming a work and base revision, a list* Ω *of edit operations, the new assets* A*, a lineage record* L *relative to the base, a rights declaration* Λ*, and a description* m*.*

A proposal transmits only what is new. It may repair a transcript, restore damaged audio, supply a translation, replace an illustration, suggest a different camera selection, correct a mathematical explanation, or reconstruct the whole edit. The author may preview the resulting composition, compare it passage by passage with the base, listen to both versions at matched loudness, and accept, decline, or request changes. Acceptance creates a revision signed by the author that names the proposal, so that credit remains attached to the work actually done. Nothing in a contributor’s submission grants authority over the official history.

8.2 Structural Operations

Line-oriented differencing of a serialized composition produces conflicts and agreements that have no relation to the media. Operations are therefore defined on identified objects and use relative placement in preference to absolute time.

8.3 Merging

Definition 8.2 (Footprint and dependency)**.** *The* read set *of an operation is the set of identified objects whose state it presupposes, including anchors; its* write set *is the set of objects it creates, removes, or changes, including the ordering position it occupies. A composition may declare* support edges *from one object to another on whose content it depends, as from a diagram to the spoken explanation it illustrates.*

Definition 8.3 (Structural independence)**.** *Operations* α *and* β *defined on one base are* structurally independent *if neither write set meets the other’s read or write set, and no support edge joins an object written by one to an object written by the other.*

Operation	Effect
`insert`	Place new occurrences after a named anchor on a track
`remove`	Remove named occurrences or declared subintervals
`substitute`	Replace the asset or interval of an occurrence
`reorder`	Move a named passage to a new anchor
`retime`	Change rate or relative placement of named occurrences
`transform`	Alter the transformation list of named occurrences
`annotate`	Add or alter text, references, or attributes
`align`	Add or alter a temporal correspondence

Table 8.1: Initial edit operations.

Proposition 8.4 (Commutation)**.** *Suppose each operation is deterministic, its effect depends only on its read set, and it modifies only its write set. If α and β are structurally independent, then applying them to the base in either order yields the same composition.*

Proof. Applying α changes only objects in its write set, none of which β reads. The inputs of β are therefore the same before and after α, so β has the same effect in either position, and symmetrically for α. Since the write sets are disjoint, neither effect overwrites the other. Both orders produce the base modified by the two effects. □

Two insertions after one anchor both write the position following it and are therefore reported as a conflict, since no order between them is implied. A removal of introductory material and annotations attached by identifier to later passages are independent even though every later programme time moves. A replacement of an explanation conflicts with a new diagram supporting the original explanation, through the support edge, although the two touch different files. Independence guarantees a well-defined composition and nothing about its sense. Support edges record only the dependencies that someone declared, and a merge MUST be presented for human review before it is signed. After merging, the client MUST recheck total lineage accounting, asset availability, correspondences, license compatibility, and the absence of overlap on tracks that forbid it.

8.4 Relations Between Editions

Relation	Meaning	Endorsement
Official revision	Authorized continuation of the work	By definition
Remaster	Technical or presentational reworking	Only if signed
Alternative cut	Different selection or arrangement	Only if signed
Translation	Realization in another language	Only if signed
Accessibility edition	Adaptation to perceptual or cognitive needs	Only if signed
Commentary	Synchronized interpretation	Only if signed
Supplement	Separate material intended to accompany	Only if signed
Fork	Independently maintained derivative work	Only if signed

Table 8.2: Machine-readable relations between editions.

A derivative edition is a work of its own, with its own identity and keys, whose founding record names the source revision. An author may countersign it to express endorsement, and the absence of a countersignature MUST be visible. Credits attach to contributions: a restorer is credited with a restoration and is not thereby made responsible for the original

argument, while the original speaker is not presented as approving a cut that rearranges the speech. A viewer MAY subscribe to a work and choose a preferred translation or quiet mix separately. When the source advances, a derivative records which source revision it follows, and the client shows any lag.

8.5 Rights, Consent, and Misuse

The technical ability to make a derivative confers no legal permission. Each asset and edition carries a license declaration, for which the Creative Commons instruments offer established terms [7]. A gain envelope or timing correspondence contains no copied media, which may matter legally in ways that vary by jurisdiction and that this specification does not decide. Recordings of people raise questions of consent that licenses do not answer. A person recorded on one of ten cameras may have agreed to the official cut and not to the release of every raw view; source release therefore requires its own declaration, and privacy removal MUST be able to reach individual regions, intervals, and assets. Edits can also place words in a speaker’s mouth. Because each derivative identifies its source revision and operations, a viewer can compare them, and a client SHOULD make that comparison readily available for any edition that alters the order or content of speech.

Chapter 9

Synchronization

9.1 The Aim

A viewer should be able to press play once and have a film on one screen, a transcript on another, a quieter soundtrack through headphones, and a commentary from an unrelated server all follow together. Existing tools can play several online videos side by side, and some retain a manually chosen offset between two of them [34]. A constant offset fails as soon as one version omits, adds, or retimes material. The relationship between two recordings must be a document of its own.

9.2 Correspondences

Definition 9.1 (Segment pairing)**.** *A* segment pairing *between timelines T_A and T_B is a triple $([a_0, a_1), [b_0, b_1), \kappa)$ with kind κ, representing the affine map*

$$\psi(t) = b_0 + (t - a_0)\,\frac{b_1 - b_0}{a_1 - a_0}, \qquad t \in [a_0, a_1). \tag{9.1}$$

Definition 9.2 (Temporal correspondence)**.** *A* temporal correspondence *between two identified editions is a finite set of segment pairings together with explicit lists of intervals on each side that have no counterpart.*

Requirement 9.3 (Coverage)**.** On each side, the paired intervals and unmatched intervals of a correspondence MUST together cover the declared timeline. Overlapping pairings denote ambiguity and MUST be marked as such.

A correspondence is a relation between timelines. A single function from one reference timeline to each resource cannot express material that exists only in the resource, such as three minutes of added commentary, whereas an unmatched interval on one side expresses it directly. A clock-rate difference appears as a slope other than one. Repeated material appears as marked overlap. Removed material appears as an unmatched interval and never as an association with whatever happens to follow.

Correspondences compose: if ψ_{AB} and ψ_{BC} are partial maps, their composite is defined where the image of the first meets the domain of the second. Error accumulates along such chains, so a correspondence MUST state a tolerance. If the first map is uncertain by at most ε_1 in the intermediate timeline, and the second is L-Lipschitz on the whole uncertainty region and itself uncertain by at most ε_2, then

$$\varepsilon_{\text{composite}} \le L\,\varepsilon_1 + \varepsilon_2, \tag{9.2}$$

Reference edition	Alternative audio	Kind
00:00 to 02:00	none	unmatched in reference
02:00 to 15:00	00:00 to 13:00	same material
15:00 to 15:30	13:00 to 13:30	altered level
15:30 to 42:00	13:30 to 40:00	same material
none	40:00 to 43:00	unmatched in alternative

Table 9.1: A correspondence with material absent from either side.

by the triangle inequality. A composite MUST report at least this bound. Where the region crosses a discontinuity, an unmatched interval, or an ambiguous pairing, the bound does not apply and the composite MUST report the branches or the gap separately. A correspondence can be created by hand, from shared captions, or by audio fingerprint alignment, and its method MUST be declared.

9.3 Companion Behaviour

During an unmatched interval of the leading resource, a companion has nothing to present. During an unmatched interval of a companion, the session must either skip it, suspend the leader while it plays, or mix it over the leader. The correspondence MUST state a default policy and the viewer MAY override it. A paused leader, a mixed commentary, and a skipped scene are visibly different choices, and a client MUST NOT make one silently.

9.4 Sessions

Definition 9.4 (Session)**.** *A* session *is $S = (\mathcal{P}, t_0, p_0, v, \zeta, M)$, where $\mathcal{P}$ is the presentation plan, p_0 the position on the leading timeline at session clock time t_0, v the rate, ζ the transport state, and M the participating devices with their roles and permissions.*

While playing, the intended position is

$$p(t) = p_0 + v\,(t - t_0). \tag{9.3}$$

Every change of position, rate, or state issues a new tuple with a sequence number and an effective time slightly in the future, so that devices can apply it together. The idea that independent media components on several devices may follow a shared timing resource was developed in the Timing Object report of a World Wide Web Consortium community group [33]. That report is not a consortium standard and states that work on it has been discontinued; it serves here as a design precedent. Requirements for timed events associated with web media have also been studied [32], and the multimedia integration language SMIL addressed declarative temporal composition much earlier [30].

Synchronization Is Subordinate to Authorization

A controller's command takes effect on participant i only if

$$\mathrm{Permit}_i(c) = \mathrm{ControllerAuthorized}(c) \wedge \mathrm{LocallyAuthorized}_i(c). \tag{9.4}$$

A seek into material that a participant excludes, or a change to an edition that participant has not granted, does not override the local decision. That device skips, waits, or leaves synchronization according to its own setting and shows that it has diverged. It SHOULD report divergence without disclosing which restriction caused it. Divergence is preferable to treating a shared clock as power over another person's refusal.

9.5 Clocks and Error

A device estimates the offset θ of the session clock from its own by exchanging four times: local send t'_0, remote receipt t'_1, remote send t'_2, and local receipt t'_3. In the manner of the Network Time Protocol [21],

$$\hat{\theta} = \frac{(t'_1 - t'_0) + (t'_2 - t'_3)}{2}, \qquad \delta = (t'_3 - t'_0) - (t'_2 - t'_1). \tag{9.5}$$

Proposition 9.5 (Offset bound)**.** *If both one-way delays are nonnegative and the offset is constant during the exchange, then* $|\hat{\theta} - \theta| \le \delta/2$.

Proof. Let $d_1, d_2 \ge 0$ be the outward and return delays, so $t'_1 = t'_0 + d_1 + \theta$ and $t'_3 = t'_2 - \theta + d_2$. Substitution gives $\hat{\theta} = \theta + (d_1 - d_2)/2$ and $\delta = d_1 + d_2$. Since $|d_1 - d_2| \le d_1 + d_2$, the bound follows. □

Network offset is only one source of error. Audio output buffers, display latency, decoder delay, and wireless headphones add device-specific delays that a protocol cannot observe from message times, so clients SHOULD offer measured or manual calibration. A device corrects small errors by slight rate adjustment within a bound that preserves pitch, and larger errors by seeking. Required tolerance depends on use: a transcript following speech can lag noticeably without harm, lip synchronization demands far tighter alignment, and several channels of one musical recording split between devices may be unusable at errors that either of the others would accept. A composition MUST declare its tolerance, and a client unable to meet it MUST say so.

9.6 Distributed Presentations

A university may host a lecture, an animator the diagrams, a translator a parallel narration, and a critic a commentary. A *compound presentation* is a small signed document naming each resource by work and revision, the correspondences among them, default roles, and tolerances. It is itself a work that can be revised, cited, and shared. Where reproducibility matters it pins revisions; where it follows current heads it MUST say so and re-evaluate correspondences against each new head, using the resolution outcomes of Chapter 4. A missing or incompatible resource is reported by name. Retrieval across origins is subject to the access rules of each host, and a compound presentation grants no access that its viewer did not already have. Roles in a session are permissions: a controller may be allowed to pause for everyone, while another participant may only follow, and joining a shared session MUST NOT expose a participant's history or local annotations.

Chapter 10

Discovery, Federation, and Stewardship

10.1 Separable Services

Publication, storage, indexing, recommendation, playback, and annotation communicate through documented records, and each may be replaced without migrating the others. A developer can build a specialized listening player against the same records as any other client. Hosts MAY announce new head records and receive responses using a federation protocol such as ActivityPub [1], which supplies delivery among independent servers; the authority of a record still derives from its signature and log entry and never from the server that relayed it. Web archives expose past states of resources through datetime negotiation [23], and an archive of this system SHOULD additionally preserve revision identifiers, which name states exactly.

10.2 Moderation

No single authority decides what every participant must host or see. A host chooses what it stores, subject to the law that applies to it. An index chooses what it lists and MUST publish its exclusion rules and, where lawful, a record of exclusions. Independent labelling services MAY issue signed labels against works, revisions, or passages, and viewers choose which to apply. Distributed responsibility does not dispose of the hardest cases. Material that is abusive or dangerous must be removable by the people who hold it, shared digest lists can prevent its readmission, and content addressing must never be presented as a reason that removal is impossible. Passage-level references allow a complaint, correction, or warning to attach to the precise material concerned.

10.3 Comments and Annotation

Discussion attached to references remains connected to the argument it addresses. When the passage changes, the comment is resolved like any other reference and displayed with its outcome, so that a reply to a withdrawn claim cannot appear to answer its replacement. Comments are separate signed objects held by their authors or by chosen hosts. A work’s author controls what is displayed in the author’s own presentation and not what independent readers may publish elsewhere.

10.4 Accessibility

Captions, transcripts, audio description, sign language video, simplified wording, reduced flashing, reduced motion, and quiet mixes are all companions or editions under the structures already defined. Player controls MUST meet established accessibility guidelines [35]. An accessibility edition may be made by anyone permitted to make a derivative, without waiting for the original publisher, and may be offered for adoption through a proposal.

Chapter 11

Economic Architecture

11.1 The Obligation to Account for Cost

Excluding advertising removes one source of money and none of the expenses. If the expenses are left without a described means of payment, the pressure that brought advertising to other platforms returns, and it returns when hosts are already depended upon. The protocol cannot guarantee that anyone will pay. It can make costs visible, keep several arrangements for meeting them interchangeable, and ensure that no arrangement requires surrendering an invariant.

11.2 Kinds of Cost

Storage recurs for as long as material is kept and was estimated in Chapter 5 as Π_{store}. Transmission grows with use. If y_c is the number of retrievals of chunk c in a period, $\ell(c)$ its size, $\varphi_c \in [0,1]$ the share supplied by peers or local caches, and π_b the price per byte paid by the host, then the host's transmission cost is

$$\Pi_{\text{send}} = \pi_b \sum_{c \in \mathcal{U}} y_c \, \ell(c) \, (1 - \varphi_c). \tag{11.1}$$

Peer supply moves this cost to participants and does not abolish it, and it may expose their network addresses to one another. Rendering and transcoding cost computation in proportion to the invalidated products. Moderation, review of proposals, answering lawful requests, software maintenance, log operation, and independent observation of logs consume human labour, which falls much less than storage prices do as hardware improves and which is likely to dominate for a small community.

11.3 Arrangements for Payment

Definition 11.1 (Resource charge)**.** *A* resource charge *is a payment calculated from a published tariff for a stated quantity of storage, computation, transmission, or human service actually supplied.*

Requirement 11.2 (Transparent charges)**.** A host levying resource charges MUST publish its tariff, state the quantity measured for each charge, give notice before a change takes effect, and retain no billing record beyond what the quantity, the amount, and applicable law require. Metering MUST NOT be used to build a viewing history. A host SHOULD offer prepaid or aggregate allowances that do not associate particular works with an identified person.

Arrangement	Suited to	Characteristic weakness
Self-hosting	Authors with modest audiences	Fragile availability
Direct patronage	Works with committed audiences	Concentration on already visible authors
Cooperative hosting	Groups sharing capacity and rules	Governance labour; unequal use
Institutional repository	Teaching, research, and archives	Selection by institutional priorities
Voluntary contribution	Shared infrastructure and mirrors	Unpredictable income
Resource charges	Storage, computation, and transmission supplied	Excludes those unable to pay

Table 11.1: Arrangements that may be combined for one work.

The requirement meets a real tension. Charging for transmission requires measuring it, while Invariant 3 forbids turning playback into behavioural reporting. A host necessarily sees the requests it serves; the discipline lies in recording a total where a total is sufficient. Charges may be paid by an author, a viewer, an institution, or a cooperative on behalf of its members, and the choice MUST NOT alter rank or conformance.

11.4 Preservation and Departure

Preservation is a promise about years in which present participants may be absent. Suppose keeping a collection costs c_0 in the first year, the cost falls each year by a fraction d with $0 \le d < 1$, and reserved funds earn a constant annual return i with $i + d > 0$. Paying at the start of each year indefinitely requires a present reserve of

$$F = \sum_{t=0}^{\infty} c_0 \left(\frac{1-d}{1+i} \right)^{t} = c_0 \, \frac{1+i}{i+d}. \tag{11.2}$$

The expression is a planning identity and no forecast. Storage prices may stop falling, returns may be negative, and media must also be migrated and checked by people. A host MUST therefore state the period for which retention is funded and MUST NOT describe storage as permanent without a stated means. Before ending service for reasons within its control, it MUST give published notice, keep export available throughout the notice period, and identify any successor. Content addressing allows others to adopt a collection without changing identifiers, provided someone obtains the bytes and the right to hold them.

The intended order of survival is explicit. A work should outlive its host, a host’s failure should not disturb the protocol, the protocol should outlive its reference implementation, and the whole should outlive the founding steward. Export under Invariant 6 covers manifests, permitted source materials, revision history, contribution records, correspondences, and the annotations that their holders choose to carry.

Chapter 12

Governance Against Capture

12.1 Why a Motto Is Insufficient

A promise to behave well names no forbidden act, assigns no observer, provides no remedy, and can be reinterpreted by its maker. The aim behind such a promise can be approached by asking, of each harm to be avoided, who would be able to commit it, who would notice, and what those harmed could then do. The preceding chapters answer these questions for particular technical acts. Governance must answer them for changes to the rules themselves.

12.2 The Invariants

The five invariants of Chapter 6 are closed presentation, absence of compulsion, absence of reporting, viewer refusal, and unsold position. Five further invariants complete the set.

Invariant 6 (Portable identity)**.** A work's identity, records, and permitted assets shall be exportable in complete and documented form, and shall remain verifiable on any host without the consent of a previous host.

Invariant 7 (Honest history)**.** No conforming implementation shall present one revision as another, substitute material under an existing identifier, or report an inexact reference resolution as exact.

Invariant 8 (Open implementation)**.** The complete specification and conformance suite shall be freely available, and implementing them shall require no permission, fee, account, or secret.

Invariant 9 (Independence of central services)**.** Verification, playback of locally held works, and retrieval by identifier from a known host shall not depend on any single organization's service.

Logs, indices, naming services, and signing conveniences tend to centralize even in open systems. A client MUST accept configuration of its logs and indices, although added observers and mirrors do not lower the evidence from a work's named logs required to report a head as current, and a conforming distribution MUST NOT make the default irreplaceable.

Invariant 10 (No withheld local capability)**.** No conforming implementation or service shall disable, degrade, or reserve for payment a capability that the distributed program can perform on the viewer's device using material the viewer is already permitted to hold. This includes playback with the screen off, adjustment of tempo and pitch, rendering of subtitles, and synchronization of locally held files. No implementation shall introduce a

dependency on payment, authentication, or a network service for an operation that could be performed locally with resources already present, where the effect is to make that operation conditional on any of them.

The invariant leaves a developer free to sell a program, or to be paid for building a capability, and leaves a host free to charge for what it stores, computes, or transmits. It forbids shipping a function and then renting back permission to use it. Cloud transcription, remote rendering, and coordination among distant devices may be sold as supplied computation, provided that a local operation is not routed through them to manufacture a charge. The test concerns the effect and does not require proof of motive.

12.3 What Licensing Can and Cannot Do

It is tempting to license the software on condition that it never carry advertising. Such a condition restricts a field of use, which the Open Source Definition does not permit [17]; a program so licensed may be a reasonable product but is not open source, and the restriction would fragment the body of code that others could reuse. A network copyleft license such as the GNU Affero General Public License requires an operator who modifies the program and offers it to users over a network to make the modified source available to them [2]. It thereby exposes changes to examination and allows the unmodified or repaired version to be run by others. It does not forbid a modification that inserts advertising, and it does not reach independent implementations.

Dedication of work to the public domain maximizes reuse and also permits any recipient to build a closed or advertising-financed derivative. That consequence may be acceptable for essays and for the specification text. For the reference implementation, the choice between public domain dedication and network copyleft trades unrestricted adoption against assurance that modifications offered as services remain inspectable. This draft recommends network copyleft for the reference host and client, a permissive license for verification libraries so that independent software can adopt them without friction, and public domain dedication for the specification and test vectors. The recommendation is a policy choice that remains open.

12.4 The Conformance Mark

Three instruments with different purposes are therefore kept independent. The software license protects access to implementation source. The technical specification, freely implementable by anyone, defines interoperable behaviour. Conformance certification establishes that an implementation satisfies the requirements on advertising and viewer autonomy. A proprietary or advertising-financed program may implement the record formats and verification procedures, wholly or in part, and exchange data with conforming software. It may not represent itself as conforming while violating a normative requirement. Interoperability thus does not wait on any organization’s approval.

Neither license nor specification can by itself control what may be called conforming. A name or mark held for that purpose can. Its terms SHOULD permit use by any implementation that passes the public conformance suite and observes the invariants, require withdrawal on demonstrated violation, publish the evidence and decision, and forbid the holder to charge for the right or to grant exceptions. The mark does not stop anyone from building something else. It denies the something else the ability to present itself to viewers as the system they chose.

Requirement 12.1 (Constraints on the steward)**.** The organization holding the mark MUST NOT itself operate the dominant host or index, MUST NOT derive revenue from advertising or sale of rank, MUST publish its finances and decisions, MUST commit in its governing instrument to transfer the mark only to a body bound by the same constraints, and MUST NOT hold exclusive rights that would let it relicense contributors' code on more restrictive terms.

A steward can still be captured, dissolved, or compelled. The final remedy is departure. Because identities are keys and digests, records are portable, specifications and test vectors are free, and implementations may be copied, participants can establish a successor without asking the old steward to release anything essential. Control of a familiar name and of an existing audience would still be real advantages to the captor, and the design reduces the cost of leaving without making it zero. Doctorow's account of platforms that first serve users and later extract from them identifies the loss of an inexpensive exit as a central enabling condition [11].

12.5 Amendment

Protocol changes proceed by public proposal, stated effect on each invariant, at least two interoperating implementations, test vectors, and a period for objection. A change that weakens an invariant MUST NOT be released under the same protocol name or mark, whatever majority approves it. Strengthenings are permitted when existing records retain their meaning. Clients MUST identify the protocol version of every record and MUST NOT reinterpret an old signature under new semantics.

Chapter 13

Security and Threat Model

13.1 Assumptions

Cryptographic functions are assumed to have their stated properties, viewers are assumed able to obtain one authentic client, and at least one log observer among those a viewer consults is assumed honest. Hosts, indices, contributors, and network intermediaries are untrusted. Authors are trusted to speak for their own works and for nothing else. Devices and operating systems are outside the boundary.

13.2 Threats

Threat	Response	Residual risk
Altered or inserted media	Digest and segment verification	Compromised client
Reversion to an old revision	Increasing head sequence numbers and logs	First contact with a stale head
Different histories shown to different viewers	Inclusion and consistency proofs; checkpoint exchange	Isolated viewers
Stolen author key	Threshold keys, rotation, revocation	Forgeries before revocation
False derivation map	Alignment checks and named signer	Content changes that preserve timing
Malicious media file	Maintained decoders run with minimal privilege	Unknown decoder defects
Resource exhaustion through composition graphs	Limits on depth, size, and render time; cycle rejection	Costly legitimate works
Proposal flooding	Rate limits, local queues, author choice	Labour of review
Impersonating work or alias	Identifier display and alias change notices	Human confusion
Surveillance by hosts	Prefetch, relays, multiple sources	Request metadata
Disappearance of dependencies	Pinned digests, mirrors, explicit report	Permanent loss

Threat	Response	Residual risk

Table 13.1: Principal threats, responses, and remaining exposure.

Compositions are untrusted input. A renderer MUST NOT interpret them as shell commands or pass their strings to a command interpreter, MUST construct tool invocations from validated structured values, and MUST refuse network retrieval during rendering except for declared assets obtained through the verifying layer. Containers and codecs have long histories of memory safety defects, so decoding SHOULD occur in a process without access to keys or personal records. Signing keys SHOULD never reside on a host that serves the public.

Chapter 14

Conformance

14.1 Profiles

Profile	Required capabilities
Core verifier	Canonical records, signatures, digests, head sequence, schema closure
Resolver	Core verifier, lineage validation, reference resolution
Listening client	Resolver, audio playback, offline storage, annotations, invariants of Chapter 6
Audiovisual client	Listening client, video, cuts, stems, local selection of variants
Session client	Audiovisual or listening client, correspondences, shared sessions
Host	Content retrieval, uniform publication, export, no substitution
Index	Unsold position, reproducible responses, published exclusions
Log	Inclusion and consistency proofs, public checkpoints

Table 14.1: Conformance profiles.

14.2 Claims and Their Scope

Definition 14.1 (Conformance claim)**.** *A* conformance claim *is a signed statement of component, version digest, profile, configuration, suite version, evidence, and claimant.*

Requirement 14.2 (Nontransferable claims)**.** A claim MUST NOT be extended to an enclosing product or service. The conformance mark MUST distinguish *format compatible*, meaning that records are exchanged correctly; *verifying component*, meaning that a named component passes its profile; and *viewer conforming*, meaning that the complete service and client a viewer meets satisfy every applicable invariant. Only the last may be described without qualification as conforming Revision-Native Media.

14.3 Test Cases

Each test supplies fixed input records and media and states the one expected result. Synthetic audio whose samples encode their own positions makes exact outcomes unambiguous.

Case	Expected result
Passage inserted wholly outside the contextual margin	Continued at the shifted time

Case	Expected result
Passage inserted within the contextual margin	Recontextualized
Marked sentence recorded again	Altered, no sample position
Passage divided around a removed interval, occurrence edges declared	Recontextualized for a surviving part; Altered for the removed interval
Passage removed	Withdrawn, with all ending revisions
Removed passage inserted again in a later revision	Rediscovered
Source interval used twice, marked use removed, two others remain	Ambiguous over both
Filtered asset with declared derivation and occurrence edge	Processed, with attributes
Filtered asset without derivation, lineage declared	Altered
Predecessor passage unaccounted for	Revision rejected
Negation removed under a technical label	Difference displayed beside label
Segment replaced by host	Playback halts for that material and reports failure
Appended segment absent from commitment	Segment refused
Record with an undefined tracking field	Record rejected
Unpaid user plays a local file with screen off and altered tempo	Capability available
Older head offered after newer one	Head rejected
Equal sequence numbers, differing revisions	Equivocation reported
Viewer excludes paid passages	Declared intervals omitted and omission indicated
Query response differing from recomputation	Discrepancy reported with evidence
Companion lacks material for an interval	Declared companion policy applied visibly
Two insertions at one anchor	Merge conflict
Illustration supporting a replaced explanation	Merge conflict
Complete export loaded on an unrelated host	All signatures and digests verify

Table 14.2: Initial conformance cases.

The following adversarial cases use valid signatures, valid schemas, or authentic assets in ways that must nevertheless fail.

Adversarial input	Required outcome
Authentic segment from another work	Event refused
Authentic segment at a wrong position or track	Event refused

Adversarial input	Required outcome
Authentic segment repeated by a host playlist	Repetition ignored or playback halted with report
Modified initialization segment or edit list	Verification failure
Valid companion not granted by the viewer	Nothing presented
Subtitle companion attempting audio output	Role refused
Signed vector image with an external tracking reference	No retrieval; feature refused
Source samples reused in an unrelated argument	Rediscovered, never Continued
Interval reference with only its first sample retained	Partial coverage reported
Parts of a referenced interval reversed	Order change reported
Passage deleted on one branch and kept on another before merge	Identical Contested result across implementations
Higher sequence head restoring an old revision	Reinstatement displayed
First contact offered a stale head without a recent checkpoint	Freshness unknown or expired
Signed rendition inconsistent with its composition	Render-verified status withheld
Unavailable variant replaced by one with added material	Authorization requested
Paid overlay on an unpaid passage	Overlay alone excluded where separable
Local pitch change routed through a paid service	Conformance failure
Client promoting its own service during playback	Conformance failure
Index client enlarging one result outside its published layout rules	Conformance failure
Core verifier claim applied to a whole platform	Claim rejected

Table 14.3: Adversarial negative cases.

14.4 Differential Fixtures

Two independent implementations given the same records, selections, and resources must report the same structural facts and authorization boundaries. Interfaces, rendering algorithms, and preferences may differ. Each fixture yields a canonical machine-readable result, or, where variation is intended, a stated set of admissible results.

Fixture	Required agreement
Original sample reused beside a restored descendant	Both candidates; summary Processed
Two derivation paths reaching different positions	Two candidates; Ambiguous if both witnessed
Signed but contradicted derivation	Same candidates; ineligible under default policy

Fixture	Required agreement
Nested and overlapping passages	Identical contextual projections
Identifier retained on a branch lacking passage descent	No witness
Fractional correspondence at an interval boundary	Identical rational containment
Reference spanning unequal sampling rates	Both coverage figures
Locally unique, jointly incompatible interval parts	Fragmented
Material trimmed in one revision and restored in the next	Fragmented
Witnessed descendant interval beside reused original samples	Descendant determines summary
Derivation maps crossing at the start of a part	Two candidates retained
Two derivation records applied within one revision	Witness carries both
Occurrence divided into contiguous identical parts	Unchanged projection
Interval of one undivided part	Point summary at its positions
Recontextualized commercial annotation	Not applied automatically
Gain envelope with temporal remapping	Attenuation follows source samples
Standing exclusion against explicit selection	Exclusion holds; conflict reported
Grant removed during buffered playback	No emission beyond declared bound
Corrected narration under default update classes	Accepted
Update adding a companion or paid declaration	Viewer asked
Update removing a paid declaration	Viewer asked
Superseded head with a fresh checkpoint	Greater head found or freshness unknown
Descendant with an earlier composition	Reinstatement
Substitute with unaccepted evidence kind	Viewer asked
Controller seeking into excluded material	Local exclusion holds
Host repeating a segment	Trace unchanged or halted

Table 14.4: Differential fixtures.

These tables specify intended behaviour. A partial reference implementation exercises selected resolution and authority cases on simplified records whose signatures and digests are assumed verified, and the questions it raised are settled in this version. A conformance suite further requires canonical records, signatures, media, one expected result for each case, and where relevant an expected network trace, and independent implementations written by other people.

A network monitor placed beside a client under test MUST observe no transmission other than requests for declared records and assets, configured index queries initiated by the viewer, log operations, and session messages the viewer enabled. The suite cannot demonstrate the absence of all misconduct in a program, and passing it establishes behaviour on the tested cases. Source availability, reproducible releases, and independent examination supply further evidence.

Chapter 15

Implementation Order and Open Decisions

15.1 Order of Construction

The order follows dependency. A reference resolver, a grant and transition evaluator, and the differential fixtures are the first executable objects, since they decide whether the semantics are determinate. Records, canonical serialization, signatures, and content addressing come first, followed by compositions over audio, lineage validation, and resolution. Transcripts and a verifying listening client come next, since they exercise the central guarantees at small cost. Correspondences, stem mixing, and sessions on a single machine and then several machines follow. Proposals and merging require a stable resolver. Multiple cameras, video segmentation, public logs, independent hosts, and indices extend the same foundations. The closed presentation, verified delivery, and no reporting requirements apply from the first client, because a capability once shipped acquires users who object to its removal.

15.2 Decisions Not Yet Made

Several matters require further work or choice. The contextual margin, default update classes, annotation bases, log counts and thresholds, and limits on candidate enumeration are provisional. The record schemas given in Appendix A are sketches, and numerical limits, error codes, and wire formats are unspecified. The tolerance by which derivation maps and nondeterministic renditions are compared has not been established empirically. The policy on declared paid passages may be tightened to prohibition at the price discussed in Chapter 6. Licensing of the reference implementation, the legal form of the steward, ownership and terms of the mark, and funding of at least one log and its observers are undecided. Key recovery that is humane without becoming a means of seizure is unsolved here. Live broadcast, interactive works, spatial video, branching narratives, restricted access encryption, and digital rights management are outside this draft; any access control mechanism later adopted must be examined for conflict with Invariants 2, 8, and 9. Storage of large multi-camera sources and compensation of authors have no assured solution.

15.3 Conclusion

The proposed medium keeps a work’s identity apart from every file that realizes it. States are immutable and signed, so a publication can improve without erasing what it was. References are resolved against source material, declared lineage, and context, so that change

produces an explicit answer that does not confuse surviving bytes with continuing meaning. Recordings, soundtracks, views, and texts remain separable, which allows other people to repair, translate, quieten, rearrange, and accompany a work while credit and endorsement stay distinct. Signed commitments establish authenticity and integrity; verification of renditions addresses fidelity; and an explicit account of grants, restrictions, and session transitions allows a player to refuse any sequence of uses that the viewer did not authorize. The defence ends where evidence ends: at undisclosed agreements, at compromised devices, and at institutions willing to break their own commitments before anyone can compare records. What remains in each case is the ability to see what was signed, to choose among independent implementations and services, and to leave with the work intact.

Appendix A

Record Sketches

Schema versions are numbered independently of drafts of this document; the sketches below are at schema version 0.8, reflecting changes to resolution and update classes. The following examples are illustrative and use shortened identifiers. Signature envelopes are omitted. Positions are integer samples, and rates are written as pairs of integers.

A.1 Revision

```
{
  "type": "revision",
  "version": "0.8",
  "work": "sha256:4be1...",
  "previous": ["sha256:90aa..."],
  "composition": "sha256:c03d...",
  "lineage": "sha256:77f2...",
  "published": "2026-11-28T12:00:00Z",
  "key": "ed25519:Qm9v...",
  "description": "Second paragraph recorded again",
  "labels": ["substantive"],
  "paid_statement": "none-declared"
}
```

A.2 Composition Fragment

```
{
  "type": "composition",
  "version": "0.8",
  "occurrences": [
    {"id": "occ-0042", "asset": "sha256:1d9e...",
     "source": [480000, 710400], "start": [25, 1],
     "rate": [1, 1], "track": "dialogue", "transforms": []}
  ],
  "passages": [
    {"id": "passage-0042", "occurrences": ["occ-0042"],
     "script": "sha256:5a10...", "observed": "sha256:8c44...",
     "references": ["ref-017"], "consideration": null}
  ],
```

```
  "support": [["occ-0051", "passage-0042"]]
}
```

A.3 Lineage

```
{
  "type": "lineage",
  "version": "0.8",
  "from": "sha256:90aa...",
  "edges": [
    {"source": "passage-A", "target": "passage-A", "kind": "preserved"},
    {"source": "passage-B", "target": "passage-B2", "kind": "replaced"},
    {"source": "passage-C", "target": "passage-C1", "kind": "split"},
    {"source": "passage-C", "target": "passage-C2", "kind": "split"}
  ],
  "occurrences": [
    {"source": "occ-0007", "target": "occ-0107"}
  ],
  "deleted": ["passage-D"],
  "inserted": ["passage-intro"]
}
```

A.4 Correspondence

```
{
  "type": "correspondence",
  "version": "0.8",
  "a": "sha256:aa01...",
  "b": "sha256:bb02...",
  "pairs": [
    {"a": [[120, 1], [900, 1]], "b": [[0, 1], [780, 1]],
     "kind": "same"},
    {"a": [[900, 1], [930, 1]], "b": [[780, 1], [810, 1]],
     "kind": "altered-level"},
    {"a": [[930, 1], [2520, 1]], "b": [[810, 1], [2400, 1]],
     "kind": "same"}
  ],
  "only_a": [[[0, 1], [120, 1]]],
  "only_b": [[[2400, 1], [2580, 1]]],
  "tolerance_ms": 80,
  "method": "manual",
  "companion_policy": {"only_b": "suspend-leader"}
}
```

A.5 Head, Grant, and Claim

```
{
```

```
  "type": "head",
  "version": "0.8",
  "work": "sha256:4be1...",
  "revision": "sha256:e51c...",
  "sequence": 7,
  "valid_from": "2026-11-28T12:00:00Z",
  "valid_until": "2026-12-28T12:00:00Z",
  "key": "ed25519:Qm9v..."
}
{
  "type": "grant",
  "version": "0.8",
  "resource": "sha256:7ab3...",
  "revision_rule": "pinned",
  "capabilities": [["subtitle", "render-text", "lower-third"]],
  "update_classes": ["media", "text"]
}
{
  "type": "claim",
  "version": "0.8",
  "component": "example-verifier",
  "build": "sha256:19fe...",
  "profile": "core-verifier",
  "level": "verifying-component",
  "configuration": "sha256:aa90...",
  "suite": "0.8",
  "evidence": "sha256:c771...",
  "claimant": "ed25519:Zm9v..."
}
```

A grant is a local record held by the viewer's client and is not published.

A.6 Gain Envelope

```
{
  "type": "gain-envelope",
  "version": "0.8",
  "target_asset": "sha256:1d9e...",
  "intervals": [
    {"source": [43200000, 44640000], "gain_db": -30,
     "ramp_samples": 2400, "reason": "sudden-loud-event",
     "basis": "acoustic"}
  ]
}
```

Bibliography

[1] Webber, Christine Lemmer, and Jessica Tallon, eds. *ActivityPub*. W3C Recommendation, 2018. `https://www.w3.org/TR/activitypub/`.

[2] Free Software Foundation. *GNU Affero General Public License, Version 3*. 2007. `https://www.gnu.org/licenses/agpl-3.0.html`.

[3] Sanderson, Robert, Paolo Ciccarese, and Benjamin Young, eds. *Web Annotation Data Model*. W3C Recommendation, 2017. `https://www.w3.org/TR/annotation-model/`.

[4] Benet, Juan. "IPFS: Content Addressed, Versioned, P2P File System." arXiv:1407.3561, 2014.

[5] Brin, Sergey, and Lawrence Page. "The Anatomy of a Large-Scale Hypertextual Web Search Engine." *Computer Networks and ISDN Systems* 30 (1998): 107–117. The appendix on advertising appears in the full version of the paper.

[6] Coalition for Content Provenance and Authenticity. *Content Credentials: C2PA Technical Specification*, Version 2.3. 2026. `https://spec.c2pa.org/`.

[7] Creative Commons. *Creative Commons Attribution 4.0 International Public License*. 2013. `https://creativecommons.org/licenses/by/4.0/`.

[8] Chacon, Scott, and Ben Straub. *Pro Git*. 2nd ed. New York: Apress, 2014.

[9] Crosby, Scott A., and Dan S. Wallach. "Efficient Data Structures for Tamper-Evident Logging." In *Proceedings of the 18th USENIX Security Symposium*, 317–334. 2009.

[10] International Organization for Standardization. *ISO/IEC 23009-1: Dynamic Adaptive Streaming over HTTP (DASH), Part 1: Media Presentation Description and Segment Formats*. Geneva: ISO.

[11] Doctorow, Cory. *Enshittification: Why Everything Suddenly Got Worse and What to Do About It*. New York: MCD, 2025.

[12] European Broadcasting Union. *EBU R 128: Loudness Normalisation and Permitted Maximum Level of Audio Signals*. Geneva: EBU.

[13] FFmpeg Developers. *FFmpeg Documentation*. `https://ffmpeg.org/documentation.html`.

[14] National Institute of Standards and Technology. *Secure Hash Standard (SHS)*. FIPS PUB 180-4, 2015.

[15] Merkle, Ralph C. "A Digital Signature Based on a Conventional Encryption Function." In *Advances in Cryptology: CRYPTO '87*, 369–378. Berlin: Springer, 1988.

[16] 9to5Google. "YouTube Looks to Be Testing Server-Side Ad Injection to Counter Ad Blockers." June 12, 2024. https://9to5google.com/2024/06/12/youtube-ad-injection.

[17] Open Source Initiative. *The Open Source Definition (Annotated)*. https://opensource.org/osd-annotated.

[18] Framasoft. *PeerTube Documentation*. https://docs.joinpeertube.org/.

[19] Reproducible Builds Project. *Definitions: When Is a Build Reproducible?* https://reproducible-builds.org/docs/definition/.

[20] Bradner, Scott. *Key Words for Use in RFCs to Indicate Requirement Levels*. RFC 2119, 1997.

[21] Mills, David, Jim Martin, Jack Burbank, and William Kasch. *Network Time Protocol Version 4: Protocol and Algorithms Specification*. RFC 5905, 2010.

[22] Valin, Jean-Marc, Koen Vos, and Timothy Terriberry. *Definition of the Opus Audio Codec*. RFC 6716, 2012.

[23] Van de Sompel, Herbert, Michael Nelson, and Robert Sanderson. *HTTP Framework for Time-Based Access to Resource States: Memento*. RFC 7089, 2013.

[24] Josefsson, Simon, and Ilari Liusvaara. *Edwards-Curve Digital Signature Algorithm (EdDSA)*. RFC 8032, 2017.

[25] Leiba, Barry. *Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words*. RFC 8174, 2017.

[26] Pantos, Roger, and William May. *HTTP Live Streaming*. RFC 8216, 2017.

[27] Rundgren, Anders, Bret Jordan, and Samuel Erdtman. *JSON Canonicalization Scheme (JCS)*. RFC 8785, 2020.

[28] Laurie, Ben, Eran Messeri, and Rob Stradling. *Certificate Transparency Version 2.0*. RFC 9162, 2021.

[29] van Beurden, Martijn Q. C., and Andrew Weaver. *Free Lossless Audio Codec (FLAC)*. RFC 9639, 2024.

[30] Bulterman, Dick, et al., eds. *Synchronized Multimedia Integration Language (SMIL 3.0)*. W3C Recommendation, 2008. https://www.w3.org/TR/SMIL3/.

[31] Ramachandran, Ajay, and Contributors. *SponsorBlock*. https://sponsor.ajay.app/.

[32] World Wide Web Consortium. *Requirements for Media Timed Events*. W3C Group Note. https://www.w3.org/TR/media-timed-events/.

[33] W3C Multi-device Timing Community Group. *Timing Object*. Final Community Group Report, December 3, 2024. https://www.w3.org/community/reports/webtiming/CG-FINAL-timingobject-20241203/.

[34] Video Sync. *Browser Extension Listing*. Chrome Web Store. https://chromewebstore.google.com/detail/video-sync/npplmbdcndipnaemaoljdppjodakaicn.

[35] World Wide Web Consortium. *Web Content Accessibility Guidelines (WCAG) 2.2*. W3C Recommendation, 2023. https://www.w3.org/TR/WCAG22/.

[36] World Wide Web Consortium. *WebVTT: The Web Video Text Tracks Format*. https://www.w3.org/TR/webvtt1/.

[37] Wikipedia Contributors. "Don't Be Evil." *Wikipedia*. https://en.wikipedia.org/wiki/Don%27t_be_evil. Accessed November 28, 2026.