# The Digital Audio Assets Protocol (DAAP)

**A Protocol for Structuring, Preserving, and Distributing Session-Derived Music Assets**
**Version:** v1.0k
**Prepared for:** The Digital Audio Assets & Production Attribution Consortium
**Prepared by:** Vanessa N. Adams

### Abstract

In this paper, DAAP refers to the Digital Audio Assets Protocol and is unrelated to Apple's historical "Digital Audio Access Protocol.". The Digital Audio Assets Protocol is a standard-based framework for defining, packaging, and distributing modern musical works as structured digital assets rather than flat audio files. Contemporary music production is inherently systemic, emerging from complex environments composed of digital audio workstations, plugin chains, routing architectures, automation systems, hybrid hardware, and increasingly, AI-assisted tools. However, prevailing distribution formats preserve only the rendered waveform, collapsing the production environment and externalizing attribution, provenance, and structural identity.

DAAP addresses this structural limitation by introducing a containerized asset model that binds audio payloads to a canonical manifest and an integrity layer. The canonical manifest captures the essential structure of the session environment, including track topology, routing summaries, plugin chain identities, contributor roles, rights pointers, and provenance, in a durable, machine-readable form. The integrity layer enables independent verification and authentication of the asset across distribution systems, archives, and future playback environments.

## INTRODUCTION

Music production has undergone a profound structural transformation. What was once a linear, tool-bounded process centered around isolated recordings has evolved into a complex, systemic environment composed of interdependent digital and physical components. Today's musical works emerge from digital audio workstations, plugin chains, synthesis engines, routing architectures, automation systems, hybrid hardware, and increasingly, AI-assisted tools. These components do not merely support the creative process, they *define* it. They shape the identity, behavior, and long-term value of the work.

Yet despite this evolution, the industry continues to distribute music as if it were still produced in the era of tape and consoles. The rendered audio file, WAV, AIFF, FLAC, MP3, remains the canonical unit of distribution. These formats were designed to preserve *sound*, not *structure*. They carry the final waveform but discard the production system that created it. They cannot represent session topology, plugin identities, routing relationships, automation logic, contributor roles, or provenance. They cannot express the creative and systems decisions that shape the work. They cannot survive the complexity of modern production.

This mismatch, between systemic production and flat-file distribution, is no longer a minor inconvenience. It is a workflow breakage at export where credits and recall context become external. When the production system collapses into a single waveform, attribution becomes optional, provenance becomes fragile, and identity becomes externalized. Metadata becomes a patchwork of optional fields scattered across DAWs, distributors, and platforms, none of which can reliably survive editing, transcoding, or archival migration. The result is a persistent erosion of trust, transparency, and economic clarity.

At the same time, the industry is entering a new era of session-aware, context-preserving digital assets. These assets do not treat the waveform as the whole story. They recognize that the structure of the production environment, the topology of the session, the identities of the tools, the lineage of the work, is essential to understanding, valuing, and preserving the music itself. As these assets move from concept to deployment, the limitations of flat files become more than a technical inconvenience; they become a barrier to innovation, attribution, and long-term industry health.

The Digital Audio Assets Protocol (DAAP) is designed to address this structural gap. DAAP introduces a standards-based, containerized asset model that preserves the essential structure of the production environment in a durable, interoperable form. It binds audio payloads to a canonical manifest and an integrity layer, ensuring that identity, provenance, and production topology remain attached to the work itself, not scattered across external systems or lost in translation.

DAAP is not a DAW format, not a plugin host, and not a DRM system. It does not attempt to recreate the full editing environment or expose proprietary DSP algorithms. Instead, it preserves the approved, distributable representation of the session, the structural information necessary for attribution, provenance, and long-term interpretability, while remaining compatible with existing distribution standards such as DDEX, ISRC, and EBU metadata frameworks.

As the industry confronts the challenges of AI-mediated production, dynamic rendering, and large-scale catalog operations, the need for durable, asset-bound identity becomes urgent. DAAP provides the systems foundation required for transparent attribution, verifiable provenance, and future-proof distribution. It is the missing layer between the studio and the ecosystem, the structural substrate upon which the next generation of music infrastructure will be built.

## SECTION 1: THE PROBLEM WITH FLAT-FILE DISTRIBUTION

The modern music industry continues to rely on a distribution model built around a fundamental assumption: that a rendered audio file is the complete and authoritative representation of a musical work. This assumption made sense in an era when recordings were produced through linear workflows and fixed hardware chains. But in today's environment, where music is created through complex, interdependent digital systems, the flat audio file has become a structural bottleneck.

A WAV or AIFF file is excellent at preserving the *sound* of a finished mix. It is not designed to preserve the *system* that produced it. When a session is rendered to a flat file, the entire production topology collapses into a single waveform. The relationships, decisions, tools, and structures that define the work disappear. This collapse is not a minor technical detail; it is the root cause of attribution loss, metadata fragility, and long-term interpretability failures across the industry.

---

### 1.1 The Collapse of Production Topology

A rendered audio file cannot represent:

- the track and stem structure
- routing relationships
- plugin chain identities
- automation and control logic
- tool usage and contribution pathways
- the lineage of decisions that shaped the sound
- the context in which the work was created

These elements are not optional. They are the *production topology*, the structural identity of the work. When topology collapses, attribution collapses with it. The tools and contributors that shaped the music become invisible. The decisions that define the work become irretrievable. The production environment becomes untraceable.

This is why metadata loss is not a metadata problem; it is a topology problem. No number of optional metadata fields can compensate for the structural collapse that occurs when a session is flattened into a waveform.

---

### 1.2 Distribution-Stage Degradation

Even when metadata is present in a rendered file, it rarely survives the distribution pipeline intact. During ingestion, transcoding, and repackaging:

- iXML and BWF metadata are often stripped
- platform-specific schemas overwrite or truncate fields
- distributor pipelines normalize or discard nonstandard chunks
- attribution is reconstructed externally rather than preserved internally
- provenance becomes dependent on external databases

This creates a fragile, inconsistent ecosystem where identity is maintained through reconciliation rather than through the asset itself. The result is a persistent erosion of trust and transparency.

---

### 1.3 The Structural Consequences

As production volume increases and versions proliferate, alternate edits, stem packs, spatial mixes, clean/instrumental variants, and authorized derivatives, the ability to track identity and intent becomes more important than signal similarity alone. Studio deliverables increasingly need a way to preserve which version is approved, what it derives from, who contributed, and what toolchain references shaped the final sound. DAAP treats that information as part of the deliverable, not as external paperwork of flat audio files.

The consequences of flat-file distribution are systemic:

- **Attribution gaps** - contributors and tools disappear from the asset.
- **Economic opacity** - valuation becomes guesswork rather than evidence-based.
- **Catalog instability** - provenance cannot be reliably demonstrated.
- **Archival fragility** - future systems cannot interpret the work's structure.
- **AI indistinguishability** - without provenance, human and machine outputs blur.
- **Platform dependency** - identity becomes externalized and platform-controlled.

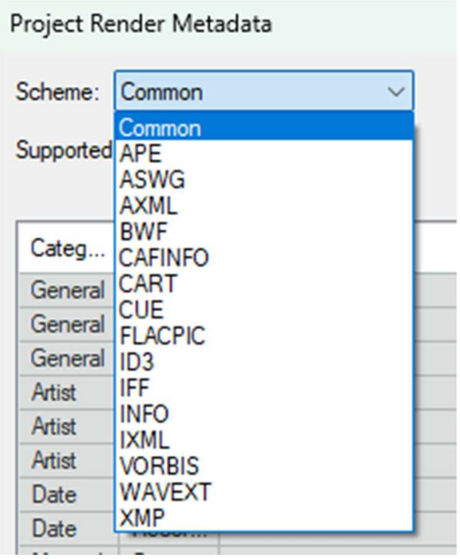
These failures are not the result of bad metadata practices. They are the result of a distribution model that cannot carry the structure of modern production.

---

### 1.4 Metadata Render/Export Schemes in Modern Audio Workflows

Most professional production environments can write metadata at the moment audio is rendered or exported. Rather than treating metadata as a single universal "tag set," these systems expose multiple metadata schemes because different file containers, broadcast standards, music platforms, and archival pipelines expect metadata in different structures.

The menu shown here represents common export-time choices for embedding (or attaching) metadata so that a rendered deliverable can carry identifiers, descriptive fields, time references, cueing, artwork, and production notes in a way that downstream tools are more likely to recognize.



Project Render Metadata

Scheme: Common

**Metadata scheme types and where they're used**

- **Common:** A practical "best effort" set of core fields (title, artist, album, date, etc.) written using the most appropriate mechanism for the chosen file format. Good for general delivery when standard tags without targeting a specific broadcast or platform schema are wanted.

- **APE**: APEv2 tags. Commonly used in Monkey's Audio (APE) and also supported in some WAV workflows and tagging tools. Favored in some archival/user libraries because it supports flexible fields, but it's not a broadcast standard.
- **ASWG**:  AudioMD / ASWG (Association for Saving Western Civilization) metadata practices used in some archival preservation and library workflows (often adjacent to Broadcast WAV preservation practices). Typically encountered in institutional archiving contexts more than consumer music platforms.
- **AXML**: The AXML chunk in Broadcast WAV (BWF), used to embed XML metadata inside a WAV/BWF container. Common in broadcast, film/post, and archival contexts where structured metadata must travel with the audio.
- **BWF:** Broadcast Wave Format metadata, primarily the bext chunk (and related BWF conventions). Used in broadcast, post-production, and archival pipelines for time reference, originator, description, and other exchange-critical fields.
- **CAFINFO:** Metadata fields specific to CAF (Core Audio Format), used mainly in Apple/Core Audio workflows and tools that support CAF's container metadata.
- **CART:** Radio automation metadata (often associated with the NAB "CART" chunk) used by broadcast playout systems for cut numbers, categories, rotation, and traffic/automation fields. Common in radio station delivery.
- **CUE:** Cue markers (cue points/regions). Used for track indexing, edits, and navigation points inside a file—helpful for long-form renders, assembled programs, or mastering/prep where markers need to travel with the file.
- **FLACPICT:** Embedded picture/artwork blocks for FLAC. Used when delivering FLAC masters or promo files with cover art embedded for library players and distribution tools that read FLAC picture metadata.
- **ID3:** The most common tag format for MP3 and also used by AAC in MP4/M4A contexts in various ways (platform dependent). Used widely for consumer/music platform metadata: title, artist, album, artwork, etc.
- **IFF:** Interchange File Format metadata, associated with AIFF and related IFF-based containers. Used in workflows where AIFF is the deliverable and you want tags stored in AIFF's native chunk structures.
- **INFO:** The RIFF INFO chunk set. Common legacy metadata for WAV/RIFF files (title, artist, comment, etc.). Often recognized by older Windows-oriented tooling; not as robust as BWF for professional interchange.
- **XML:** Generic XML metadata payloads (not necessarily BWF aXML). Used when a pipeline expects a structured XML document embedded or attached;

compatibility depends heavily on the receiving system.

- **VORBIS:** Vorbis Comments used by Ogg Vorbis and also by FLAC (FLAC uses Vorbis Comment blocks for text tags). This is the standard tagging method for Ogg/FLAC ecosystems.
- **WAVEXT:** WAVE “extensible” and related extended WAV structures used for multichannel/extended format descriptors (more about format structure than descriptive tags). Relevant for pro deliverables where channel layouts and extended format details matter.
- **XMP**: Adobe’s Extensible Metadata Platform, used widely across media workflows (photo/video/document) and sometimes adopted in audio asset management environments. Most relevant when audio is being managed in media asset management or cross-media pipelines that already standardize on XMP.

### 1.5 Why This Problem Cannot Be Solved by Metadata Alone

Attempts to solve attribution and provenance through metadata alone, whether through BWF, iXML, XMP, or platform-specific schemas such as C2PA, fail for three reasons:

1. Metadata is optional
   If it is not required, it will not be consistently present.
   In most studio and distribution pipelines, metadata fields are treated as “best effort.” Engineers may fill them in, distributors may normalize them, and platforms may ingest only what fits their internal models. But because the audio remains playable and commercially deliverable even when metadata is incomplete, there is no systemic forcing function that ensures consistent capture. Over time, and especially across handoffs, optional fields become uneven, which means that downstream systems cannot depend on them as an authoritative source of identity.

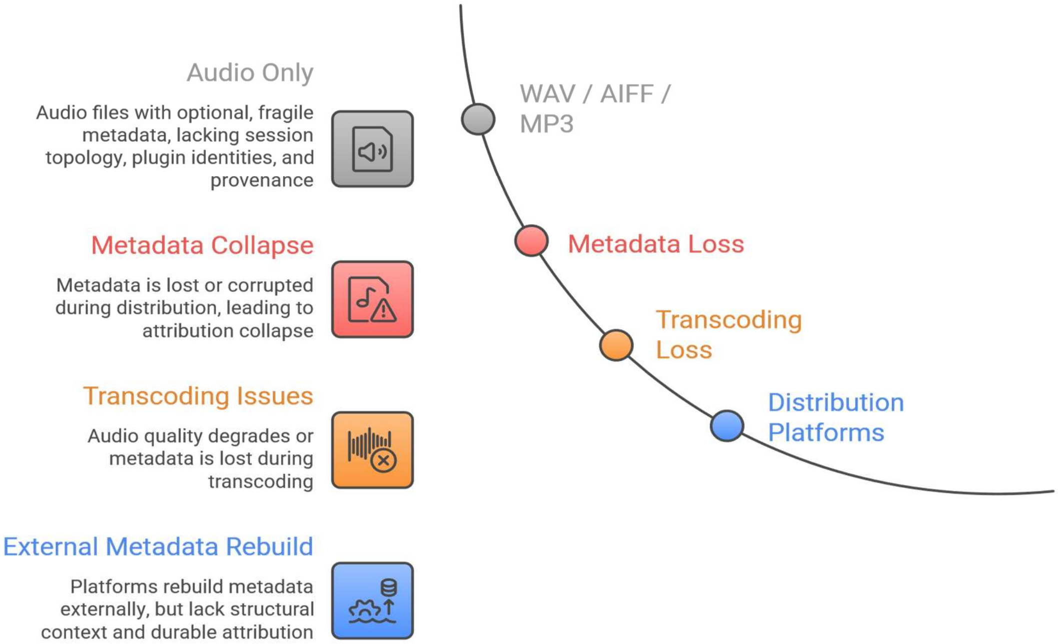
2. Metadata is fragile. It is routinely stripped, overwritten, or truncated during distribution. Even when metadata is present at export, it often passes through tooling layers that do not preserve it uniformly. Common operations such as transcoding to AAC/Ogg variants, repackaging into MP4 containers, generating previews, or re-rendering files for different territories can drop metadata fields entirely or collapse them into a smaller set of supported tags. Some systems rewrite files and preserve only what they recognize, while others translate metadata into platform databases and no longer treat the file as the canonical carrier. The result is that embedded metadata behaves more like a convenience than a durable identity layer.

3. Metadata is externalized
   It lives in separate systems, not in the asset itself. In modern distribution, the unit of truth is rarely “the file.” It is a database record at a distributor, a DSP, a publisher administrator, or a rights registry that points to the file. Once meaning is externalized in that way, identity becomes reconciliation work. Each system maintains its own IDs and mappings, and alignment depends on cross-database consistency rather than on verifying a single authoritative object. This is why even good metadata practices still produce disputes: the same work can be represented correctly in one database and ambiguously in another, with no intrinsic mechanism to prove which representation governs.

DAAP addresses these limitations by making identity, structure, and provenance intrinsic to the asset, not external to it. Instead of treating metadata as optional annotations that may or may not survive, DAAP treats the manifest and integrity layer as the authoritative definition of the deliverable. Audio remains an essential payload, but the asset’s meaning: who made it, what version it is, how it was derived, and how it can be verified and travels with the asset itself.

---

Flat File Distribution Challenges



Audio Only

Audio files with optional, fragile metadata, lacking session topology, plugin identities, and provenance

Metadata Collapse

Metadata is lost or corrupted during distribution, leading to attribution collapse

Transcoding Issues

Audio quality degrades or metadata is lost during transcoding

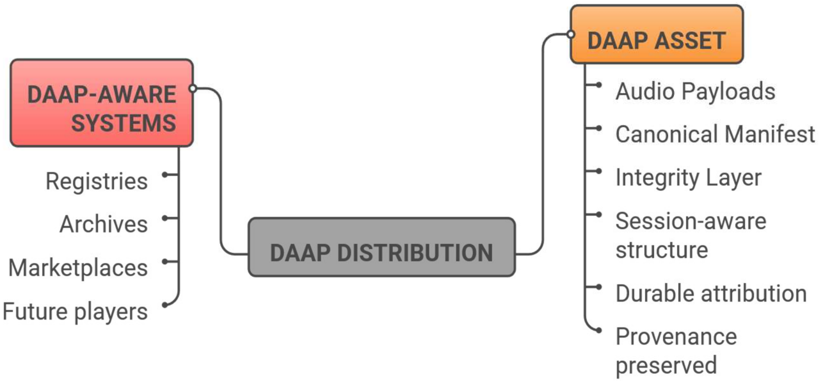
External Metadata Rebuild

Platforms rebuild metadata externally, but lack structural context and durable attribution

## SECTION 2: DAAP PROTOCOL OVERVIEW

The Digital Audio Assets Protocol (DAAP) defines a new foundational unit for music distribution: the session-derived digital asset. Unlike traditional audio files, which preserve only the final waveform, a DAAP asset preserves the structural, contextual, and attributional information that defines how the work was created. DAAP does this through a standards-based container model that binds audio payloads to a canonical manifest and an integrity layer, ensuring that identity and provenance remain attached to the asset across every stage of its lifecycle.

DAAP Distribution: Asset and System Interaction



DAAP is not a DAW format, not a plugin host, and not a replacement for existing distribution standards. It does not attempt to recreate the full editing environment or expose proprietary DSP algorithms. Instead, DAAP captures the approved, distributable representation of the session, the structural information necessary for attribution, provenance, and long-term interpretability, in a durable, interoperable form.

At its core, DAAP defines three interlocking components:

1. **Audio Payloads**
   The distributable audio representations of the work, including the master mix, session-derived stems, and optional preview assets. These payloads are the audible expression of the asset.

2. **Canonical Manifest**
   A machine-readable, schema-versioned document that captures the structural identity of the work: contributor roles, session topology, plugin chain identities, rights pointers, provenance, and versioning. The manifest is the authoritative definition of the asset.

3. **Integrity Layer**
   A set of cryptographic hashes,

optional digital signatures, and verification metadata that ensure the asset can be authenticated, validated, and trusted across distribution systems, archives, and future playback environments.

Together, these components form self-describing, self-verifying digital objects that can survive editing, transcoding, ingestion, redistribution, and archival migration without losing its identity or structural meaning.

DAAP is designed to be implementation-agnostic. It does not prescribe how DAWs must operate, how plugins must be built, or how distributors must ingest assets. Instead, it defines the minimum structural guarantees required for a musical work to be considered a DAAP asset. This ensures that the protocol can be adopted incrementally, without disrupting existing workflows or requiring immediate industry-wide changes.

DAAP also establishes clear protocol boundaries. It does not define pricing, licensing, valuation, or economic participation. These are governance-level decisions that belong to the consortium. DAAP provides the systems substrate upon which attribution standards, valuation frameworks, and ethical guidelines can be built, but it does not dictate those frameworks itself.

By defining music as a structured digital asset rather than a flat file, DAAP enables a new generation of tools, registries, archives, and marketplaces to operate with clarity, transparency, and long-term stability. It provides the missing layer of infrastructure required for session-aware durable attribution, and verifiable provenance.

### 2.1 DAAP Asset Structure

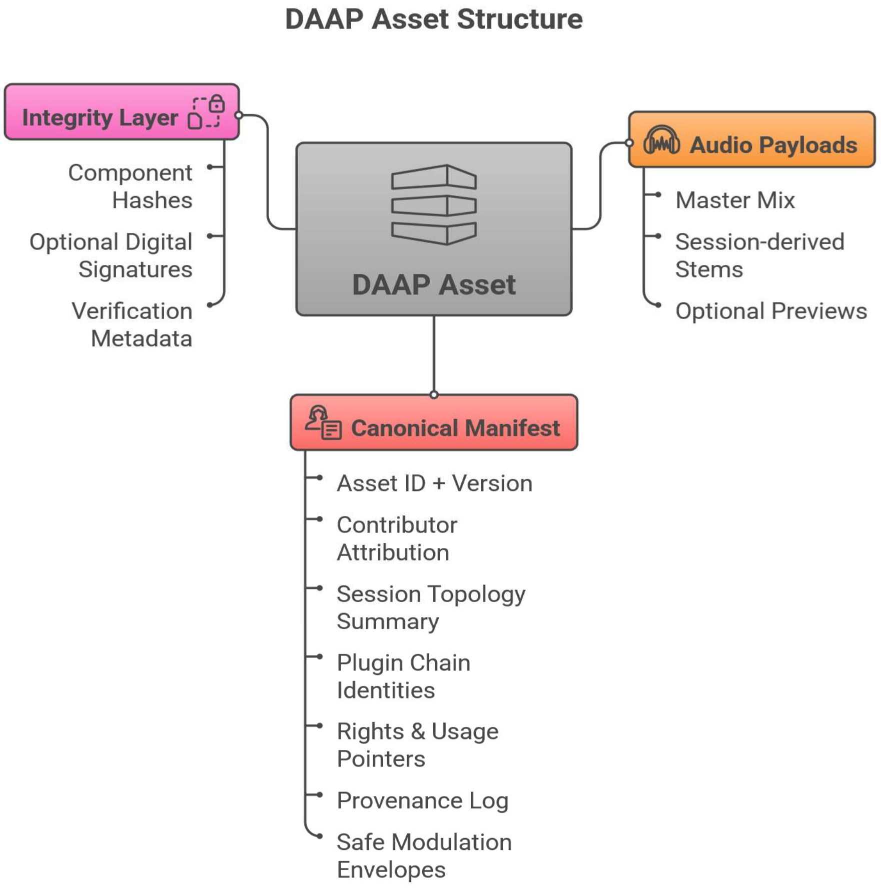
A DAAP asset is defined as a **self-contained, session-derived digital object** composed of three tightly bound components: audio payloads, a canonical manifest, and an integrity layer. This structure replaces the traditional flat audio file as the unit of distribution, ensuring that the audible output of a work is inseparable from the structural, attributional, and provenance information that defines how the work was created.

Each component serves a distinct role, but the asset derives its meaning and durability from their combined operation as a single, verifiable object.

### 2.2 What a DAAP Deliverable Contains

A DAAP deliverable is designed to behave like a studio-grade “package,” not a loose file export. Instead of handing off a master WAV plus scattered recall notes and spreadsheets, DAAP produces a single asset object that keeps the work’s identity and essential context attached to the audio.

At a minimum, a DAAP asset contains three things. First, it contains one or more audio payloads, such as the approved stereo master and any required alternates (clean edit, instrumental, alt mix, etc.). Second, it contains a canonical manifest that describes what the asset is, who contributed, what version it represents, and what it derives from. Third, it contains an integrity layer, hashes and signatures, that allow the recipient to verify that the manifest and its referenced payloads have not been altered.



The goal of DAAP is not to ship a DAW session to end users and not to require plugins to be executable downstream. The goal is to ensure that once audio leaves the session environment, the work retains enough structured context to remain attributable, version-legible, and verifiable.

DAAP is intentionally defined as a protocol rather than a product or platform. This distinction is essential:

- A product is controlled by a single entity.
- Protocol is governed collectively and implemented broadly.

DAAP's role is to define the rules of structure, identity, and verification, not to enforce licensing, dictate business models, or prescribe playback behavior. This separation ensures that DAAP remains neutral, extensible, and compatible with diverse tools and workflows.

### 2.3 Protocol Guarantees

DAAP provides the following guarantees:

- **Durable attribution** - contributor roles and tool identities remain attached to the asset.

- **Structural preservation** - session topology is captured in canonical form.

- **Provenance continuity** - the lineage of the work is recorded and verifiable.

- **Interoperability** - assets can be mapped into existing standards (DDEX, ISRC, EBU).

- **Long-term interpretability** - future systems can understand the asset without the original DAW or plugins.

- **Integrity and trust** - assets can be authenticated and validated independently.

These guarantees form the foundation for the next generation of music distribution and archival systems.

### 2.4 Why DAAP Is Not "Another Database"

DAAP is not proposed as a new centralized database or a replacement for existing catalog systems. Contemporary music infrastructure already depends on many databases at labels, distributors, DSPs, publishers, CMOs, and fingerprinting vendors and these systems are often internally coherent while remaining globally non-authoritative. In that environment, the same recording is represented multiple times under different identifiers, and coherence is maintained through synchronization and reconciliation rather than through verification of a single object.

DAAP's architectural distinction is that it relocates authority upstream into the asset itself. The canonical manifest is treated as the primary carrier of identity, attribution, and lineage, while databases become secondary indexes and operational mirrors rather than the sole sources of truth.

This does not eliminate the need for registries or catalog management; it changes what those systems point to. Instead of asking which database record should be trusted at a given moment, DAAP enables systems to validate a self-describing asset whose identity and provenance claims are designed to persist through downstream transformations.

## SECTION 3: DAAP ASSET ARCHITECTURE

A rendered audio file is an accurate representation of sound, but it is not an accurate representation of the production system that produced that sound. In a studio context, this matters because a finished record is rarely "just audio." It is the sum of routing choices, gain staging, processing chains, automation decisions, editorial work, and approvals across versions. When the work is exported into a flat deliverable (a master WAV, an alternate mix, or a set of stems), the production topology that made the work intelligible inside the session stops traveling with the deliverable.

This is not a critique of WAV or any other audio format. WAV and related containers were built to preserve and transport audio samples for playback and exchange. They were not built to preserve the relational context that studio professionals depend on for attribution, recall, and version identity. DAAP exists to preserve that context at export without requiring the session file itself to be distributed.

### 3.1 What "Flattening" Means in Recording Studio Systems Terms

Inside a DAW, a mix exists as a graph: tracks feed buses, buses feed the master, processors and instruments operate in ordered chains, and automation changes parameters over time. When you render the mix, the DAW evaluates that graph and produces a linear stream of audio samples. The resulting file preserves the outcome, not the graph.

The practical result is familiar to any engineer: after export, the deliverable can still sound perfect, but it no longer "knows" what it is. It does not carry the relationships that distinguish one approved version from another, or one stem package from another. Any meaning that remains is reconstructed externally from file names, folder structures, delivery emails, and institutional memory.

DAAP defines the deliverable differently. It treats the export not as "a file," but as "an asset state" whose identity and essential production context are attached to the deliverable itself.

### 3.2 Embedded Metadata Is Not the Same as Asset Identity

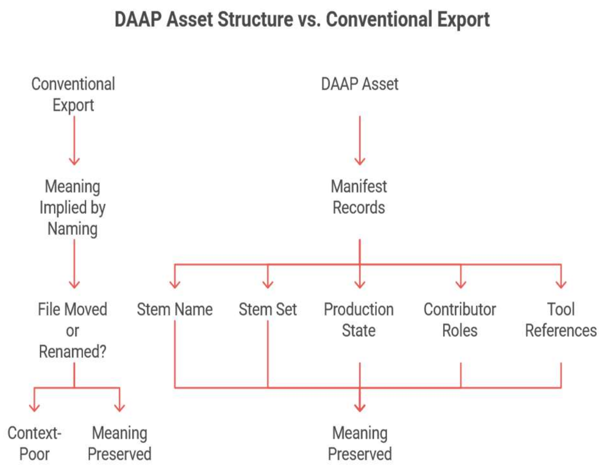
Audio containers can store metadata, and professional workflows often rely on embedded fields in certain environments. However, embedded metadata is optional, inconsistently supported across tools, and commonly reduced during routine operations such as transcoding, editing, repackaging, or delivery through third-party systems. Even when a file retains some embedded fields, those fields are rarely authoritative for studio-grade questions such as which version was approved, which stems belong to which master, who contributed in what role, and what the intended lineage of derivatives is.

DAAP does not depend on fragile embedded tags for its primary identity. It uses a canonical manifest that remains authoritative across the asset package and binds to payloads through integrity checks. The audio files remain audio files; the asset carries the identity.

### 3.3 "kick.wav" Versus a Context-Bearing Component

A stem export illustrates the limitation clearly. In a conventional export, a stem becomes a standalone file such as "kick.wav." Its meaning is implied by naming conventions and folder placement. If that file is moved, renamed, combined with other stems, or separated from its delivery notes, the file remains playable but becomes context-poor.

In a DAAP asset, the same stem is treated as a component within a declared structure. The manifest records that this file is the “Kick Stem” for a specific asset version, that it belongs to a defined stem set, and that it derives from the same approved production state as the master. It can also associate contributor roles and tool references as identifiers, without shipping executable session dependencies. The objective is not to recreate the session; it is to prevent stems and alternates from becoming detached from their meaning.



### 3.4 What DAAP Preserves (and What It Does Not)

DAAP preserves the information studio professionals routinely lose at export: the identity of the approved deliverable, the relationship between deliverables (master, alternates, stems), structured contributor roles, tool references as identifiers, and version lineage markers. It also preserves integrity information that allows a recipient to verify whether a package has been altered or whether a master file is still the approved one.

DAAP does not claim to preserve DAW editability. It does not promise that a mix can be reopened with working plugins in an unknown future environment. It does not ship plugin binaries, sessions, or licensing dependencies to consumers. DAAP preserves the deliverable as an attributable, version-legible, verifiable object, and it preserves enough structural context to make later work reissues, derivatives, and archives more reliable.

### 3.5 Canonical Manifest Canonical Manifest (What the Asset “Says” It Is)

The Manifest is the authoritative description of the deliverable. It is the part that makes the DAAP package more than a folder of files. In the studio context, the manifest records the identity and scope of what was approved and delivered, including the version state of the work.

A DAAP manifest can carry, at an appropriate disclosure tier, the essential “liner notes” information that often gets lost at export: contributors and roles (artist, producer, mix engineer, mastering engineer, musicians where applicable), tool references (plugin and instrument identifiers and versions as references), and a synopsis of how the deliverable relates to the session output (for example, that the master was derived from the approved mix state and that certain stems were derived from that same state). The manifest also captures dates and approval events that help distinguish “work-in-progress exports” from the approved deliverable.

For studio professionals, the practical value is that the manifest becomes a reliable companion to the audio. A future remaster, a label reissue, a sync request, or a derivative package can reference the manifest directly rather than reconstructing intent from folder names and memory.

The canonical manifest is the authoritative definition of the asset. It is a machine-readable, schema-versioned document that captures the structural identity of the work. The manifest includes:

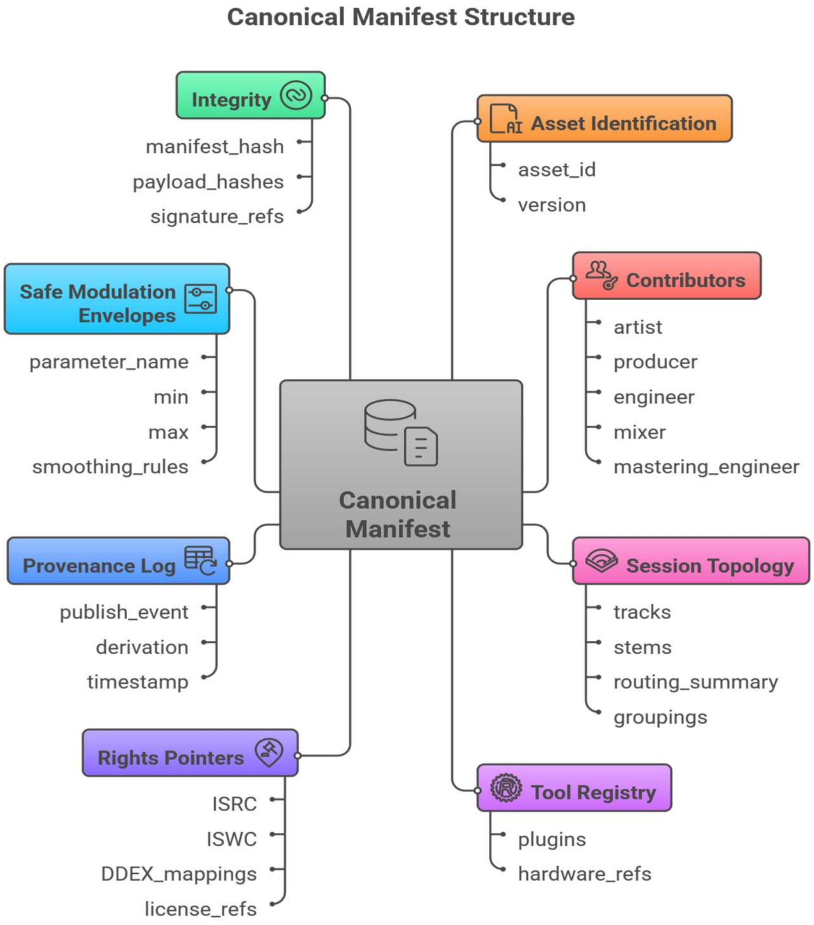
- **Contributor Attribution**
  Artists, producers, engineers, mixers, mastering engineers, and other contributors.

- **Session Topology Summary**
  Track structure, stem-mapping, routing relationships, and groupings.

- **Plugin Chain Identities**
  Plugin IDs, versions, vendors, and chain order - not binaries or proprietary DSP.

- **Rights & Usage Pointers**
  ISRC, ISWC, DDEX mappings, license references, and registry identifiers.

- **Provenance Log**
  Publish events, derivations, timestamps, and lineage.



- **Versioning**
  Asset version, manifest version, and schema version.

- **Optional Dynamic Envelopes**
  Artist-approved modulation ranges for future presentation-layer transformations.

The manifest is not a DAW session file. It is a canonicalized representation of the session environment, the structural information necessary for attribution, provenance, and long-term interpretability.

**Diagram: 3.5 Canonical Manifest Schema**

### 3.6 Integrity Layer (How the Deliverable Can Be Verified)

DAAP includes integrity information so that recipients can verify the deliverable's authenticity and integrity in a deterministic way. The integrity layer binds the manifest to the audio payloads through hashes and, when used, digital signatures. This prevents a common downstream problem: audio files get renamed, repackaged, or re-exported, and it becomes unclear whether the package still represents the approved deliverable.

In practice, integrity verification answers simple but essential questions: "Is this the same master that was delivered?" "Has anything inside the package been changed?" "Is this version the one that was approved?" DAAP's integrity layer is not about restricting playback; it is about preserving traceability and preventing silent mutation of a deliverable that is supposed to be definitive.

The integrity layer ensures that a DAAP asset can be authenticated, validated, and trusted across its entire lifecycle. It includes:

- **Component Hashes**
  Cryptographic hashes for each audio payload and the manifest.

- **Optional Digital Signatures**
  Publisher signatures, contributor signatures, or registry signatures.

- **Verification Metadata**
  Information required to validate the asset independently.

The integrity layer does not enforce DRM. It provides tamper detection, not access control. Its purpose is to ensure that the asset remains authentic and unaltered, regardless of where it travels.

### 3.7 Studio Workflow Walkthrough (What DAAP Looks Like in Practice)

A DAAP export fits cleanly into a familiar workflow. After a mix or master is approved, the engineer prints the deliverables exactly as required: the stereo master and any requested alternates or stems. At export time, a DAAP-aware tool generates a manifest that records the deliverable as an asset: it assigns an asset identifier, records the version label, and enumerates the printed payloads as referenced components.

Next, the export tool records the contextual information that would normally be scattered across recall notes: contributor roles and credits at the level the studio chooses to disclose, tool references and versions as identifiers, and a synopsis of the deliverable's relationship to the approved production state. The tool then computes hashes over each payload and over the canonical manifest representation and, where configured, applies a digital signature representing the delivery authority (e.g., studio, label, authorized release entity). If the workflow includes timestamped acceptance or release events, those events are recorded as part of the manifest lifecycle.

The resulting output is a single DAAP asset package. From the studio's perspective, it replaces "master.wav +

stems folder + notes + screenshots + email thread" with a deliverable that remains coherent as it travels. From the recipient's perspective, it provides two capabilities: the audio is immediately usable for playback and distribution, and the asset can be verified and interpreted without reopening a DAW session.

This is the key studio distinction. DAAP preserves the integrity and meaning of the approved deliverable, while still allowing the industry to distribute audio through existing channels.

### 3.8 Clarification: "Print-Down / Freeze" Does Not Mean Shipping Plugins

DAAP may describe audio payloads as the result of a controlled print-down process. In this document, "print-down" means that plugin processing and automation are resolved into the rendered audio outputs you already deliver. DAAP does not require shipping plugin binaries, licensing files, or executable session environments. Tool references in the manifest are identifiers and versions used for attribution and interpretability, not dependencies required to play the delivered audio.

Put simply, the audio payloads are the "frozen result," while the manifest captures the "who/what/version" context that would otherwise disappear at export.

### 3.9 How DAAP Preserves the Session Environment

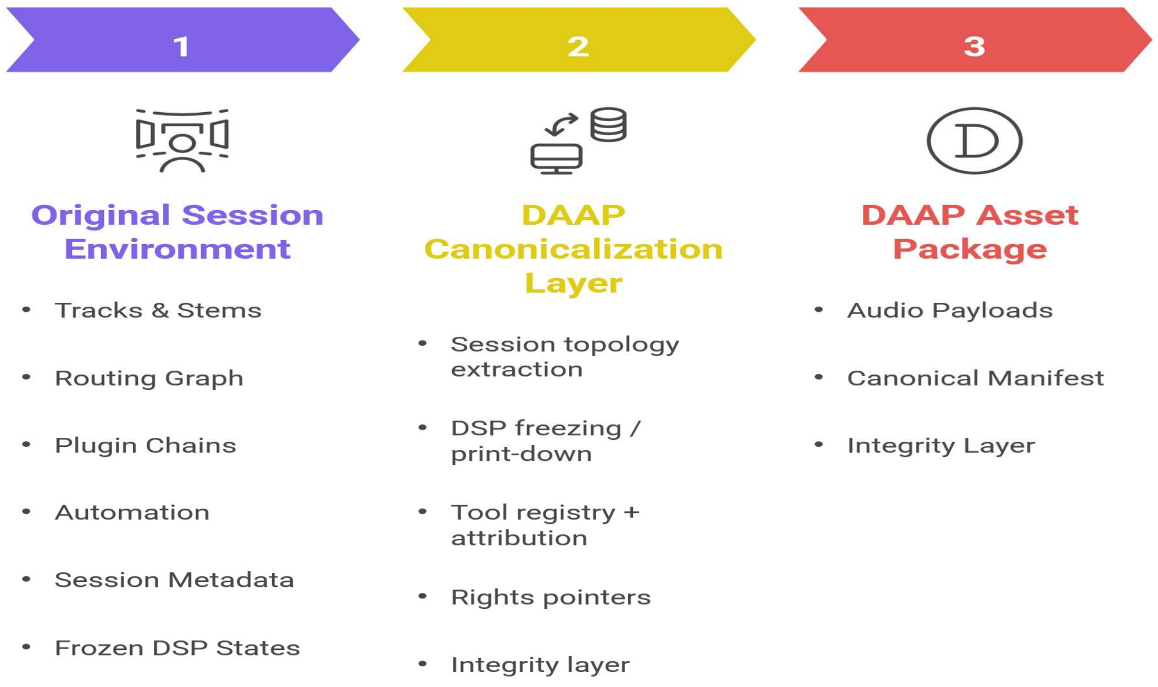
DAAP preserves the session environment not by embedding the DAW project file or plugin binaries, but by capturing the canonicalized structure of the session, as a portable, machine-readable synopsis that remains meaningful after audio is rendered. This allows future DAAP-aware systems to reconstruct the approved, distributable representation of the session, even if the original tools no longer exist.

DAAP also preserves version lineage. When an alternate mix, stem pack, clean edit, or later derivative is created, the asset can record that relationship explicitly, so downstream parties can distinguish between similar-sounding renders that are structurally different or contractually distinct. This reduces reliance on folder conventions and memory to determine what is "the approved version," what changed, and what a derivative was derived from.

In short, DAAP preserves the session environment by preserving the session's *meaningful structure* - who did what, what was delivered, how it relates, and what version it represents while keeping the deliverable lightweight, interoperable, and suitable for professional distribution and archival practice.

The full technical specification for canonicalization, integrity, and verification behavior is defined in the AES/IEEE-oriented version of this work this section describes the practical mechanics at a studio-professional level

## DAAP Session Preservation



1

**Original Session Environment**

- Tracks & Stems
- Routing Graph
- Plugin Chains
- Automation
- Session Metadata
- Frozen DSP States

2

**DAAP Canonicalization Layer**

- Session topology extraction
- DSP freezing / print-down
- Tool registry + attribution
- Rights pointers
- Integrity layer

3

**DAAP Asset Package**

- Audio Payloads
- Canonical Manifest
- Integrity Layer

### 3.10 DAAP vs DAW Session

To clarify the distinction between creation and distribution, DAAP defines a distributable environment, not an editable one.

DAAP does not replace the DAW.
It replaces the **flat file** as the unit of distribution.

Modern music production environments are extraordinarily complex. A DAW session contains editable tracks, live plugin chains, automation curves, routing graphs, hardware dependencies, and proprietary project structures. These elements form the creative environment in which music is made, but they are not suitable for distribution, preservation, or long-term interpretability.

DAAP does not attempt to replicate or replace the DAW. Instead, it defines a distributable representation of the session, one that captures the structural identity of the work without requiring the full editing environment. This distinction is essential: DAAP is not an authoring format; it is a publishing format.

A DAW session is inherently:

- **editable**
- **proprietary**
- **tool-dependent**
- **non-portable**

- **fragile across time**
- **unsuitable for distribution**

A DAAP asset is intentionally:

- **canonicalized**
- **portable**
- **DAW-agnostic**
- **plugin-agnostic**
- **durable across time**
- **suitable for distribution and archival**

DAAP preserves the *structure* of the session, not the *mechanics* of the DAW. It captures the topology, identities, and frozen DSP outputs that define the work, while excluding proprietary elements such as plugin binaries, presets, or DAW-specific editing constructs.

This approach ensures that future DAAP-aware systems can reconstruct the approved, distributable representation of the session without requiring the original tools. It also ensures that the asset remains interpretable decades into the future, even if the DAW, plugins, or hardware used to create it no longer exist.

The distinction between the editable environment and the distributable environment is illustrated below.

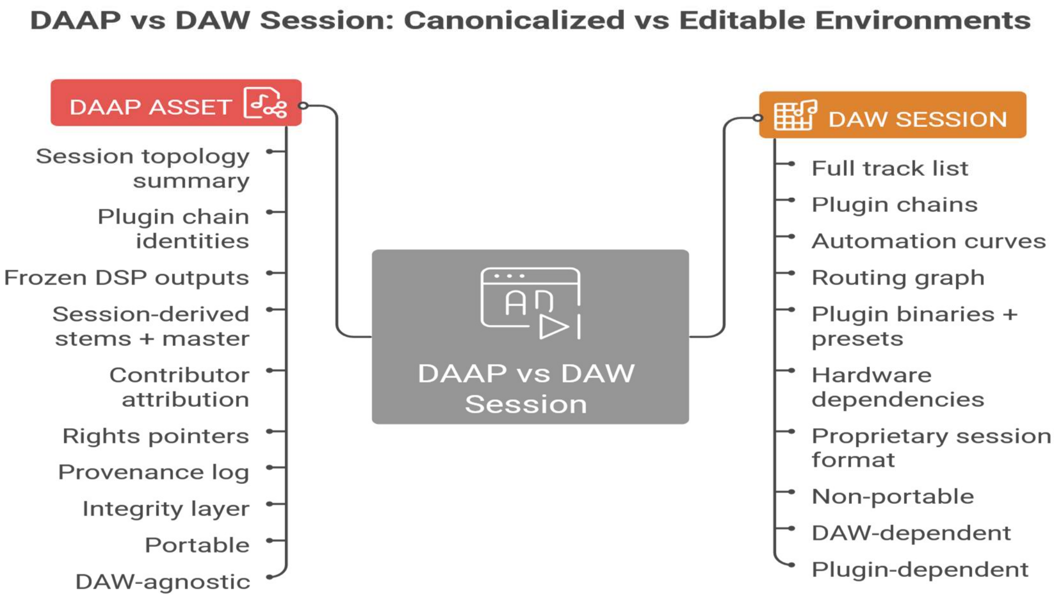


Illustration 3.10

### 3.11 What DAAP Enables for Studio Work (Immediate Value)

For studio professionals, DAAP's value is immediate and operational. It provides a consistent deliverable that retains credits and role attribution beyond the DAW. It preserves approved version identity so that alternates and later derivatives can be tied back to the correct source. It reduces recall friction because the deliverable carries structured provenance rather than relying on informal records. It also allows studios and labels to produce deliverables that can be verified later, which is increasingly important for reissues, catalog management, archival integrity, and downstream derivative packages.

DAAP is therefore best understood as a studio-grade deliverable format that preserves session-derived meaning after export, rather than as a replacement for existing audio formats or playback systems.

### 3.12 Why This Matters for Studio Delivery and Archive Practice

Studios are routinely asked to produce additional deliverables long after a release: instrumentals, clean edits, stems, spatial mixes, format-specific masters, or licensed derivatives. When the only authoritative artifact is a flat file and scattered notes, these requests create friction and risk.

DAAP reduces that risk by ensuring the approved deliverable carries structured identity and provenance that survives hand-offs. In practical terms, DAAP helps studios deliver work that remains intelligible and attributable even when it leaves the environment that produced it.

For studio professionals, this means the deliverable can carry more than the waveform. It can carry the attribution, structural relationships, and version lineage that define the approved production state.

The listening experience for the audience does not change. The difference is that the production context that created the record no longer disappears when the audio leaves the studio environment.

## SECTION 4: WHY DAAP MUST BE A PROTOCOL (STUDIO INTEROPERABILITY)

DAAP must be a protocol because the studio deliverable does not live in one tool, one vendor ecosystem, or one point in time. A finished record routinely moves across DAWs, facilities, labels, distributors, archives, and downstream partners. Even when the audio itself remains intact, the surrounding context, credits, version identity, stem lineage, and recall-relevant notes, tends to fragment as it crosses those boundaries. A protocol is the only practical way to ensure that a deliverable carries consistent meaning when it leaves the originating session environment.

In studio practice, “interoperability” does not mean that every downstream system can reopen a DAW session. It means that a deliverable can be handed off and later understood without relying on informal conventions. Today, a great deal of the industry’s continuity depends on non-standard habits: file naming conventions, folder structures, screenshots of plugin chains, and email threads describing approvals. Those methods work until they don’t, especially when multiple versions exist, when stems are requested years later, or when work is transferred between teams that were not present at the time of production.

DAAP addresses this by standardizing what it means to deliver a session-derived work as an asset. The protocol defines a consistent asset package composed of audio payloads, a canonical manifest, and integrity data. The manifest expresses the deliverable in a structured way: which audio payloads are included, what roles they serve (master, alternate, stem), what version state they represent, and what attribution and tool references are associated with that version. Integrity data binds the manifest to the payloads so that downstream parties can verify they are receiving the approved deliverable and not a silently altered variant.

### 4.1 Why DAAP Must Preserve the Session Environment

This protocol boundary is important because studios cannot assume downstream homogeneity. A mix may begin in one DAW, receive editorial work in another, be mastered in a third environment, and then move through distribution channels that normalize and repackage audio. Without a protocol, each handoff becomes a translation problem and a trust problem. With a protocol, studios can deliver a standardized object whose identity remains stable across those handoffs, even when the audio is repackaged for different workflows.

DAAP preserves the session environment because:

- attribution depends on tool usage
- provenance depends on lineage
- valuation depends on contribution modeling
- AI interpretability depends on structural context
- archival stability depends on topology
- Future playback depends on canonicalized structure

A flat file cannot support these requirements.

### 4.2 DAAP as the Successor to the Flat File

DAAP also allows studios to control how much context they disclose without losing verifiability. Not every deliverable needs to expose sensitive production details. DAAP supports the ability to preserve attribution and lineage while withholding private elements when needed. The essential point is that the deliverable can remain coherent and verifiable even when disclosure is limited.

In short, DAAP is a protocol and not just a flat audio file because studio deliverables require a consistent definition that survives tool changes, personnel changes, and time. It turns “what we meant by this deliverable” from informal documentation into a standardized, machine-readable, verifiable asset state that can be carried forward without re-creating the entire session environment.

### 4.3 Tool Attribution in Modern Production

Modern music production depends heavily on digital signal-processing tools, including software instruments, effects processors, sampling engines, and hybrid hardware integrations. These tools contribute directly to the sonic characteristics of a finished recording.

Within the production environment, the role of these tools is explicit. Plugin chains, instrument engines, and signal-processing decisions are visible as part of the session topology. Once the project is exported to a flat audio file, however, this information typically disappears.

The resulting distribution artifact preserves the audible output but not the production environment that generated it. As a consequence, the contribution of tools, processors, and instrument engines cannot participate in upstream revenue once the recording leaves the studio.

DAAP does not attempt to distribute plugin software or replicate the original processing environment without strict security. Instead, it records tool identities and versions as references within the canonical manifest. These references provide contextual attribution without exposing proprietary DSP algorithms.

### 4.4 Implications for production attribution

Digital production workflows introduce additional contributors whose impact is embedded in the sound of the recording itself.  Despite this, the digital distribution system does not preserve any reference to the tools that participated in the creation of the recording. Once a session is rendered into a flat audio file, the signal chain collapses into a waveform and the production environment becomes invisible outside the studio.

Current distribution systems, however, recognize only the final audio file and a limited set of metadata identifiers. Because the production environment does not travel with the asset, the technical contributors to the sound of the recording remain disconnected from the distributed object.

DAAP allows production tools to be referenced as part of the structural record of the asset. These references function similarly to contributor credits, identifying the tools that shaped the recording without exposing proprietary processing algorithms or requiring the playback system to recreate the session.

By preserving a structured representation of the production environment, DAAP creates the technical conditions under which these relationships can remain visible after distribution. This information allows the production environment of a recording to remain intelligible after distribution and establishes the technical foundation required for future attribution frameworks that may extend beyond traditional performer and songwriter credits.

The protocol does not prescribe any specific economic model for this information. Instead, it ensures that the structural record of the production process can persist alongside the audio asset should future attribution or licensing frameworks choose to reference it.

## SECTION 5: INTEROPERABILITY & STANDARDS ALIGNMENT

The Digital Audio Assets Protocol is designed to strengthen, not disrupt, the existing music-industry standards ecosystem. DAAP does not replace DDEX, ISRC, ISWC, EBU, C2PA metadata frameworks, or platform-specific ingestion schemas. Instead, it provides the structural substrate that these systems have always lacked: a durable, asset-bound representation of the production environment.

Today’s metadata and rights systems operate out-of-band relative to the audio file. They rely on external databases, ingestion pipelines, and reconciliation processes to maintain identity and attribution. This approach works only as long as the audio file remains stable and the metadata remains intact, conditions that are increasingly difficult to guarantee in a world of dynamic rendering, AI-mediated workflows, and large-scale catalog operations.

DAAP addresses this fragility by binding identity, structure, and provenance directly to the asset. It ensures that the information required for attribution, rights management, and long-term interpretability travels *with* the work, not separately from it. This makes DAAP a natural complement to existing standards, providing the missing layer of structural integrity that enables those standards to function more reliably.

### 5.1 Interoperability: DAAP as a Studio Deliverable That Still Ships Normally

DAAP is designed to be adopted in studio and mastering workflows without forcing downstream systems to change how they ingest or play music. A DAAP deliverable can coexist with standard distribution pipelines because it does not require new audio codecs or new playback behavior. It provides a stronger deliverable object for studios and labels while allowing the same audio payloads to be extracted and delivered through existing channels.

In practical terms, interoperability in DAAP means two things. First, the DAAP asset contains audio payloads that remain compatible with conventional mastering and delivery requirements. Second, the manifest can export or map key identifiers and credits fields into the interfaces the industry already uses, while still preserving the complete DAAP context inside the asset package

### 5.2 Mapping to Today’s Delivery Interfaces (Without Becoming Them)

Studios and labels already operate within established delivery interfaces and identifier regimes. DAAP is designed to map to those systems rather than compete with them. Where a label or distributor requires identifiers such as ISRC and UPC, DAAP can carry those identifiers in the manifest and expose them during export. Where delivery systems use structured message formats, DAAP can export compatible data fields without requiring those systems to interpret the full DAAP manifest.

DAAP's position is simple: existing delivery standards remain the transport layer for distribution, while DAAP remains the authoritative asset layer for the studio deliverable. If downstream systems truncate or ignore extended attribution fields, DAAP still preserves the full context inside the asset.

### 5.3 Alignment with DDEX

DDEX standards such as RIN (Recording Information Notification) and ERN (Electronic Release Notification) define how metadata is exchanged between labels, distributors, and platforms. They do not define how that metadata is preserved inside the asset itself.
DAAP fills this gap by:

- capturing contributor roles and tool usage in the canonical manifest
- providing stable identifiers that can be mapped into DDEX fields
- ensuring that attribution survives beyond the ingestion pipeline
- enabling DDEX-compliant systems to validate the asset's provenance

DAAP does not replace DDEX. It strengthens it by ensuring that the metadata DDEX expects is structurally tied to the asset.

### 5.4 Alignment with ISRC and ISWC

ISRC and ISWC identifiers remain essential for recording and composition identity. DAAP does not alter or replace these identifiers. Instead, it:

- embeds ISRC and ISWC references directly into the manifest
- ensures that these identifiers remain attached to the asset
- provides a structural context that enhances their interpretability
- supports future registry systems that may require asset-bound identity

DAAP ensures that ISRC and ISWC are not just external labels, they become part of the asset's internal identity.

### 5.5 Alignment with EBU Metadata Standards

EBU Tech 3293 and related metadata frameworks define broadcast-oriented production metadata. DAAP complements these standards by:

- providing a canonicalized session topology
- capturing plugin chain identities and routing summaries
- preserving provenance in a machine-readable form
- enabling broadcasters to interpret the asset without external reconstruction

DAAP does not duplicate EBU metadata. It provides the structural foundation that EBU metadata can reference.

### 5.6 Relationship to C2PA / Content Credentials

DAAP is not competing provenance standard. DAAP is an audio-production–

native *asset definition* whose canonical manifest and integrity layer provide durable, machine-verifiable structure for studio works. C2PA (Content Credentials) is a cross-media provenance framework designed to communicate origin and edit history across heterogeneous platforms and consumers.

DAAP aligns with C2PA by exposing a stable, cryptographically bound identity surface specifically the manifest hash and signer attestations that can be carried into C2PA claims as an external reference or embedded assertion, enabling DAAP assets to participate in the broader provenance web without collapsing DAAP's studio-grade attribution model into lossy consumer tags. C2PA solves cross-platform provenance signaling; DAAP solves studio topology preservation. (Different layers, different failure modes.

### 5.7 Identifiers and Credits: What DAAP Preserves Versus What Platforms Display

Many downstream platforms display only a subset of the credits and contextual information that exists upstream. This is not unusual; it is a consequence of platform interface choices, schema limits, and catalog normalization. DAAP is built to tolerate that reality. DAAP preserves contributor roles, tool references (where disclosed), lineage markers, and approval events as part of the deliverable, even if a downstream platform ultimately publishes only title, artist, and label.

For studio professionals, this is a key point: DAAP is not dependent on platform UX support for its value. DAAP is a studio-grade archive and deliverable format. The information you preserve at export remains attached to the asset whether or not a downstream platform chooses to display it.

### 5.8 Alignment with Platform-Specific Ingestion Schemas

Every major DSP (Spotify, Apple Music, Amazon, YouTube, etc.) uses its own ingestion schema. These schemas are not standardized, and they often overwrite or discard metadata during processing.

DAAP resolves this by:

- preserving attribution and structure inside the asset
- enabling ingestion systems to extract canonical metadata
- reducing reliance on fragile external metadata feeds
- ensuring that identity survives transcoding and repackaging

DAAP is not a replacement for ingestion schemas. It is a source of truth that ingestion schemas can rely on.

### 5.9 Interoperability for Reissues, Derivatives, and Long-Tail Requests

The most important benefit of interoperability for studios is not initial distribution. It is long-tail operational continuity. DAAP is built to support real-world scenarios such as:

- a label requesting stems years later,
- producing a clean edit or instrumental for licensing,
- remastering a catalog title with clear version lineage,
- delivering alternate mixes for different territories,
- preparing derivatives without losing attribution context.

In these situations, DAAP reduces dependency on informal records and provides a deliverable that can be validated and interpreted even when original session tooling has changed.

### 5.10 DAAP as a Compatibility Layer

DAAP is intentionally designed to be:

- backward-compatible with existing metadata systems
- forward-compatible with future session-aware assets
- agnostic to DAW, plugin, and distributor implementations
- **neutral** with respect to business models and licensing frameworks

DAAP does not require industry to abandon existing standards. It provides a structural foundation that allows those standards to function more reliably and evolve more gracefully.

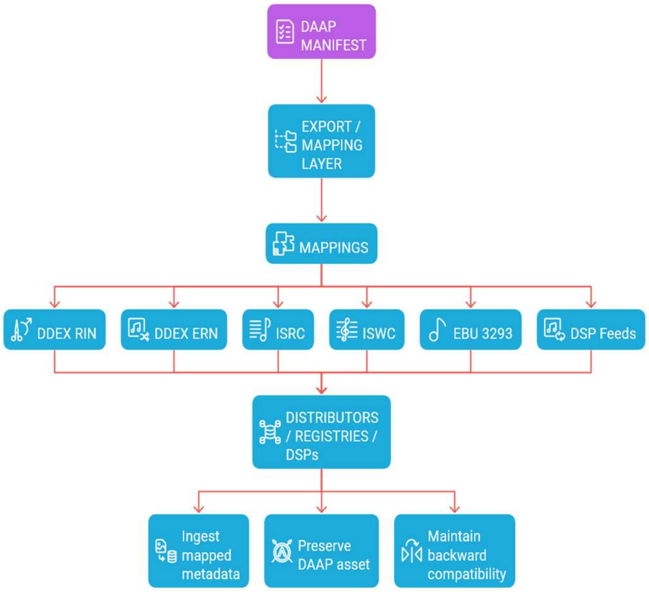
### 5.11 Why Interoperability Matters

Interoperability is not a convenience; it is a requirement for long-term industry health. Without it:

- attribution remains inconsistent
- Rights management remains fragile
- provenance remains unverifiable
- archives remain incomplete
- AI systems remain ungrounded
- catalog operations remain error-prone

DAAP ensures that the structural identity of a musical work is preserved across every system it touches, enabling a more transparent, accountable, and durable music ecosystem.

DAAP Interoperability Flowchart



### 5.12 What Interoperability Does Not Mean

Interoperability in DAAP does not mean that DAAP assets must be playable in every consumer application. DAAP also does not mean that a DAAP asset allows downstream recipients to reopen or edit the session. DAAP preserves the approved deliverable state and its provenance. Playback remains possible through conventional audio payloads, while full DAAP-aware interpretation is available to tools that choose to implement the protocol.

The intent is to give studios a deliverable that remains coherent across systems without requiring every system to adopt DAAP simultaneously.

### 5.13 PROs: Where DAAP Connects and Where It Does Not

Performing Rights Organizations (PROs) and collective management organizations (e.g., ASCAP, BMI, SESAC, PRS, GEMA, SACEM) sit downstream of the studio deliverable, but they are directly affected by metadata discontinuity.

It is important to distinguish recording metadata from publishing metadata. Studio deliverables are primarily concerned with the sound recording, the mastered audio and the production context that produced it. Publishing and PRO/CMO systems are primarily concerned with the musical work, songwriter identities, publisher shares, and performance-related licensing. These two layers are related but not identical. In practice, a single work (ISWC) can correspond to multiple recordings (ISRCs), and a single recording may involve multiple parties beyond the composition layer (performers, producers, engineers, tool vendors, etc.).

When the identifiers and contributor context connecting these layers are incomplete or inconsistent, downstream systems must reconcile identity across registries, distributor catalog IDs, DSP track IDs, and rights databases. DAAP does not attempt to redefine publishing registration or splits, but it supports more reliable linkage by preserving a stable recording-layer identity and provenance object that can be referenced alongside existing publishing registrations. In other words, it improves upstream identity clarity by preserving recording-level provenance and contributor context in a deliverable form that can be referenced more reliably downstream.

## SECTION 6: GOVERNANCE & CONSORTIUM ALIGNMENT

The Digital Audio Assets Protocol is intentionally designed as a neutral, open, and extensible standard. Its long-term viability depends on governance that is transparent, practitioner-driven, and independent of any single company, platform, or commercial interest.

For this reason, DAAP is stewarded by the Digital Audio Assets & Production Attribution Consortium, a multi-stakeholder body composed of educators, engineers, developers, archivists, researchers, and industry professionals.

The consortium does not control the creative process, the economics of music, or the business models of distributors. Instead, it governs the standards layer, the schemas, definitions, versioning rules, and compliance requirements that ensure DAAP remains stable, interoperable, and aligned with the needs of the people who build and use modern production systems.

This separation of concerns is essential. DAAP defines the systems substrate: how assets are structured, how identity is preserved, how provenance is recorded, and how integrity is verified.

The consortium defines the governance substrate: how schemas evolve, how attribution models are standardized, how ethical guidelines are maintained, and how compliance is validated.

Together, they form a layered architecture in which the protocol provides the systems foundation and the consortium provides the institutional stability required for long-term adoption.

---

### 6.1 The Role of the Consortium

The consortium is responsible for:

- **Schema Versioning**
  Maintaining and evolving the canonical manifest schema, ensuring backward compatibility and forward extensibility.
- **Attribution Standards**
  Defining contributor roles, tool categories, and attribution models that reflect real-world production practices.
- **Valuation Frameworks (Non-Prescriptive)**
  Establishing optional, non-binding frameworks that enable evidence-based valuation without dictating economic outcomes.
- **Ethical Guidelines**
  Ensuring that DAAP supports transparency, fairness, and responsible use of production data, especially in AI-mediated workflows.
- **Compliance Testing**
  Providing reference implementations, validation tools, and certification processes to ensure consistent adoption.
- **Education & Outreach**
  Supporting educators, students, and practitioners in understanding and using DAAP effectively.

The consortium does **not** control pricing, licensing, royalties, or business models. These remain the responsibility of rights holders, distributors, and market participants. The consortium’s role is to ensure that the information infrastructure underlying those decisions is accurate, durable, and trustworthy.

---

### 6.2 Why Governance Matters

Without governance, technical standards drift, fragment, or become captured by dominant platforms. DAAP is designed to avoid these outcomes by ensuring that:

- no single company controls the protocol
- no platform can unilaterally redefine attribution
- no vendor can lock the ecosystem into proprietary extensions
- no distributor can dictate the structure of the asset
- no tool can bypass the integrity layer

Governance ensures that DAAP remains:

- **neutral**
- **transparent**
- **practitioner-driven**
- **academically grounded**
- **industry-aligned**
- **future-proof**

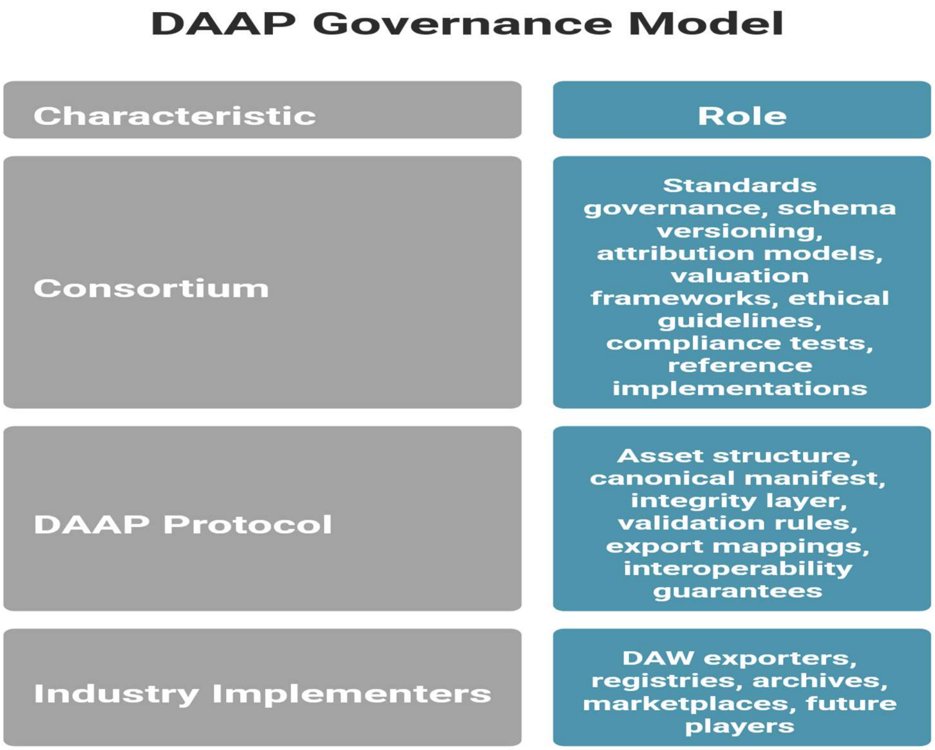
This is essential for building trust across the ecosystem.

---

### 6.3 The Consortium as a Long-Term Steward

The consortium's role is not to dictate creative practice or economic policy. Its role is to ensure that the information infrastructure of music, attribution, provenance, structure, identity, remains durable, interpretable, and aligned with the needs of creators, educators, technologists, and archivists.

DAAP provides the systems foundation. The consortium provides the institutional foundation.



DAAP Governance Model

| Characteristic | Role |
|---|---|
| Consortium | Standards governance, schema versioning, attribution models, valuation frameworks, ethical guidelines, compliance tests, reference implementations |
| DAAP Protocol | Asset structure, canonical manifest, integrity layer, validation rules, export mappings, interoperability guarantees |
| Industry Implementers | DAW exporters, registries, archives, marketplaces, future players |

Together, they enable a healthier, more transparent, and more resilient music digital distribution ecosystem.

## SECTION 7: CONCLUSION

The Digital Audio Assets Protocol represents a structural shift in how the music industry defines, preserves, and distributes recorded works. For decades, the industry has relied on flat audio files as the canonical representation of music, formats that faithfully preserve sound but discard the production systems, contributor identities, and contextual information that define how modern music is created. This model was sufficient for an earlier era, but it is fundamentally incompatible with the realities of contemporary production and the demands of future-facing music ecosystems.

DAAP addresses this gap by introducing a session-derived, structurally complete digital asset that binds audio payloads to a canonical manifest and an integrity layer. This asset preserves the topology of the session, the identities of the tools, the roles of contributors, the lineage of the work, and the metadata required for long-term interpretability. It ensures that identity and provenance remain attached to the asset itself, not scattered across external databases or lost during distribution.

DAAP does not attempt to recreate the DAW or expose proprietary DSP algorithms. Instead, it captures the approved, distributable representation of the session, the structural information necessary for attribution, provenance, and future interpretability, in a durable, interoperable form. This approach enables DAAP-aware systems to reconstruct the essential context of the work without requiring the original editing environment, plugins, or hardware.

The protocol is intentionally neutral, open, and extensible. It is designed to complement existing standards such as DDEX, ISRC, ISWC, EBU metadata frameworks and C2PA provenance standards, not replace them. DAAP provides the structural foundation that allows these standards to function more reliably and evolve more gracefully. It strengthens the entire ecosystem by ensuring that the information required for attribution, rights management, and archival preservation is embedded directly within the asset.

The consortium provides the governance necessary to steward this protocol responsibly. By maintaining schema versioning, defining attribution models, establishing ethical guidelines, and supporting compliance testing, the consortium ensures that DAAP remains aligned with the needs of creators, educators, technologists, archivists, and industry professionals. This governance structure protects the protocol from fragmentation, platform capture, and commercial distortion, ensuring that DAAP remains a stable and trustworthy foundation for decades to come.

As the industry moves toward session-aware, dynamically interpretable assets, and as AI-mediated production becomes increasingly integrated into creative workflows, the need for durable, asset-bound identity becomes urgent.

DAAP provides the systems substrate required for transparent attribution, verifiable provenance, and future-proof distribution. It enables a healthier, more accountable, and more resilient music information infrastructure.

DAAP is not merely a new format. It is a new definition of what a musical work *is* in the digital era: not a flattened waveform, but a structured, self-describing, self-verifying digital asset that preserves the creative and technical systems from which it emerged.

DAAP is the foundation.
The consortium is the steward.
Together, they define **the future of music information technology.**

# References

Apple Inc. (2023). *AAC encoding guidelines*. Apple Developer Documentation.

Audio Engineering Society (AES**).** (2019). *AES-X098B: Best Practices for Audio Metadata (Working Group Draft)*. https://aes2.org/standards/

Billboard Media. (2019). *Why music credits are still broken in the streaming age*. Billboard.

Bradner, S. (1997). *Key words for use in RFCs to indicate requirement levels* (RFC 2119). Internet Engineering Task Force.

Coalition for Content Provenance and Authenticity (C2PA). (2024). *C2PA Technical Specification, Version 2.2*. https://spec.c2pa.org/specifications/specifications/2.2/specs/C2PA_Specification.html.

Content Authenticity Initiative. (n.d.). *Content Authenticity Initiative*. https://contentauthenticity.org/

Content Authenticity Initiative. (n.d.). *Secure mode enabled*. https://contentauthenticity.org/secure-mode-enabled

Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., & Polk, W. (2008). *Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile* (RFC 5280). Internet Engineering Task Force.

DDEX. (2022). *DDEX standards for recording, release, and rights information*.  Digital Data Exchange, LLC.

England, P., Malvar, H. S., Horvitz, E., Stokes, J. W., Fournet, C., Burke-Aguero, R., & Zaman, A. (2020). *AMP: Authentication of media via provenance* (arXiv:2001.07886). arXiv. https://arxiv.org/abs/2001.07886

European Broadcasting Union (EBU)**.** (2011). *EBU Tech 3285: Specification of the Broadcast Wave Format (BWF)*. https://tech.ebu.ch/docs/tech/tech3285.pdf

European Commission. (2019). *Directive (EU) 2019/790 on copyright and related rights in the Digital Single Market and amending Directives 96/9/EC and 2001/29/EC*. Official Journal of the European Union.

FADGI**.** (2021). *Guidelines: Embedded Metadata in Broadcast WAVE Files (BWF), Version 3*. https://www.digitizationguidelines.gov/audio-visual/documents/BWF_Embed_Guideline_v3_2021.pdf

Huang, C.-Z. A., Hawthorne, C., Roberts, A., Dinculescu, M., Wexler, J., Hong, L., & Eck, D. (2023). *Music transformer: Generating music with long-term structure*. Proceedings of the International Conference on Learning Representations (ICLR).

International Federation of the Phonographic Industry. (2023). *Global music report 2023*. IFPI.

International Organization for Standardization. (n.d.). *About ISO*. https://www.iso.org/about

ISO/IEC. (2015). *ISO/IEC 14496-12: Information technology: Coding of audio-visual objects - Part 12: ISO base media file format*. International Organization for Standardization.

Ivors Academy. (2021). *Data issues are at the heart of half a billion pounds a year of unallocated or misallocated streaming royalties for songwriters and rightsholders*. The Ivors Academy.

Joint Development Foundation Projects, LLC. (n.d.). *About*. https://jointdevelopment.org/abou

*Journal of New Music Research, 48*(1), 36–55. https://doi.org/10.1080/09298215.2018.1515232

Kahn, R., & Wilensky, R. (2006). *A framework for distributed digital object services*. Corporation for National Research Initiatives.

Library of Congress. (2023). *Sustainability of digital formats: Broadcast Wave Format (BWF)*. Federal Agencies Digital Guidelines Initiative. https://www.loc.gov/preservation/digital/formats/fdd/fdd000356.shtml

Lynch, C. (1997). *Identifiers and their role in networked information applications*. Association for Computing Machinery.

Music Business Worldwide. (2020). *Metadata: The music industry's invisible infrastructure problem*. Music Business Worldwide.

Music Business Worldwide. (2025, November 27). *Metadata is the industry's weak spot in the age of AI:  the cost of half measures is far higher*. Music Business Worldwide.

OpenMusE. (2025). *Music metadata mainstreaming and EU law*. Policy Brief.

Recording Academy. (2021). *Credits due: Rebuilding music's crediting system*. The Recording Academy.

Rescorla, E. (2018). *The transport layer security (TLS) protocol version 1.3* (RFC 8446). Internet Engineering Task Force.

Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., & Adams, C. (2013). *X.509 internet public key infrastructure online certificate status protocol: OCSP* (RFC 6960). Internet Engineering Task Force.

Seaman, G. (2020). *JSON canonicalization scheme (JCS)* (RFC 8785). Internet Engineering Task Force.

Sturm, B. L., Ben-Tal, O., Monaghan, Ú., Collins, N., Herremans, D., Chew, E., & Coeckelbergh, M. (2019). *Machine learning research that matters for music creation*.

Thompson, M., & Polk, W. (2014). *Internet X.509 public key infrastructure time-stamp protocol (TSP)* (RFC 3161). Internet Engineering Task Force.

The Verge. (2019, May 29). *Metadata is the biggest little problem plaguing the music industry*. The Verge.

Wikström, P. (2013). *The music industry: Music in the cloud* (2nd ed.). Polity Press. (Digital Media and Society Series)

Xiph.Org Foundation. (2020). *Vorbis I specification*. https://xiph.org/vorbis/doc/Vorbis_I_spec.html

# APPENDIX A

**Glossary (DAAP Terms)**

**Active Manifest -** The single DAAP manifest instance designated by deterministic selection rules as the authoritative claims object for an asset. The active manifest is the reference point for verification, attribution, provenance evaluation, and policy enforcement.

**Asset (DAAP Asset)** - A packaged, signed, and versioned digital object consisting of audio payload bindings, a canonical manifest, and an integrity layer. In DAAP, an asset denotes an object whose identity, structure, and provenance are verifiable independently of any single platform, database, or distributor.

**Asset ID (asset_id)** - The stable identifier assigned to a DAAP asset across all versions. The Asset ID identifies the asset lineage as a whole rather than a specific serialized file instance.

**Asset Version (asset_version)** - A monotonic or semantically versioned label identifying a specific manifest state within an asset lineage. Asset versions are linked through explicit parent pointers and derivation events.

**Attribution** - The structured representation of contributor roles, tool usage, and creative or technical participation in the production of a musical work. In DAAP, attribution is embedded directly in the canonical manifest.

**Audio Payloads** - The audible components of a DAAP asset, including the master mix, session-derived stems, and optional preview assets. Audio payloads represent frozen DSP outputs corresponding to the approved, distributable version of the work.

**Bundle / Container** - The physical packaging form that carries the DAAP manifest store and, optionally, the associated payloads. DAAP defines a normative container model as well as an optional external-reference bridge for payload storage.

**Canonical Manifest** - The authoritative manifest encoded using a DAAP-defined format and canonicalization scheme such that all compliant implementations produce identical byte sequences for hashing and signing. DAAP v1 specifies JSON canonicalization using RFC 8785 (JCS) as the baseline.

**Canonicalization** - The deterministic process of transforming a manifest into a stable byte representation suitable for cryptographic hashing and signing. DAAP v1 requires RFC 8785 for JSON canonicalization.

**Content Binding / Payload Binding** - A cryptographic binding between a manifest and payload bytes using hashes and, typically, digital signatures. Payloads are identified by cryptographic digests rather than filenames alone, with hash baselines defined by FIPS 180-4.

**Contributor** - Any individual or entity that played a creative, technical, or supervisory role in the production of a musical work, including artists, producers, engineers, mixers, mastering engineers, session musicians, and technical contributors.

**DAAP Manifest Store** - The authoritative location within a DAAP container that holds the active manifest, associated cryptographic objects such as signatures and timestamp tokens, and optionally historical manifests.

**DAAP-Aware System** - Any tool, registry, archive, marketplace, or playback environment capable of interpreting the structure, manifest, and integrity layer of a DAAP asset without requiring the original DAW or plugins.

**DAW (Digital Audio Workstation)** - A software environment used for recording, editing, mixing, and producing audio. ("What is a DAW? Everything you need to know. - MAGIX") DAW sessions contain editable tracks, live plugin chains, automation, routing graphs, and proprietary project structures. DAAP does not replace the DAW.

**Derivation Chain** - The ordered sequence of asset versions and/or derived assets connected by explicit parent pointers and derivation events. The derivation chain supports auditability and controlled evolution of an asset over time.

**DSP Freezing (Print-Down)** - The process of rendering plugin outputs, automation, and routing behavior into fixed audio representations. DSP freezing ensures that audio payloads reflect the approved, distributable version of the work.

**Hash Commitment** - A privacy preserving representation of a value in which only the hash of the value is disclosed. Hash commitments allow verifiers to confirm stability and continuity without revealing plaintext data.

**Integrity Layer** - The collection of hashes, digital signatures, trust anchors, revocation checks, and timestamp evidence that enables a verifier to confirm that the manifest and payload bindings have not been altered and were authorized by a trusted signer.

**Interoperability** - The ability of a DAAP asset to integrate with existing metadata and rights systems such as DDEX, ISRC, ISWC, and EBU frameworks by binding identity and structure directly to the asset.

**Manifest Hash (manifest_hash)** - The cryptographic digest of the canonicalized manifest bytes. The manifest hash serves as the primary signing input, either directly or indirectly, depending on the signature packaging model.

**Non-goals** - Capabilities that DAAP explicitly does not define, including executable DSP recreation, DAW session interchange, payout or royalty policy, and DRM access enforcement.

**Optional Bridge Profile** - A DAAP packaging mode that allows payload bytes to be referenced externally while preserving the same verification semantics, including hash bindings and signatures, as fully embedded bundles.

**PartyRef** - A structured reference to a person or organization within the manifest, such as an artist, label, or vendor, typically expressed using stable internal identifiers and optional external identifiers.

**Plugin Chain Identity** - A representation of the tools used in the production of a track or stem, including plugin identifiers, versions, vendors, and chain order. DAAP captures plugin identities but does not embed plugin binaries or proprietary DSP.

**Provenance** - The lineage of a musical work, including its creation, modification, publication, and derivation events. Provenance is recorded in the canonical manifest to ensure long-term interpretability and trust.

**Registry** A system that indexes, validates, and stores DAAP assets or their metadata. Registries may be operated by labels, archives, educational institutions, or independent organizations.

**Rights Pointers** References to external rights identifiers such as ISRC, ISWC, DDEX mappings, and license information embedded directly in the canonical manifest.

**Session Environment** - The editable production environment inside a DAW, including tracks, routing, plugin chains, automation, hardware dependencies, and proprietary project structures. DAAP preserves the structure of the session environment in canonicalized form.

**Session Topology** - The structural layout of a session, including track organization, stem mapping, routing relationships, groupings, and signal flow. DAAP captures a topology summary in the canonical manifest.

**Synopsis Graph (Production Graph Synopsis)** - A portable, abstract graph representation describing structural relationships among tracks, buses, renders, and payloads at a level sufficient for lineage tracking and intelligibility without exposing DAW-specific edit state.

**Tool Registry** - A section of the canonical manifest that records the identities of plugins, hardware references, and other tools used in the production of the work.

**Validation** - The process of verifying that a DAAP asset conforms to protocol requirements and that its integrity layer matches the referenced audio payloads and manifest.

**Versioning** - The system by which DAAP assets, manifests, and schemas track changes over time, ensuring backward compatibility and forward extensibility.

**Waveform Collapse** - The loss of structural, attributional, and contextual information that occurs when a DAW session is rendered to a flat audio file. DAAP is designed to prevent waveform collapse by preserving session structure.

**Work (Musical Work)** - The creative output represented by a DAAP asset, including its audio payloads, structural identity, contributor roles, and provenance.