HIVE Wallet Secure Media Engine (HSME) v1

Status

Version: 1.0
State: Draft (Target: LOCKED)
Layer: Application / Wallet Runtime
Depends On: Chrysalis Core, CAS-φ, Chrysalis-NFT v1, Access Controller v1

---

1. Purpose

The HIVE Wallet Secure Media Engine (HSME) defines how encrypted media NFTs (video, image, audio, 3D, documents) are **securely decrypted, reconstructed, streamed, and rendered** inside the HIVE wallet environment **without exposing raw media files** to the user, operating system, browser, or external applications.

HSME ensures: - Content creators retain 100% ownership - Media never exists in plaintext at rest - Playback is permissioned, ephemeral, and auditable - Streaming works with sharded, encrypted, decentralized storage

---

2. Threat Model

HSME is explicitly designed to resist: - File extraction from local storage - Browser dev-tools capture - Screen scraping (partial mitigation) - Memory dumping attacks - Replay of decrypted shards - Unauthorized NFT duplication

HSME does **not** attempt to defeat physical device compromise.

---

3. Supported Media Types

Category	Formats
Video	AV1, H.265 (HEVC), WebM
Image	PNG, JPEG, WebP
Audio	FLAC, AAC, OPUS
3D	GLB, USDZ
Docs	PDF (rendered only)

All media MUST be wrapped as Chrysalis-NFTs.

---

4. Architecture Overview

```
Encrypted Shards (IPFS / HoneyNet)
        ↓
Shard Fetcher
        ↓
CAS-φ Reconstructor
        ↓
Ephemeral Decryption Sandbox
        ↓
Secure Media Renderer
        ↓
GPU Frame Output (No File Access)
```

---

5. Media Lifecycle

5.1 Authorization

Playback requires: - NFT ownership OR - Time-limited Access Controller grant

The HIVE wallet verifies: - Wallet signature - Access scope - Expiry

No access = no shard fetch.

5.2 Shard Retrieval

- Shards are fetched **on demand**
- Order follows CAS-φ reconstruction sequence
- Decoy shards are intentionally fetched

Shard metadata is never exposed to UI.

5.3 Decryption Sandbox

- Runs in isolated memory
- No disk writes allowed
- Keys exist only in RAM
- Keys destroyed on pause, stop, tab blur, or wallet lock

Keys are never: - Serialized - Logged - Cached

---

6. Secure Playback Rules

Video

- Frame-by-frame decode
- No contiguous full file ever assembled
- Seeking allowed only within authorized ranges

Audio

- Stream-based buffer
- No full waveform reconstruction

Images

- Tile-based decode
- Zoom reconstructs only visible tiles

---

7. Anti-Extraction Controls

Control	Description
Memory Scrubbing	Buffers wiped every frame
GPU-Only Frames	No system framebuffer access
Obfuscated Timers	Prevent deterministic replay
Watermark (Optional)	Wallet-derived invisible watermark

---

8. Streaming Support

HSME supports: - Progressive playback - Live-like streaming of recorded content - Partial shard reconstruction

Streaming does **not** weaken encryption guarantees.

---

9. Offline Rules

- Media cannot be viewed offline

- Cached shards expire immediately
- Offline wallet mode disables HSME

---

10. Wallet UI Integration

HIVE Wallet MUST provide: - Secure media viewer - Video player - Audio player - Gallery view - Playlist support

All UI elements are HSME-controlled.

---

11. Event Hooks

HSME emits signed events: - play - pause - stop - seek - access_denied

Events may be used for: - Creator analytics - Royalty triggers - DAO metrics

---

12. Compliance

HSME complies with: - Chrysalis Core - CAS-φ - Chrysalis-NFT v1

Non-compliant wallets MUST NOT render protected media.

---

13. Explicit Non-Goals

HSME does NOT: - Provide DRM licensing to third parties - Allow raw media export - Support screen recording - Bypass creator permissions

---

14. Future Extensions

- Live stream secure rendering
- Encrypted group watch sessions
- Zero-knowledge view proofs
- Hardware enclave support

---

15. Final Principle

> **If the wallet cannot cryptographically justify playback, the media does not exist.**

This is the foundation of creator sovereignty in the Honey ecosystem.