L22: NFT Metadata and IPFS

Module C: NFTs & Digital Assets

Blockchain & Cryptocurrency Course

December 2025

Learning Objectives

By the end of this lesson, you will be able to:

- Understand the JSON metadata format standard for NFTs
- Explain how IPFS content addressing works
- Describe the role of pinning services in NFT permanence
- Compare IPFS and Arweave for decentralized storage
- Evaluate metadata permanence and availability challenges

NFT Metadata: The JSON Standard

Metadata Structure: OpenSea standard (widely adopted)

Core Fields:

- `name` – Display name of the NFT
- `description` – Human-readable description
- `image` – URI to primary visual asset (IPFS, HTTP)
- `external_url` – Link to project website or details
- `attributes` – Array of trait objects

Example Metadata JSON:

```
{
  "name": "Bored Ape #1234",
  "description": "A unique Bored Ape Yacht Club NFT",
  "image": "ipfs://QmXyZ.../1234.png",
  "attributes": [
    {"trait_type": "Background", "value": "Blue"},
    {"trait_type": "Hat", "value": "Beanie"}
  ]
}
```

Attributes: Defining Rarity

Attributes Array: Defines traits and properties

Attribute Object Structure:

- `trait_type` – Category name (e.g., "Hat", "Eyes")
- `value` – Specific value (e.g., "Beanie", "Laser Eyes")
- `display_type` (optional) – How to render (number, date, boost)

Rarity Calculation:

- Each trait has a frequency distribution in the collection
- Rarer traits (low frequency) increase NFT value
- Tools like Rarity Sniper calculate rarity scores

Example Rarity Analysis:

- Background: Blue (20% frequency) vs. Gold (2% frequency)
- Laser Eyes: 1% frequency (highly rare)

IPFS: The InterPlanetary File System

IPFS: Decentralized peer-to-peer file storage protocol

Key Concepts:

- **Content Addressing:** Files identified by cryptographic hash
- **CID (Content Identifier):** Unique hash of file content
- **Immutability:** Same content always produces same CID
- **Distributed:** Files stored across multiple nodes
- **Retrieval:** Any node with the file can serve it

Example IPFS URI:

- `ipfs://QmXyZ123abc...` (CID)
- Gateway URL: `https://ipfs.io/ipfs/QmXyZ123abc...`

Content Addressing vs. Location Addressing

Traditional Web (Location Addressing):

- URL points to server location: `https://server.com/file.jpg`
- Content can change without URL changing
- Server failure = file unavailable

IPFS (Content Addressing):

- CID derived from file content: `QmXyZ...`
- Changing content changes CID
- File available from any node that has it
- Verifiable integrity (hash mismatch = tampering detected)

Advantage for NFTs:

- Guarantees metadata/image match the CID in `tokenURI`
- No server dependency (decentralized retrieval)

IPFS URI Structure

CID Components:

- **Version:** CIDv0 (QmXyZ...) or CIDv1 (bafybeig...)
- **Multicodec:** Specifies content type (dag-pb, raw, etc.)
- **Multihash:** Hash algorithm (SHA-256) + hash value

Example:

- CIDv0: `QmYwAPJzv5CZsnA625s3Xf2nemtYgPpHdWEz79ojWnPbdG`
- CIDv1: `bafybeigdyrzt5sfp7udm7hu76uh7y26nf3efuylqabf3oclgtqy55fbzdi`

Determinism:

- Same file always produces same CID
- Different files always produce different CIDs

Pinning: Ensuring Availability

IPFS Garbage Collection: Nodes delete unpinned files to save space

Pinning: Explicitly keeping a file on IPFS nodes

Without Pinning:

- File may be removed from all nodes (lost)
- NFT metadata/images become unavailable
- Broken image links in wallets and marketplaces

Pinning Services:

- **Pinata:** Popular commercial pinning service
- **NFT.Storage:** Free pinning for NFT data (funded by Filecoin)
- **Infura IPFS:** Enterprise-grade IPFS infrastructure
- **Self-hosting:** Run your own IPFS node and pin files

Critical Issue: If all pinning stops, files may disappear from IPFS

IPFS Gateways

Gateway: HTTP bridge to access IPFS content

Public Gateways:

- `https://ipfs.io/ipfs/[CID]`
- `https://gateway.pinata.cloud/ipfs/[CID]`
- `https://cloudflare-ipfs.com/ipfs/[CID]`

Why Gateways Matter:

- Browsers do not natively support `ipfs://` protocol
- Wallets and marketplaces use gateways to display NFTs
- Gateway downtime = NFTs appear broken (even if IPFS file exists)

Centralization Risk:

- Most users rely on centralized gateways (ipfs.io, Cloudflare)
- Gateway can censor or throttle content
- Solution: Native IPFS support in browsers (IPFS Companion extension)

Arweave: Permanent Decentralized Storage

Arweave: Blockchain designed for permanent data storage

Key Differences from IPFS:

- **Permanence:** One-time payment for perpetual storage
- **Blockchain-based:** Data stored on Arweave blockchain
- **Economic model:** Storage endowment fund pays miners forever
- **No pinning needed:** Data guaranteed to persist

Arweave URI:

- `ar://[Transaction ID]`
- Gateway: `https://arweave.net/[Transaction ID]`

Use Cases:

- High-value NFTs requiring guaranteed permanence
- Historical records and archives
- Alternative to IPFS for critical metadata

IPFS vs. Arweave Comparison

Feature	IPFS	Arweave
Storage Model	Peer-to-peer distributed	Blockchain-based
Permanence	Requires pinning	Guaranteed (one-time fee)
Cost	Free + pinning service fees	Pay once (e.g., $5-10/GB)
Retrieval	Fast (popular files cached)	Slower (blockchain retrieval)
Adoption	High (majority of NFTs)	Growing (premium projects)
Censorship Risk	Gateway-dependent	Minimal (blockchain-based)
Data Mutability	Immutable (new CID if changed)	Immutable (on-chain)

Recommendation:

- IPFS: Most NFT projects (cost-effective, fast)
- Arweave: Premium/historical NFTs (guaranteed permanence)

Metadata Permanence Challenges

Real-World Issues:

1. **Unpinned IPFS files:** Project abandons pinning, files lost
2. **Centralized metadata servers:** Company shuts down, NFTs break
3. **Mutable tokenURI:** Smart contract allows owner to change metadata
4. **Gateway failures:** IPFS gateways go offline, NFTs appear broken
5. **Link rot:** HTTP URLs stop working (404 errors)

Case Study: Nifty Gateway (2021):

- Platform used centralized servers for metadata
- Outage caused all NFTs to display broken images
- Community backlash led to IPFS migration

Mutable vs. Immutable Metadata

Mutable Metadata:

- Smart contract owner can update `tokenURI`
- Allows bug fixes and metadata improvements
- Risk: Owner could change artwork or traits (rug pull)

Immutable Metadata:

- `tokenURI` frozen after minting (contract locked)
- Guarantees metadata cannot be altered
- Standard for high-value collections (e.g., CryptoPunks)

Verification:

- Check smart contract code for `setTokenURI()` functions
- Verify contract ownership is renounced (no admin control)
- Use Etherscan to audit contract mutability

Best Practices for NFT Metadata

For NFT Projects:

1. **Use IPFS or Arweave:** Avoid centralized servers
2. **Pin all files:** Use reputable pinning services (Pinata, NFT.Storage)
3. **Freeze metadata:** Make `tokenURI` immutable after reveal
4. **Redundancy:** Pin to multiple services (IPFS + Arweave)
5. **Document storage:** Inform buyers where metadata is hosted
6. **Community pinning:** Encourage holders to pin collection files

For NFT Buyers:

- Verify metadata is on IPFS/Arweave (not HTTP)
- Check if `tokenURI` is immutable
- Confirm pinning service reputation

Decentralized Storage Adoption

Current NFT Storage Breakdown (Estimated):

- **IPFS:** 60-70% of NFT projects
- **Centralized servers:** 20-25% (HTTP URLs)
- **Arweave:** 5-10% (growing)
- **Fully on-chain:** ¡5% (rare, expensive)

Trends:

- Migration from centralized to IPFS accelerating
- Premium projects adopting Arweave for permanence guarantees
- Layer 2 solutions (Polygon) enabling cheaper on-chain storage

NFT.Storage: Free IPFS Pinning

NFT.Storage: Protocol Labs initiative (Filecoin Foundation)

Service Features:

- Free IPFS pinning for NFT data
- Automatic backup to Filecoin network
- Simple API for developers
- No usage limits or expiration

How It Works:

1. Developer uploads NFT metadata/images via API
2. Files pinned to IPFS nodes
3. Backed up to Filecoin decentralized storage
4. Retrieval via standard IPFS gateways

Sustainability: Funded by Filecoin ecosystem to support Web3 adoption

Metadata Hosting Red Flags

Warning Signs of Poor Metadata Practices:

1. **HTTP URLs:** `https://project.com/metadata/123.json`
   - Centralized, subject to server failure
2. **Mutable tokenURI:** Contract has `setTokenURI()` function
   - Owner can change metadata at will
3. **No pinning documentation:** Project silent on storage strategy
   - Files may not be pinned (risk of loss)
4. **Unknown IPFS gateway:** Custom gateway not reputable
   - Gateway failure = broken NFTs
5. **Unrevealed metadata:** Placeholder images before reveal
   - Risk of metadata not matching expectations

Case Study: CryptoPunks Metadata

CryptoPunks (2017): Early NFT project, metadata evolution

Original Implementation:

- Images and traits stored on-chain (contract storage)
- No external metadata files needed
- True permanence and decentralization

Wrapped Punks (2021):

- ERC-721 wrapper for original punks (compatibility)
- Metadata on IPFS for marketplace display
- Maintains on-chain source of truth

Lesson: On-chain storage provides maximum permanence but at high cost

Lazy Metadata Loading

Lazy Loading: Metadata generated on-demand, not pre-stored

Approach:

- Smart contract computes metadata dynamically
- `tokenURI()` function generates JSON on-chain
- No external storage needed (fully on-chain)

Example: Loot (2021):

- Randomized adventure gear NFTs
- Metadata generated as Base64-encoded SVG in contract
- Zero external dependencies

Trade-offs:

- Maximum decentralization and permanence
- Limited to text/simple graphics (gas constraints)
- Not suitable for complex images or videos

Key Takeaways

1. NFT metadata follows a JSON standard (name, description, image, attributes)
2. IPFS uses content addressing (CIDs) for decentralized, immutable storage
3. Pinning is critical for IPFS file permanence (unpinned files can be lost)
4. Arweave provides guaranteed permanent storage for a one-time fee
5. Metadata permanence challenges include unpinned files, mutable URIs, and centralized servers
6. Best practice: Use IPFS/Arweave, pin to multiple services, freeze metadata after minting

Discussion Questions

1. What are the trade-offs between IPFS and Arweave for NFT metadata storage?
2. Should NFT metadata be immutable, or is mutability acceptable for bug fixes?
3. How can the NFT community ensure long-term metadata availability without centralized pinning services?
4. Is fully on-chain metadata the ideal, or are off-chain solutions sufficient?
5. What happens to NFT value if metadata becomes unavailable but ownership records remain?

Next Lesson Preview

L23: NFT Marketplaces

We will explore:

- OpenSea, Blur, and Rarible business models
- Listing mechanics and order book systems
- Marketplace fees and royalty enforcement
- Wash trading and market manipulation detection
- Aggregators and cross-marketplace trading

Preparation: Create a wallet and browse NFT collections on OpenSea