L21: NFT Technology Deep Dive

Module C: NFTs & Digital Assets

Blockchain & Cryptocurrency Course

December 2025

Learning Objectives

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

- Explain the difference between on-chain and off-chain NFT data
- Understand the role of token URIs in NFT metadata
- Describe the ERC-721 token standard internals
- Analyze NFT provenance and ownership verification
- Evaluate the technical limitations of current NFT implementations

What is an NFT? Technical Definition

Non-Fungible Token (NFT): A unique digital asset on a blockchain

Key Technical Properties:

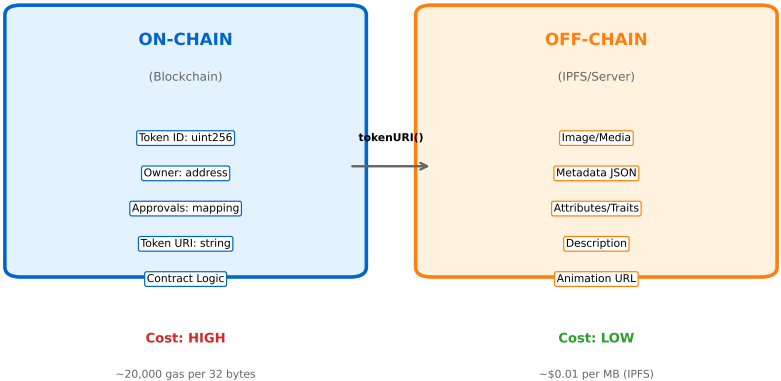
- **Non-fungible:** Each token is unique and not interchangeable
- **Token ID:** Unique identifier within a smart contract
- **Ownership:** Blockchain-verified ownership record
- **Programmability:** Smart contract logic defines behavior

Contrast with Fungible Tokens (ERC-20):

- Fungible: 1 ETH = 1 ETH (interchangeable)
- Non-fungible: Token #1 $\neq$ Token #2 (unique)

On-Chain vs Off-Chain Data

NFT Data Storage: On-Chain vs Off-Chain



Critical trade-off: On-chain is permanent but expensive; off-chain is cheap but risky

ERC-721: The NFT Standard

ERC-721 introduced by Dieter Shirley (CryptoKitties) in 2017

Core Functions:

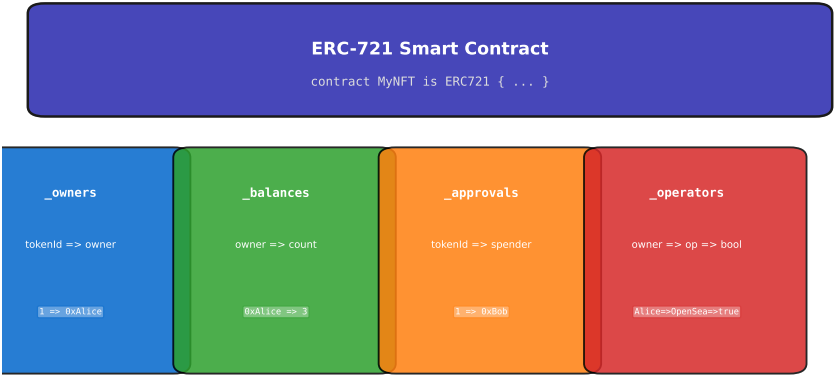
- `balanceOf(owner)` – Returns number of tokens owned
- `ownerOf(tokenId)` – Returns owner of specific token
- `transferFrom(from, to, tokenId)` – Transfers ownership
- `approve(to, tokenId)` – Grants transfer permission
- `tokenURI(tokenId)` – Returns metadata URI

Events:

- `Transfer(from, to, tokenId)` – Emitted on ownership change
- `Approval(owner, approved, tokenId)` – Emitted on approval

ERC-721 State Structure

ERC-721 Contract State Variables



Mappings provide O(1) lookup by key - efficient for sparse token IDs

Mappings provide O(1) lookup efficiency for sparse token ID spaces

Token URI: The Metadata Bridge

tokenURI Function: Links on-chain token to off-chain metadata

Example URI Patterns:

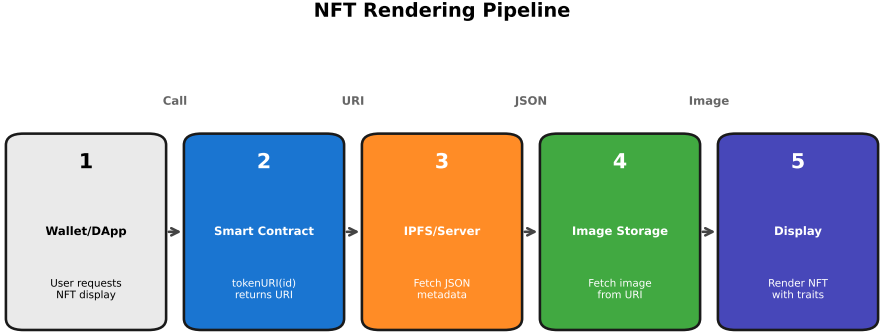
1. **IPFS:** `ipfs://QmXyZ.../metadata.json`
2. **HTTP:** `https://api.project.com/token/123`
3. **Data URI:** `data:application/json;base64,...`

Metadata JSON Structure:

- `name` – Token name
- `description` – Human-readable description
- `image` – URI to visual asset
- `attributes` – Array of traits (rarity properties)

Critical Issue: If metadata server goes down, NFT may become unrenderable

NFT Rendering Pipeline



Critical: If any step fails (server down, IPFS unpinned), NFT becomes unrenderable

Multiple points of failure: contract, IPFS gateway, image server

Provenance: Tracking Ownership History

Provenance: Complete ownership history of an NFT

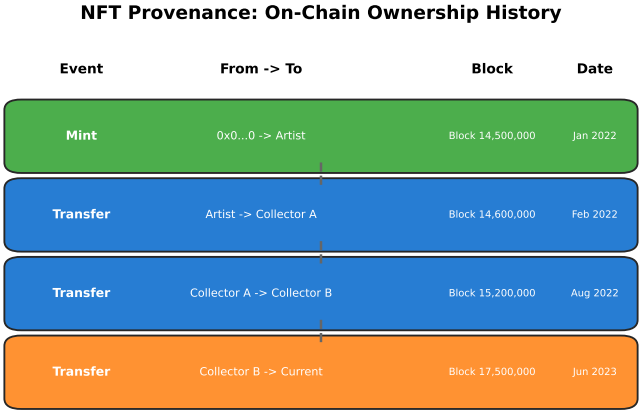
Blockchain Provides:

- Full transaction history via `Transfer` events
- Verifiable chain of custody from mint to present
- Immutable record (cannot be forged or altered)
- Creator verification (original minting address)

Why Provenance Matters:

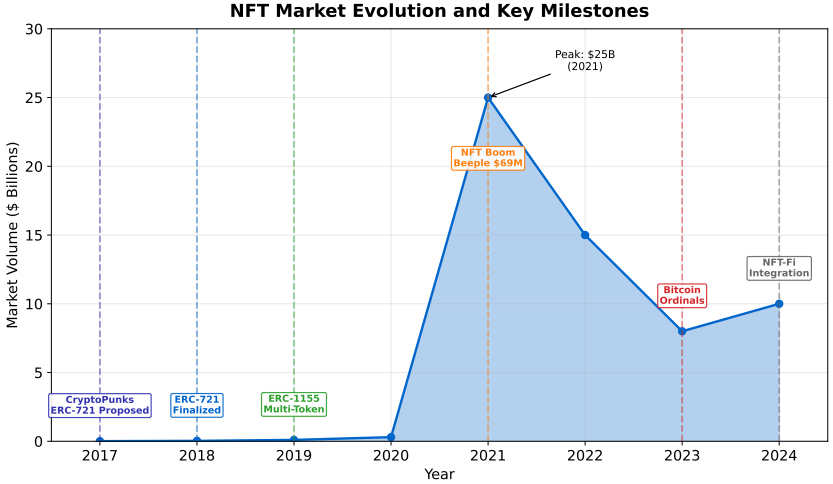
- Proves authenticity and origin
- Increases value (celebrity ownership history)
- Detects wash trading (suspicious transfers)

NFT Provenance Chain



Every Transfer event is immutably recorded on the blockchain

NFT Market Evolution



Market peaked in 2021 but technology continues to evolve

ERC-721 Extensions

Common Extensions Beyond Base Standard:

- **ERC-721 Metadata:** Adds `name()`, `symbol()`, `tokenURI()`
- **ERC-721 Enumerable:** Allows iteration over all tokens
  - `totalSupply()` – Total number of tokens
  - `tokenByIndex(index)` – Get token ID by index
- **ERC-721 Burnable:** Allows token destruction
- **ERC-721 Pausable:** Emergency stop mechanism
- **ERC-2981:** On-chain royalty information

ERC-1155: Multi-Token Standard

ERC-1155: Supports both fungible and non-fungible tokens in one contract

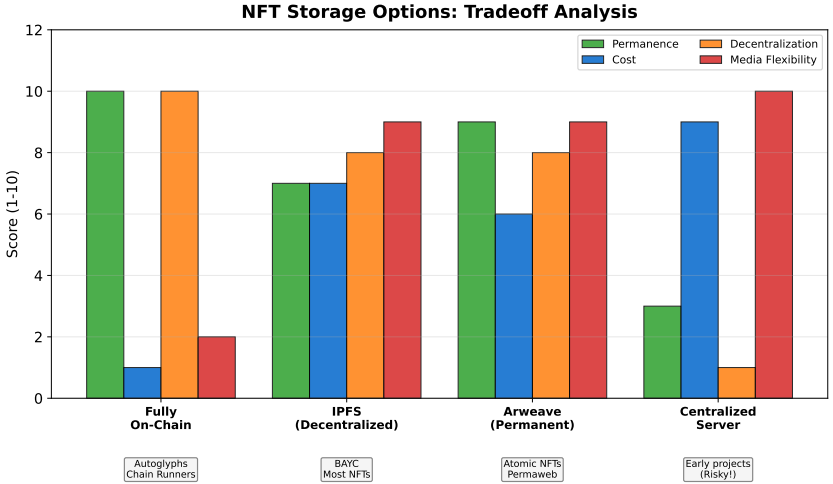
Key Differences from ERC-721:

- Single contract can manage multiple token types
- Batch transfers (gas efficient for multiple tokens)
- Semi-fungible tokens (fungible until uniqueness assigned)
- Used heavily in gaming (items, currencies, NFTs)

Example Use Cases:

- Gaming: 100 fungible "gold coins" + unique weapon NFTs
- Event tickets: 500 general admission (fungible) + 10 VIP (non-fungible)

Storage Options Comparison



IPFS offers best balance for most NFT projects; on-chain for maximum permanence

Fully On-Chain NFTs

On-Chain Maximalism: All data stored on blockchain

Examples:

- **Autoglyphs:** Generative art stored as code in contract
- **Blitmap:** Pixel art encoded in contract storage
- **Chain Runners:** SVG generation entirely on-chain

Advantages:

- True permanence (no external dependencies)
- Maximum decentralization

Disadvantages:

- Extremely high minting costs (gas for storage)
- Limited to simple/generative art (no high-res photos)

2023-2024: Bitcoin Ordinals and Inscriptions

What are Ordinals?

- NFT-like artifacts directly on Bitcoin blockchain
- Uses Ordinal Theory: assigns numbers to individual satoshis
- "Inscriptions": Data embedded in Bitcoin transaction witness
- Enabled by Taproot upgrade (November 2021)

Key Differences from Ethereum NFTs:

- Fully on-chain (no external metadata links)
- No smart contracts (Bitcoin Script limitations)
- Higher permanence guarantees
- Higher inscription costs ($10-100+ per inscription)

Key Takeaways

1. NFTs are blockchain tokens with unique identifiers governed by smart contracts (ERC-721/ERC-1155)
2. Most NFT data is off-chain (images, metadata) with on-chain pointers (tokenURI)
3. Provenance tracking provides verifiable ownership history and authenticity
4. Bitcoin Ordinals (2023): Brought NFTs to Bitcoin via inscriptions
5. Technical limitations include off-chain dependencies, gas costs, and centralization risks
6. Fully on-chain NFTs solve permanence but sacrifice complexity/cost

Discussion Questions

1. Should “true” NFTs require all data to be on-chain, or is off-chain storage acceptable?
2. How does the ERC-721 standard balance gas efficiency with functionality?
3. What are the trade-offs between using IPFS vs. centralized servers for NFT metadata?
4. How does NFT provenance compare to traditional art provenance verification?
5. What innovations could solve the scalability and cost issues of on-chain NFTs?

Next Lesson Preview

L22: NFT Metadata and IPFS

We will explore:

- JSON metadata format standards
- IPFS content addressing and pinning
- Arweave permanent storage
- Metadata permanence and availability challenges
- Best practices for decentralized NFT storage

Preparation: Review IPFS documentation and explore NFT metadata on OpenSea