Introduction to NFT-AI on Solana

AI-Driven NFT Art Gallery

- **Audience**: Developers familiar with AI/NFTs
- **Goal**: Build an NFT gallery with Rust/TypeScript
- **Topics**: Coded demo, AI-agent interaction, scaling/security

Presented by: pupplecat

Why Solana and Code?

Solana:

- 65,000+ TPS
- ~$0.00025 per transaction
- Ideal for scalable NFTs

AI-Driven:

- Generate art and analyze sentiment with Hugging Face
- Autonomous NFT management via AI agents
- Real-time AI on Solana with Pinata, Magic Eden

Let’s code an AI-driven NFT gallery!

AI-Driven NFT Gallery Use Case

Tech Stack:

- **Rust/Anchor**: Smart contracts for minting/updating
- **TypeScript**: Client for AI, IPFS, marketplace
- **Hugging Face**: AI art and sentiment analysis
- **Pinata IPFS**: Metadata storage
- **Magic Eden Devnet**: Marketplace simulation

Workflow:

1. Generate AI art ("futuristic city")
2. Upload to IPFS
3. Mint NFT with Metaplex
4. Update metadata (Popularity: High)
5. Simulate listing (1 SOL)

Letting AI Know Agent Methods

Goal: AI understands and calls agent methods (mint, update, list)

Approach:

- Define methods in `src/agent.ts`
- Expose via JSON schemas

```json
{
  "name": "mint_nft",
  "description": "Mint a new NFT",
  "parameters": {
    "name": { "type": "string" },
    "image_url": { "type": "string" }
  }
}
```

- AI script (`main.ts`) selects methods based on inputs - Example: “Mint NFT, Name: AI Art” → Anchor call

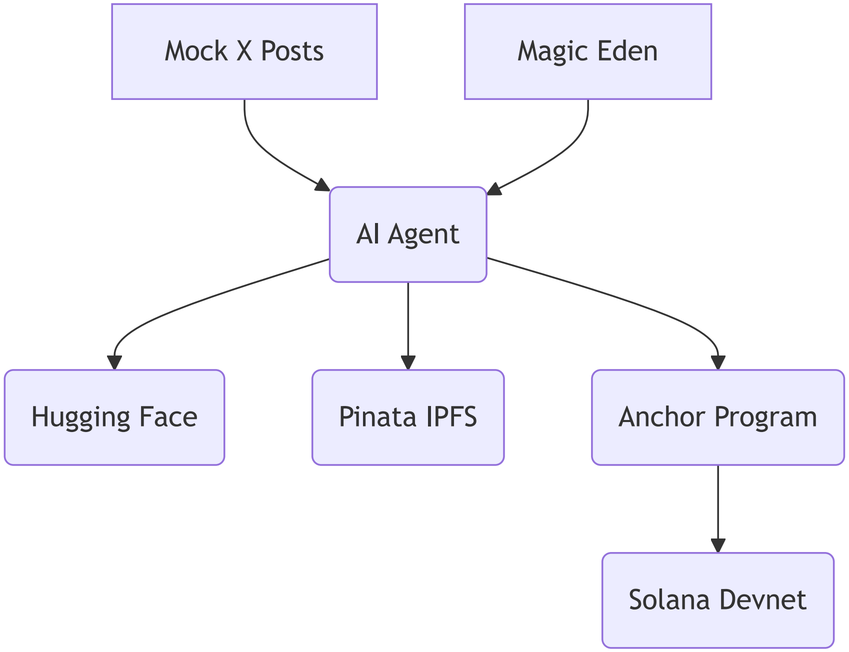
Scaling and Securing NFT-AI Agents

Scaling:

- Batch minting with Anchor
- Real-time AI via Hugging Face
- Solana's 65,000+ TPS
- Solana Agent Kit for advanced agents

Security:

- Wallet encryption
- Transaction validation in Anchor
- Metadata on IPFS/Arweave
- Monitor via logs



- Mainnet: ~0.01 SOL for 100 mints

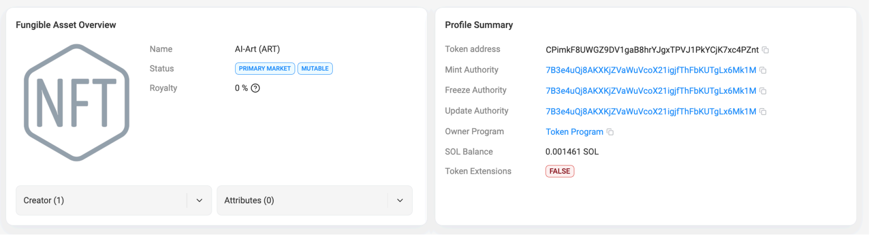
Live Demo

Watch the AI-driven NFT gallery in action!

Steps:

1. Generate art: `yarn generate-art "futuristic city"`
2. Upload to IPFS: `yarn upload-ipfs`
3. Mint NFT: `yarn mint-nft`
4. Update metadata: `yarn update-metadata`
5. Simulate listing: `yarn list-nft`
6. View on Solana Explorer

https://github.com/pupplecat/nft-ai-gallery



Fungible Asset Overview

Name	AI-Art (ART)
Status	PRIMARY MARKET MUTABLE
Royalty	0 %

Creator (1)

Attributes (0)

Profile Summary

Token address	CPimkF8UWGZ9DV1gaB8hrYJgxTPVJ1PkYCjK7xc4PZnt
Mint Authority	7B3e4uQj8AKXKjZVaWuVcoX21igjfThFbKUTgLx6Mk1M
Freeze Authority	7B3e4uQj8AKXKjZVaWuVcoX21igjfThFbKUTgLx6Mk1M
Update Authority	7B3e4uQj8AKXKjZVaWuVcoX21igjfThFbKUTgLx6Mk1M
Owner Program	Token Program
SOL Balance	0.001461 SOL
Token Extensions	FALSE

Metaplex Asset Signer: Key Summaries

- **Execute Asset Signing**: Enables NFTs to act as autonomous agents, signing transactions via the Asset Signer PDA.
- **Asset Signer PDA**: Unique PDA derived from the NFT's asset account, controlled by Metaplex Core, signs actions without a private key.
- **Execute Instruction**: Triggers Metaplex Core to validate and execute NFT actions (e.g., listing, updating metadata).
- **External Signer**: Authorized entity (owner, delegate, plugin authority) submits off-chain transaction to invoke Execute.
- **Validation & Signing**: Program validates signer and plugin logic, signs action with Asset Signer PDA via invoke_signed.
- **Autonomy**: Supports programmable NFT actions, often AI-driven, for marketplaces, DeFi, and more.
- **Security**: Separates external signer (transaction submission) from Asset Signer PDA (action signing).

Resources

- Execute Asset Signing: https://developers.metaplex.com/core/execute-asset-signing
- Solana-Ai: https://solana.com/developers/guides/getstarted/intro-to-ai
- Solana-Agent-Kit: https://github.com/sendaifun/solana-agent-kit/tree/v2/examples
- Security: https://www.helius.dev/blog/how-to-build-a-secure-ai-agent-on-solana

Q&A

Ask away!

- Scaling to mainnet?
- Advanced Anchor programs?
- Real Magic Eden integration?