

# Technical Architecture of ArtChain

## 1. Executive Summary

This project is an AI-powered NFT art platform that allows users to generate images based on an artist's unique style using Stable Diffusion integrated with LoRA models. With the help of an AI agent, users can modify Stable Diffusion settings, request explanations, and refine their prompts for better results. Once satisfied, they can mint their AI-generated artwork as NFTs on the SUI blockchain. The platform ensures a seamless and intuitive creative process while leveraging smart contracts to manage transactions transparently. This includes handling minting fees, fund distributions, and—most importantly—automated royalty payments to artists who contribute their datasets, ensuring they are fairly compensated for their work.

## 2. Core Architecture Components

### 2.1. Frontend Architecture

#### 2.1.1. Tech stack

- React.js for web applications and React Native for mobile platforms (optional).
- Web3.js and the SUI SDK to facilitate blockchain interactions.

#### 2.1.2. Features

- **User Authentication:** Using Web3-based wallet connections like Metamask, SUI Wallet, or custom wallets.
- **Gallery Interface:** Allows browsing art styles, filtering by artist names, genres, etc.
- **Minting Function:** Users can mint the currently generated image as an NFT after selecting an art style and configuring AI model parameters.
- **Transaction Monitoring:** Users should be able to view transaction statuses and royalties received by artists.
- **On-Chain Verification:** Blockchain-based proof of ownership ensures authenticity and prevents unauthorized alterations to minted NFTs.
- **Real-Time Notifications:** Users receive updates on minting progress, transaction confirmations, and AI generation statuses.

## 2.2. Backend Infrastructure

### 2.2.1. Tech stack

- **FastAPI** (or Flask) for handling API requests efficiently.
- **Python** as the primary backend language for AI processing and API development.
- **AutoGen** to manage AI agent workflows, including self-reflection and iterative improvements.
- **LLaMA 3.3-70B** as the core large language model (LLM) for text-based interactions.
- Stable Diffusion for AI-generated image creation with configurable parameters.
- **LoRA (Low-Rank Adaptation)** to fine-tune Stable Diffusion models for customization.
- Sui Blockchain for NFT-related functionalities and Web3 interactions.
- **Pinata** - IPFS for decentralized storage of artwork

### 2.2.2. Features

- **Transaction Management:** Handle minting requests, royalties for artists, and smart contract calls.
- **Artwork Storage:** The backend triggers a minting transaction on the SUI blockchain, creating an NFT with metadata pointing to the generated image on IPFS.
- **Royalty Distribution:** Ensure that royalties are correctly distributed to the artist whenever their art style is used for minting.
- **AI-Powered Image Generation:** Users can generate art using Stable Diffusion with customizable parameters, including resolution, sampling method, and CFG scale.
- **Iterative Refinement:** The AI agent can add adjustments or improvements to generated images based on user feedback, adjusting prompts while preserving core concepts.
- **Configurable AI Settings:** Users can modify Stable Diffusion parameters such as sampling steps, width, height, and seed for precise image control.
- **AutoGen AI Agent:** An intelligent assistant guides users through AI-generated art creation, provides explanations of settings, and helps refine outputs.

## 2.3. Blockchain Layer

### 2.3.1. Tech stack

- SUI blockchain: highly scalable and low latency
- The MOVE language: object-oriented and effective resource allocation programming language

### 2.3.2. Features

- **NFT Smart Contract:** A contract to mint NFTs. The NFT will include metadata about the art (artist, style, etc.) and can be tied to the AI model generated image. The contract should allow users to mint NFTs using their wallets and select the artist's style (and other parameters: colors, shapes, etc. - optional).
- **Payment Smart Contract:** A mechanism for automatically handling minting fee payment and distributing royalties when an NFT is minted. Each artist gets a percentage of the minting fee or sale price every time their art style is used.
- **Artist Registration Smart Contract (not developed yet):** A contract that allows artists to register their art styles, including the specific diffusion model inputs required to generate art in their style. Could be used to track the artist's performance and minting statistics.

## 2.4. AI Diffusion Model

### 2.4.1. Tech stack

- Diffusion Models (Stable Diffusion)
- Hugging Face API
- LoRA Models

### 2.4.2. Features

- **Train Stage:** The models are fine-tuned with specific styles of registered artists using LoRA.
- **Generate Stage:**
  - **Input:** Take user's prompts as input to generate.
  - **Output:** Generate image and allow users to mint as NFTs and the metadata (artwork, artist, associated rights, etc.) should be embedded in the NFT metadata on the SUI blockchain.

## 3. Interactions

### 3.1. Workflow Diagram

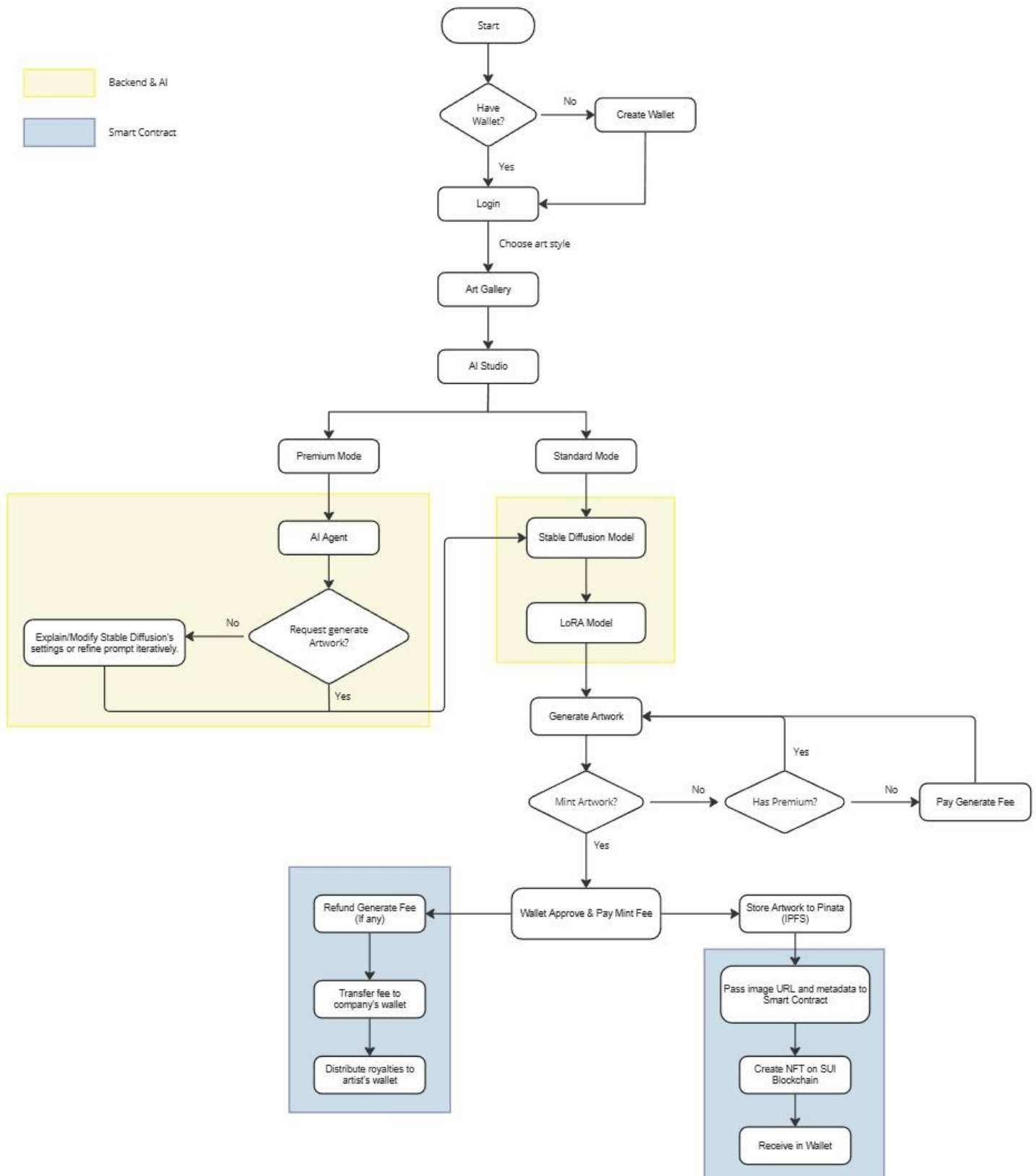


Figure 1: System Workflow

The workflow diagram above provides a comprehensive overview of the AI-powered NFT image generator platform built on the SUI blockchain, detailing the user journey from authentication to NFT creation and transaction handling.

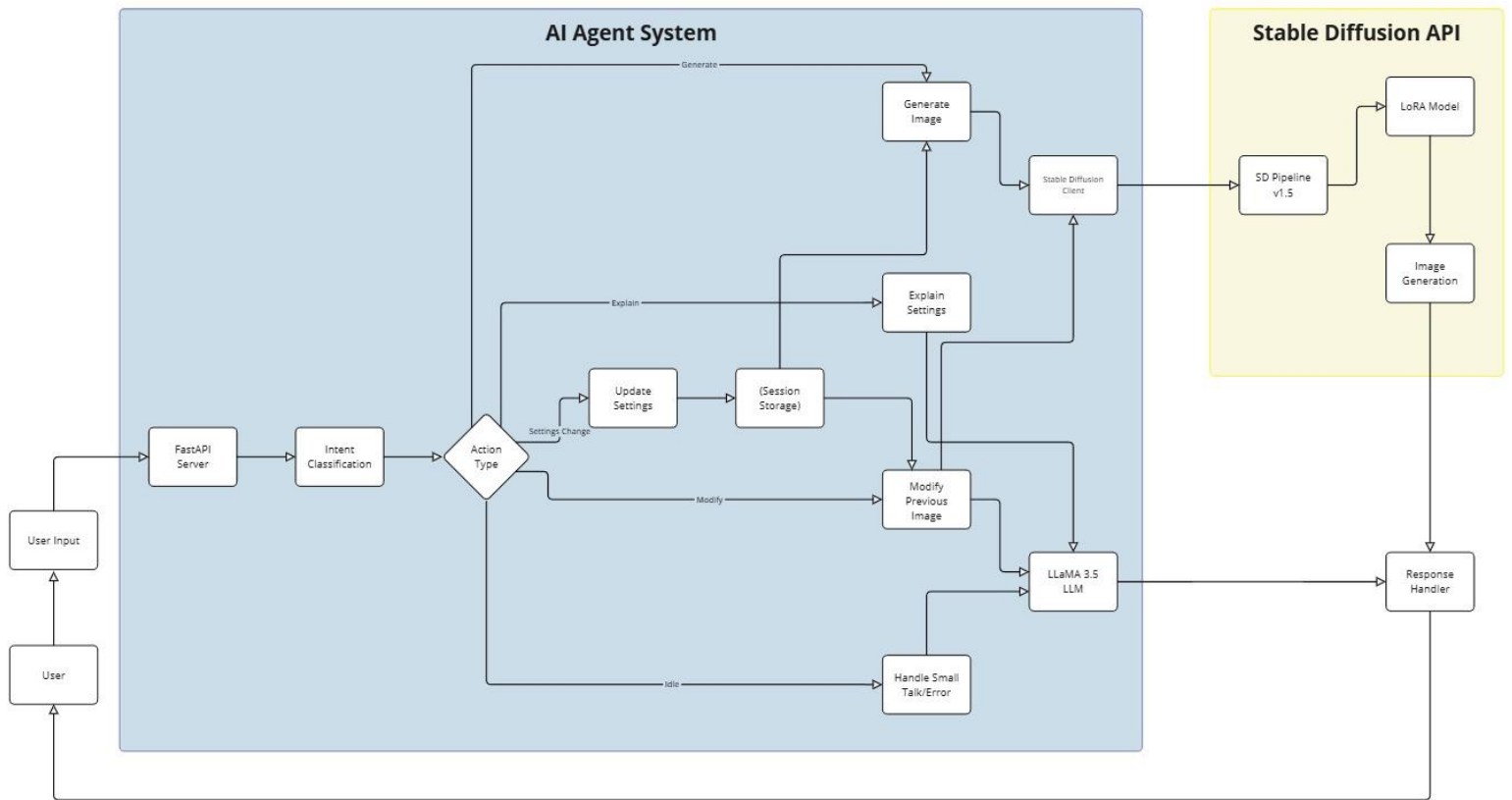
The process begins with wallet verification, where users either connect their existing Web3 wallet or they can create a new one if they do not have one. After logging in, users navigate to the Art Gallery to browse available styles. From here, they can choose between **Premium Mode** and **Standard Mode** for artwork generation in the **AI Studio**.

In **Premium Mode**, the user can request the AI agent to generate image, ask for iterative refinement prompts and explain or adjust Stable Diffusion settings. If the user decides to proceed with generating artwork, the AI agent processes the request. Meanwhile, in **Standard Mode**, the **Stable Diffusion** model—enhanced with a **LoRA** model—generates artwork directly.

Once the artwork is created, users can choose whether to mint it as an NFT. If the user has **Premium access**, the minting process continues without additional charges. Otherwise, they must pay a generation fee before proceeding. Upon approval, the system stores the artwork on **Pinata (IPFS)**, passes the metadata to the smart contract, and creates the NFT on the **SUI blockchain**. The NFT is then transferred to the user's wallet.

Additionally, the smart contract executes financial transactions, handling **royalty distribution** and **payment processing**. If applicable, any **image generation fees** will be refunded, the remaining amount is transferred to the company's wallet, and royalties are allocated to the artist's wallet.

The workflow concludes with the user receiving the NFT, after which they can either hold onto it or trade it in the marketplace, maintaining a continuous engagement loop.



**Figure 2: AI Agent System**

This diagram illustrates an AI agent system that processes user input to generate images using **Stable Diffusion**. The workflow starts with the **FastAPI Server**, which receives user input and passes it to **Intent Classification**. Based on intent, the **Action Type** decision node directs the request:

- **Generate Image** sends the request to **Stable Diffusion**, which uses a **LoRA Model** to produce the image.
- **Explain Settings** provides details about **Stable Diffusion** parameters.
- **Update Settings** modifies stored user preferences.
- **Modify Previous Image** refines existing outputs using **LLaMA 3.3-70B**.
- **Handle Small Talk/Error** responds to casual input.

The **Response Handler** delivers results back to the user, ensuring smooth interaction and iterative image refinement.

## 3.2. System Interaction

Below is the textual guide to map the flow of the system:

1. **Frontend Layer (UI)** → Users interact with the marketplace. It sends requests to:
  - **Backend (API Server)**
  - **SUI Blockchain** (for wallet interactions and minting)
2. **Backend API** → Handles requests, including:
  - Sending input parameters to the **AI Diffusion Model**.
  - Sending input parameters to the **AutoGen AI agent**.
  - Initiating minting requests on the **SUI Blockchain**.
  - Storing image to **Pinata (IPFS)**.
3. **AI Diffusion Model** → Processes the input data and generates art. This data is then sent to **IPFS** for storage.
4. **AutoGen AI Agent** → User can request the AI agent to generate image, ask for iterative refinement prompts and explain or adjust Stable Diffusion settings.
5. **SUI Blockchain** → Executes smart contracts, including:
  - **Minting NFT**: Registers the NFT with metadata, pointing to the stored image on IPFS.
  - **Payment Smart Contract**: Handles minting payments and royalties to the artist whenever their style is used for minting.

## 4. Technical Considerations

### 4.1. Scalability and Performance

As the platform grows, it needs to handle increasing numbers of users, AI-generated image requests, and blockchain transactions without slowing down. To achieve this, we plan to leverage cloud computing services like AWS or Google Cloud, which allow us to scale our backend dynamically based on demand. For AI image generation, we will use high-performance GPU instances such as AWS EC2 P4d or Google Cloud's A100 GPUs, ensuring that users experience minimal delays even during peak usage.

To keep things running smoothly, we use asynchronous processing with Redis-based job queues (e.g., Celery or RQ). This means that instead of making users wait for complex tasks like AI image generation or NFT minting to complete, these tasks run in the background while

users continue interacting with the platform.

In the future, we can further improve scalability by introducing Kubernetes for automatic service scaling and using a content delivery network (CDN) like Cloudflare to speed up image and metadata retrieval. These improvements would make the platform more responsive and capable of handling even higher traffic loads.

## 4.2. Security

Security is critical in an NFT marketplace, especially when handling blockchain transactions and valuable AI-generated art. We take multiple precautions to protect user data, transactions, and digital assets.

- **Wallet Authentication & Encryption:** Users authenticate using Web3 wallets like Metamask or SUI Wallet. These connections are encrypted to ensure secure interactions. (Already developed)
- **Smart Contract Security:** Our NFT and payment smart contracts follow best practices, including access controls and transaction validation, to prevent vulnerabilities like reentrancy attacks. We also plan to conduct security audits to ensure robustness.
- **Data Integrity & Intellectual Property Protection:** Since NFTs store metadata on IPFS, the artwork remains immutable and verifiable. We also explore fingerprinting techniques to track AI-generated images and prevent unauthorized duplication.
- **API Security & DDoS Protection:** We implement rate limiting to prevent abuse and use Web Application Firewalls (WAF) to guard against distributed denial-of-service (DDoS).

## 4.3. System Optimization and Feature Improvement

While the system is designed for efficiency, we continuously look for ways to improve its speed and reliability.

- **AI Model Optimization:** By fine-tuning LoRA models and using techniques like model distillation, we can reduce processing time while keeping image quality high.
- **AI Agent Fine-tuning Improvement:** We can develop automated configuration and minting function to the AI agent. If the user provides a vague prompt, the agent will suggest an improved prompt and the best Stable Diffusion settings base on prompt



topic, and then if the user want to mint the artwork without clicking any button, they can ask our agent to do it for them.

- **Gas Fee Optimization:** Since blockchain transactions come with costs, our smart contracts are optimized to minimize gas fees. In the future, we may explore Layer 2 scaling solutions to reduce fees further.
- **Parallel Processing & GPU Utilization:** We plan to enhance how AI tasks are distributed across GPUs, using methods like batch inference and mixed precision training to speed up image generation.
- **Marketplace:** To enhance the user experience, we can develop an optimized marketplace for AI-generated NFTs, including improved search and filtering functionalities, faster NFT listing, and more efficient transaction processing. Implementing caching mechanisms and optimized database queries would allow users to seamlessly browse, buy, and sell NFTs.