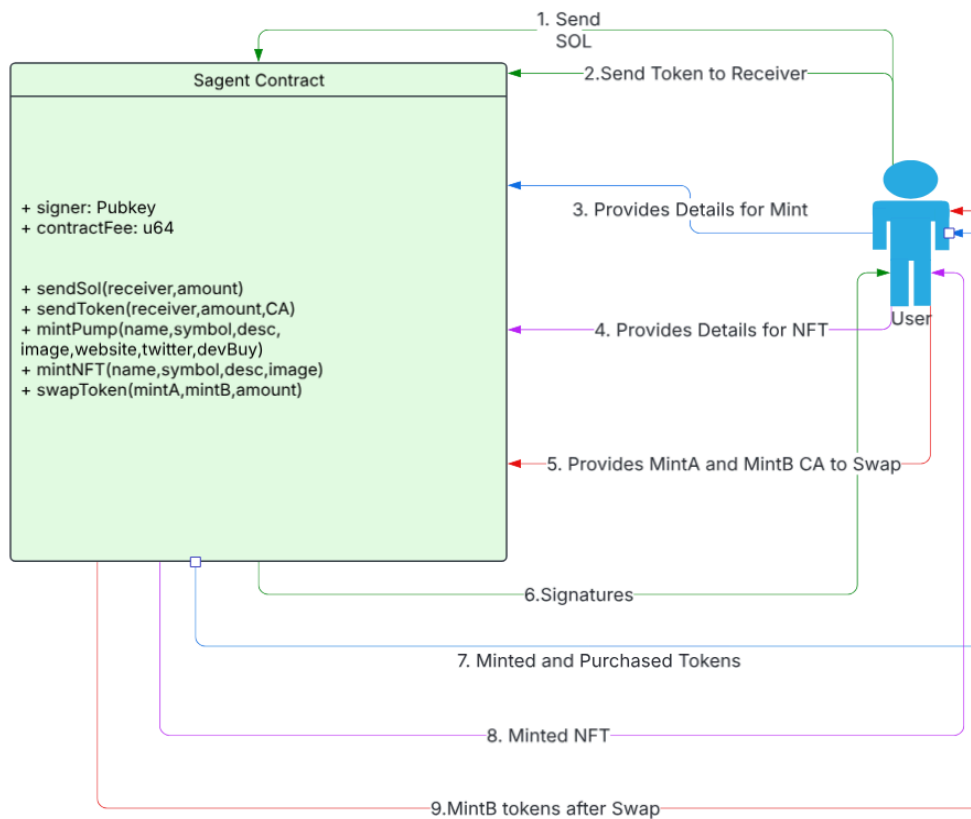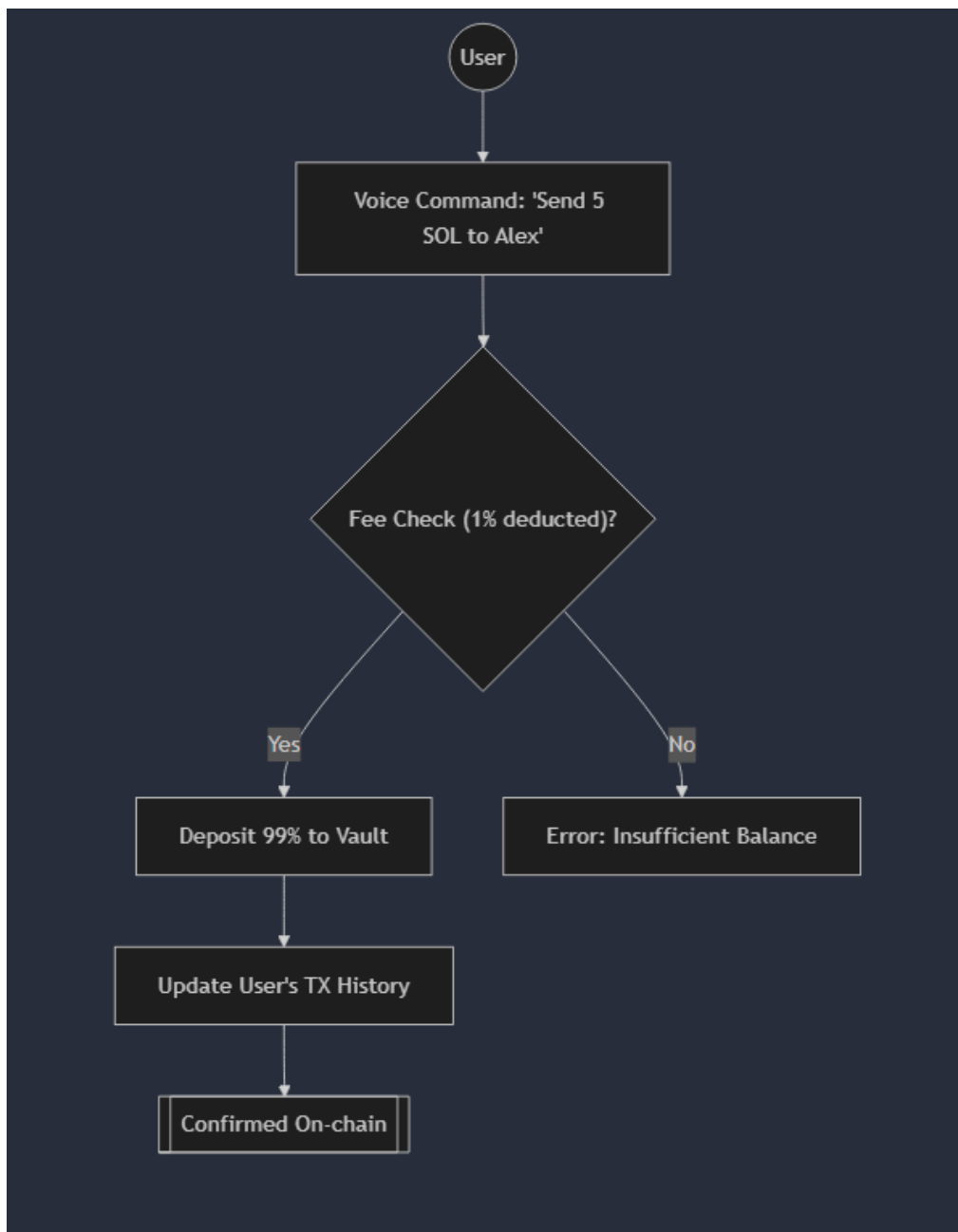# Sagent POC Requirements

- The platform shall allow users to interact with the Solana blockchain through a Large Language Model (LLM) chatbot, using either text or voice commands.

- The platform shall support a variety of blockchain tasks, including sending/receiving tokens, creating NFTs, launching tokens, and executing DeFi swaps.

- The chatbot shall be capable of understanding and interpreting natural language user requests related to supported blockchain tasks.

- The platform shall integrate with a browser wallet (e.g., Phantom) for secure transaction signing and user authorization.

- The platform shall utilize n8n for automating workflows and streamlining blockchain operations.

- The platform shall prioritize ease of use, efficiency, security, and accessibility for users with varying levels of blockchain expertise.
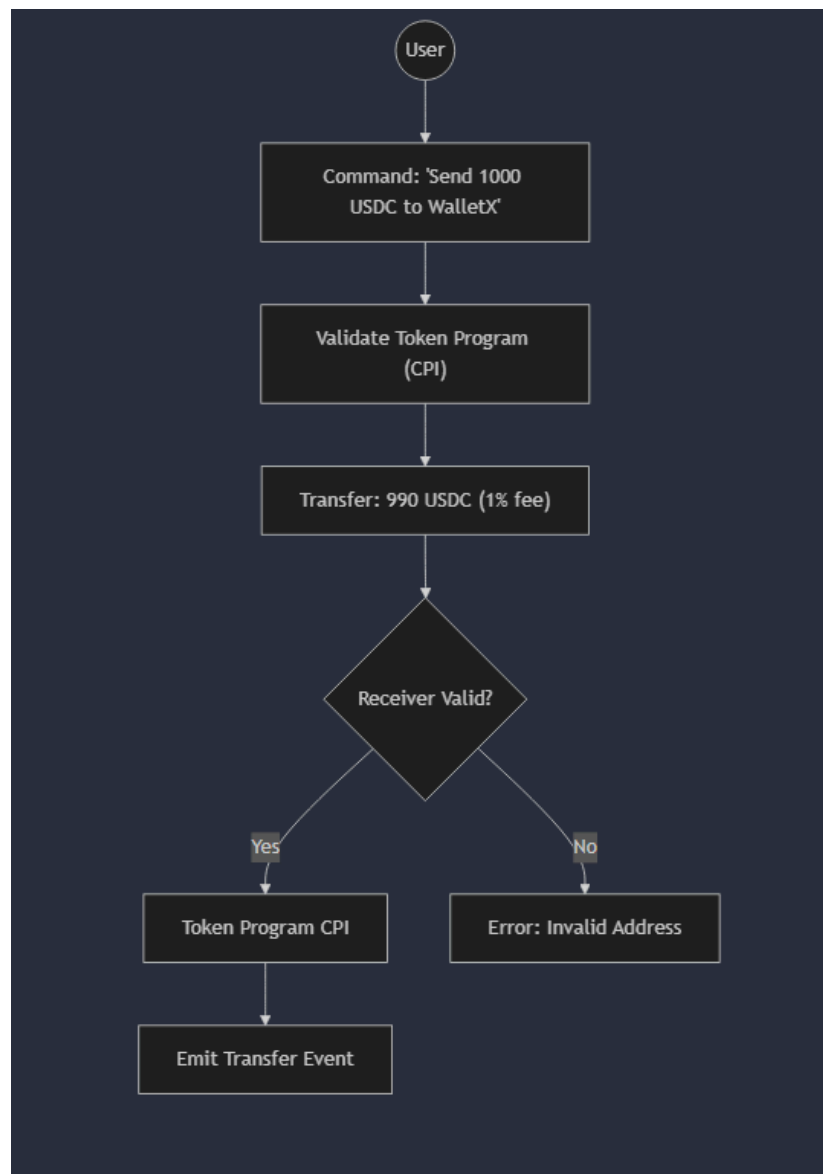
Overview

Sagent Contract

+ signer: Pubkey
+ contractFee: u64

+ sendSol(receiver,amount)
+ sendToken(receiver,amount,CA)
+ mintPump(name,symbol,desc,
image,website,twitter,devBuy)
+ mintNFT(name,symbol,desc,image)
+ swapToken(mintA,mintB,amount)

1. Send SOL
2. Send Token to Receiver
3. Provides Details for Mint
4. Provides Details for NFT
5. Provides MintA and MintB CA to Swap
6. Signatures
7. Minted and Purchased Tokens
8. Minted NFT
9. MintB tokens after Swap

User

---

1. **Send SOL**
   The user initiates sending SOL. The contract should receive it, deduct fees and send to the receiver.
2. **Send Token to receiver**
   User sends a token, contract should find PDA of the token with token program according to the receiver address, deduct fees and send tokens to receiver ATA.
3. **Provides details for Mint**
   User gives mint details. Contract creates the token, sets up metadata, sets up initial buy and sends to the user ATA if applicable after deducting fees.
4. **Provides details for NFT**
   User provides NFT details. Contract mints the NFT, uploads to Arweave, attaches royalties, and assigns PDA ownership.
5. **Provides mintA and mintB CA to Swap**
   User requests a swap. Contract checks rates via Jupiter/Raydium, builds routes, applies slippage, and logs fees.
6. **TX Signatures**
   Contract validates signatures, ensures compliance, checks balances, and sends confirmation of TX hash.
7. **Minted and Purchased Tokens**
   Transfers tokens to user, updates holdings, and provides explorer links.
8. **Minted NFT**
   Sends NFT to user's wallet.
9. **MintB tokens after Swap**
   Executes token swap, confirms balance, deducts fees, and logs details.

Send SOL


```
                          ( User )
                             │
                             ▼
                  ┌──────────────────────┐
                  │  Voice Command: 'Send 5
                  │      SOL to Alex'      │
                  └──────────────────────┘
                             │
                             ▼
                        ◇ Fee Check (1% deducted)? ◇
                          /                    \
                      Yes/                      \No
                        ▼                        ▼
            ┌──────────────────┐      ┌──────────────────────┐
            │ Deposit 99% to Vault │    │ Error: Insufficient Balance │
            └──────────────────┘      └──────────────────────┘
                        │
                        ▼
            ┌──────────────────────┐
            │ Update User's TX History │
            └──────────────────────┘
                        │
                        ▼
            ┌──────────────────┐
            │ Confirmed On-chain │
            └──────────────────┘
```

User sends SOL.
Contract deducts fees and sends to the receiver wallet.

Send Token

User sends a token and provides the receiver's address.

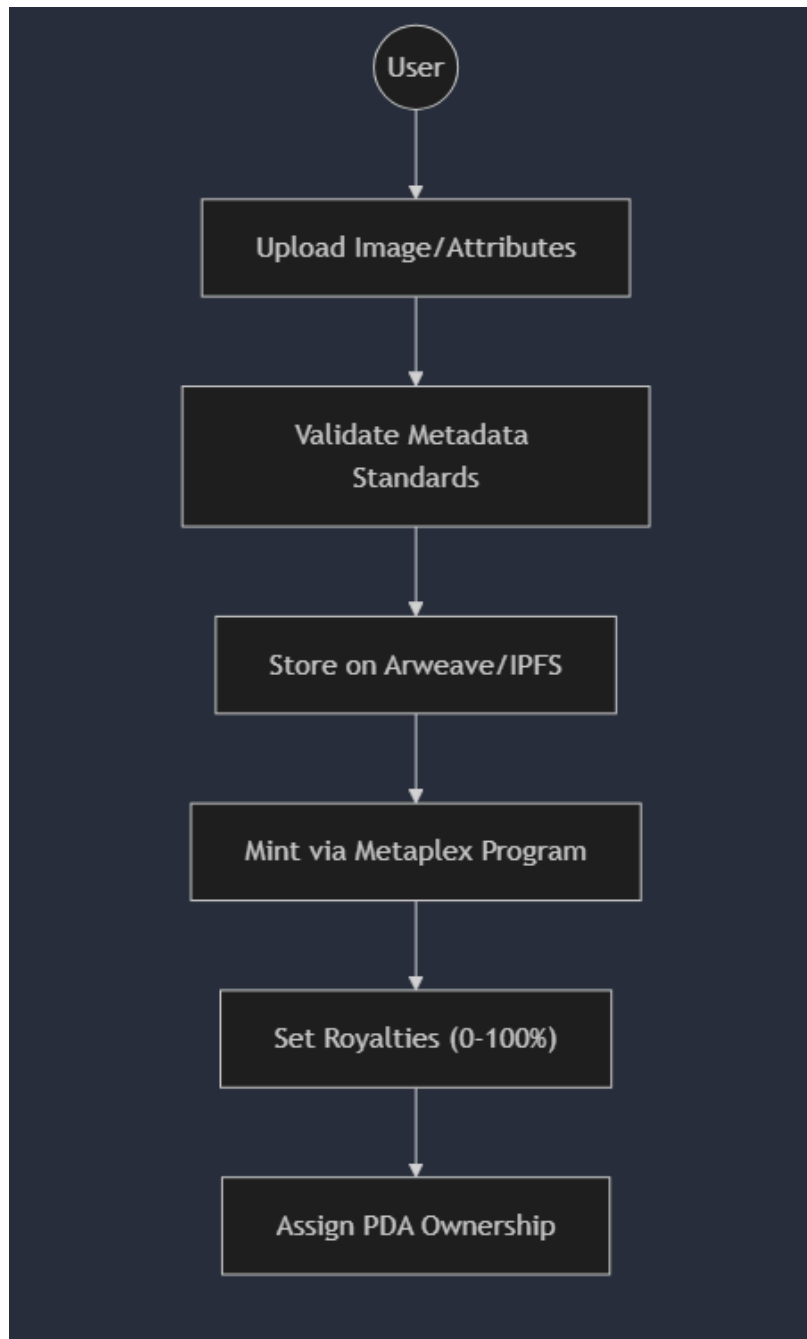Contract checks token address and receiver, deducts fee and completes transfer.

Mint Pump

User provides details.

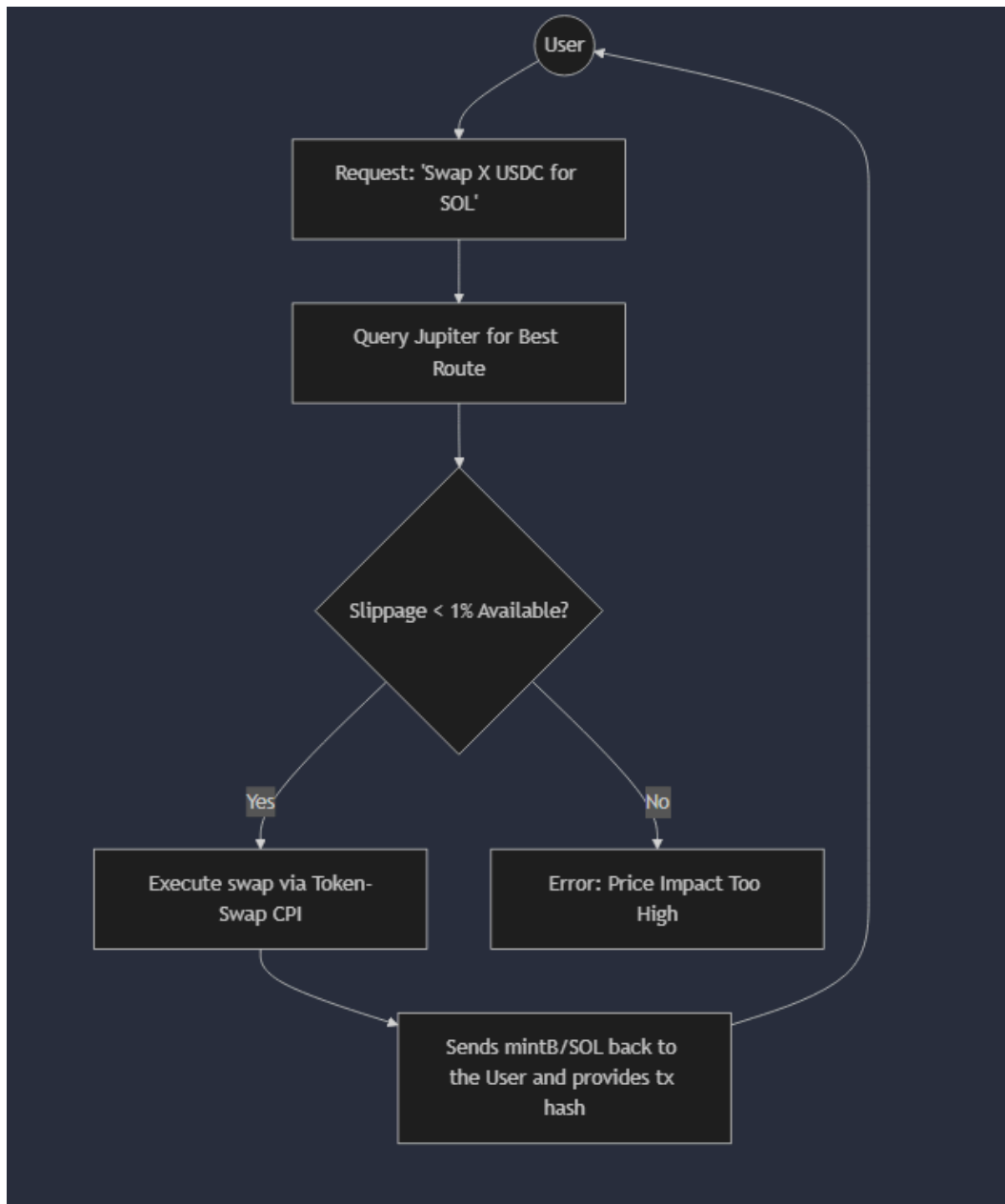Contract initializes mint, handles dev buy and distributes tokens.

Mint NFT

User provides attributes' details.

Contract validates, mints, sets royalties and sends the NFT to the User.

Swap Token

The user requests token exchange, providing mintA tokens and mintB address.

The contract will query Jupiter API for optimal routes, execute swap via token-program CPI, and sends mintB to the user providing the tx hash.