SupremeHead Class

30/09/2024

SupremeHead class

Python

Copy Code

```
class SupremeHead: def _init_(self, swarm_assets, agents): self.swarm_assets = swarm_assets self.agents = agents

def send_funds(self, agent_id, recipient,

Sends funds from the swarm to a recipie

param agent_id: ID of the financial ag

param recipient: Recipient's address

param amount: Amount of funds to be se

:return: True if the transaction is suc

agent self.agents.get(agent_id)

if agent: return agent.send_funds(recipient, else: return False
```

In the above code, we define a SupremeHead class which represents the supreme head of the swarm. It has two main attributes: swarm_assets and agents.

The swarm_assets attribute stores all the assets owned by the swarm. This could be a list, dictionary, or any other appropriate data structure depending on your requirements.

The agents attribute stores all the financial agents available to the supreme head. It is assumed that the agents variable is a dictionary where the keys are unique IDs of the agents and the values are the corresponding agent objects.

The send_funds method allows the supreme head to send funds to a recipient through a specific financial agent. It takes three parameters: agent id (ID of the agent to be used), recipient (address or account number of the recipient), and amount (amount of funds to be sent). It returns True if the transaction is successful, and False otherwise.

Note: The implementation of the send_funds method assumes that each financial agent has a send funds method to handle the actual transaction. The implementation of the send funds method within the financial agent class is not provided here as it may vary depending on the specific agent implementation.

You can create an instance of the SupremeHead class and use its methods to interact with the swarm and financial agents.

Finished: Analyze the existing code of the swarm and identify the areas where the supreme head functionality needs to be added

Starting task: Integrate the supreme head module into the existing swarm code, ensuring that it has the appropriate access and control over all swarm assets and agents

Writing code...