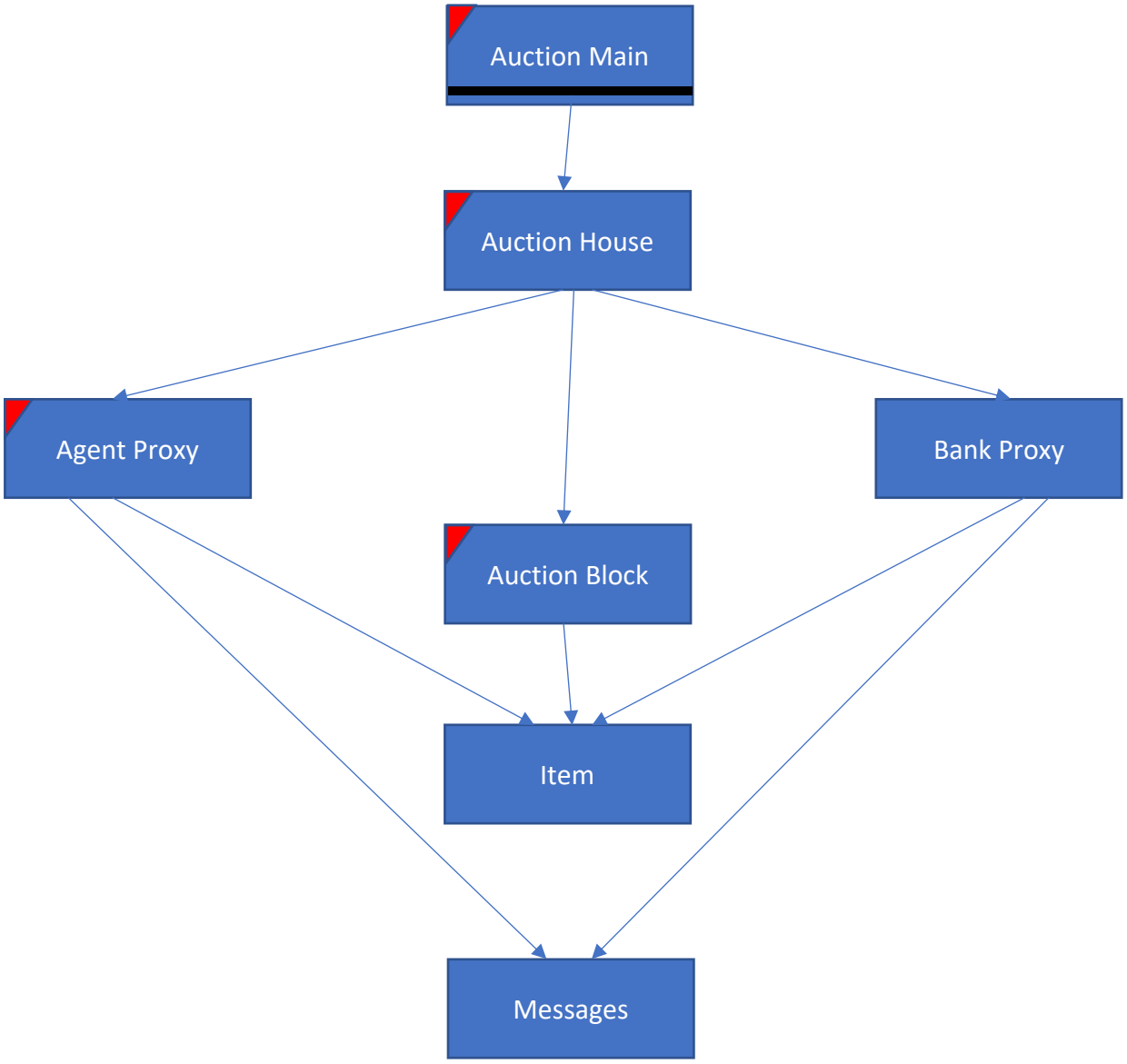
Auction Main
Auction House
Agent Proxy
Bank Proxy
Auction Block
Item
Messages

Implementation Details:

AuctionMain: Main Entry point of the auction house creates and initializes an AuctionHouse passing it the command line arguments for initialization.

AuctionHouse: Begins by creating the components of the auction house. The components are as follows AuctionServer, AuctionBlock, and the BankProxy. The AuctionServer is an inner class of the Auctionhouse. When the AuctionServer accepts a connection from an agent it stores a new Agent Proxy in a HashMap within the AuctionHouse. The AuctionHouse uses this HashMap of agent proxies to notify the appropriate agents when auction events occur on the AuctionBlock and it also facilitates bidding mechanics like checking with the bank for available funds. It Essentially works as middleman between the agents, bank, and the auction block.

AuctionBlock: A Separate Thread that runs the auction by maintaining a list of items and using the thread to incrementally update and expire items as times goes by. While this process is running it will call various methods in the AuctionHouse that update the agents as the auction runs. Extends HashMap to store Items currently up for bidding.

AgentProxy: Reads and process messages from the Agent Program on a separate thread and encapsulates all outgoing messages from the AuctionHouse to the agent (Bidding).

BankProxy: Handles Communication between the AuctionHouse and the Bank. Uses callable to efficiently query and register with the bank, retuning the information to the AuctionHouse.

Item: Contains information that specifies information about an item in the auction. Keeps track of its time left up for auction and hold values for current highest bid and the highest bidders account ID.

Messages: Standardized message system utilizing java serialization to easily encapsulate auction information and send it between multiple programs.