# BFX Developer Challenge - C#

Hello and congratulations on your interview progress with Bitfinex! As part of the next stage, we have developed a coding challenge for you to complete.

## Challenge Overview

Develop a simple Peer-to-Peer (P2P) auction system that utilizes Remote Procedure Calls (RPC) as its primary form of communication.

## Feature Requirements

### Auction Initialization

* Implement an RPC Client capable of initiating auctions (e.g., selling a picture for a specific amount).
* Upon auction creation, the client must inform other participants in the network, implying that each client also functions as an RPC Server.

### Bid Handling

* Allow participants to place bids on auctions.
* Each bid must be broadcasted to all network participants.

### Auction Conclusion

* Implement distributed transaction handling upon auction closure.
* Notify all participants of the auction outcome.

## Example Scenario

1. Client#1 opens auction: sell Pic#1 for 75 USDt.
2. Client#2 opens auction: sell Pic#2 for 60 USDt.
3. Client#2 bids 75 USDt for Client#1's Pic#1.
4. Client#3 bids 75.5 USDt for the same.
5. Client#2 counters with an 80 USDt bid.
6. Client#1 finalizes auction, informing all about the sale to Client#2 at 80 USDt.

## Technical Requirements

* Adhere to a P2P architecture (avoid traditional client/server models).
* Utilize JSON RPC (2.0) or gRPC for node communication.
* Implement a command-line interface (CLI); a graphical UI is not required.
* Use SQLite for any database requirements.
* Ensure compatibility across multiple operating systems: Linux, OSX, Windows.

## Timeframe and Completion

Aim to complete within 6-8 hours however it is understood that full completion within this time may not be feasible.   
  
If you are unable to complete the task please document the missing elements and how you would implement them.  
  
If your solution has any limitations or issues, document potential solutions and how you would implement them if more time were available.

Good Luck!

## Further Resources:

* JSON RPC 2.0 Specification: [JSON RPC Website](https://www.jsonrpc.org/specification)
* gRPC and C#: [gRPC Documentation](https://grpc.io/docs/languages/csharp)