Communication Protocol -
All in this table will be the form of a JSON object -
{
command: String
data: [Array of Strings]
}

Command	List of Strings data
“Join” Front end sends to middleware if client chooses to join a game	
“Host” Front end sends to middleware if client chooses to host a game	
“gameOption” Middleware will send this to front end if the client chooses the above “join” option, it is to show you the available games from the backend’s UDP.	[game name, number of players joined, max number of players in game, ip address of the host hosting this game, port number] You will only display the game name, number of players joined and max players in game
“joinStart” Front end will send this to middleware, when the client chooses from the list of ‘gameOptions’	[ip address of game joining, port number of game joining]
“hostStart” Front end will send this to middleware, when the client wants to host a game and chooses the game/number of players they want	[gameName, gameId, numberPlayers]
“ready” This is a command the middleware will send to the front end when a player joins the game	
“initHand” At the beginning of each hand this is sent to the front end from middleware to indicate the client’s player id, the play order and their	[client player id, player order ids separated by “|”, scores separated by “|”, client’s card numbers separated by “|”, corresponding client’s card suits separated by “|”, trump suit if there is one (if there isn’t a | will be in

scores for the current game, the client's deck of cards is sent, if there is one - the trump suit.	place)]. Example: ["2", "0|1|2|3", "0|0|0|0", "2|3|4|5", "CLUBS|DIAMONDS|HEARTS|CLUBS","CLUBS"] ["2", "0|1|2|3", "0|0|0|0", "2|3|4|5", "CLUBS|DIAMONDS|HEARTS|CLUBS","CLUBS"] Example 1 and 2 both have - client id is 2, play order is 0,1,2,3, scores are 0,0,0,0 respectively, and the client's deck is 2 of clubs, 3 of diamond, 4 of hearts and 5 of clubs. Example 1 has no trick suit (indicated by "|") whereas example 2 has a trick suit ("CLUBS").
"validMove" This is a command middleware sends to front end after a card is played	
"invalidMove" This is a command middleware sends to front end after a card is played	
"trickEnd" Sent from middleware to front end	[play order ids seprated by "|", corresponding scores to those player ids] Example: ["0","0|1|2|3", "3|0|3|0"] This means the order for the next trick will be players 0,1,2,3. Player 0 won last trick player 0 and 2 have won 3 TRICKS Player 1 and 3 have won 0 TRICKS
"gameEnd" Sent from middleware to front end	[winning team ids seprated by "|"] Example: ["1|4"] - this means players 1 and 4 won the game

These follow the supergroup protocol:

JSON object for playing cards, this is bidirectional between front end and middleware:

```
{
Type: “play”,
Suit: “CLUBS/DIAMONDS…”,
Rank: “4,KING…”
}
```