

LIAR'S POKER

Distributed Systems
A.Y. 2024/2025

OUR TEAM

Buda Francesco

Sanchi Emanuele

Severi Tommaso

INTRODUCTION

This project aims to develop a web-based game application for playing Liar's Poker, a fun and engaging card game centred around deception and bluffing. Inspired by traditional poker, Liar's Poker is played with a standard deck of cards and is suitable for two or more players. Our application will provide web interface where users can enjoy the game with others in real-time.

RULES

Turn-based card game

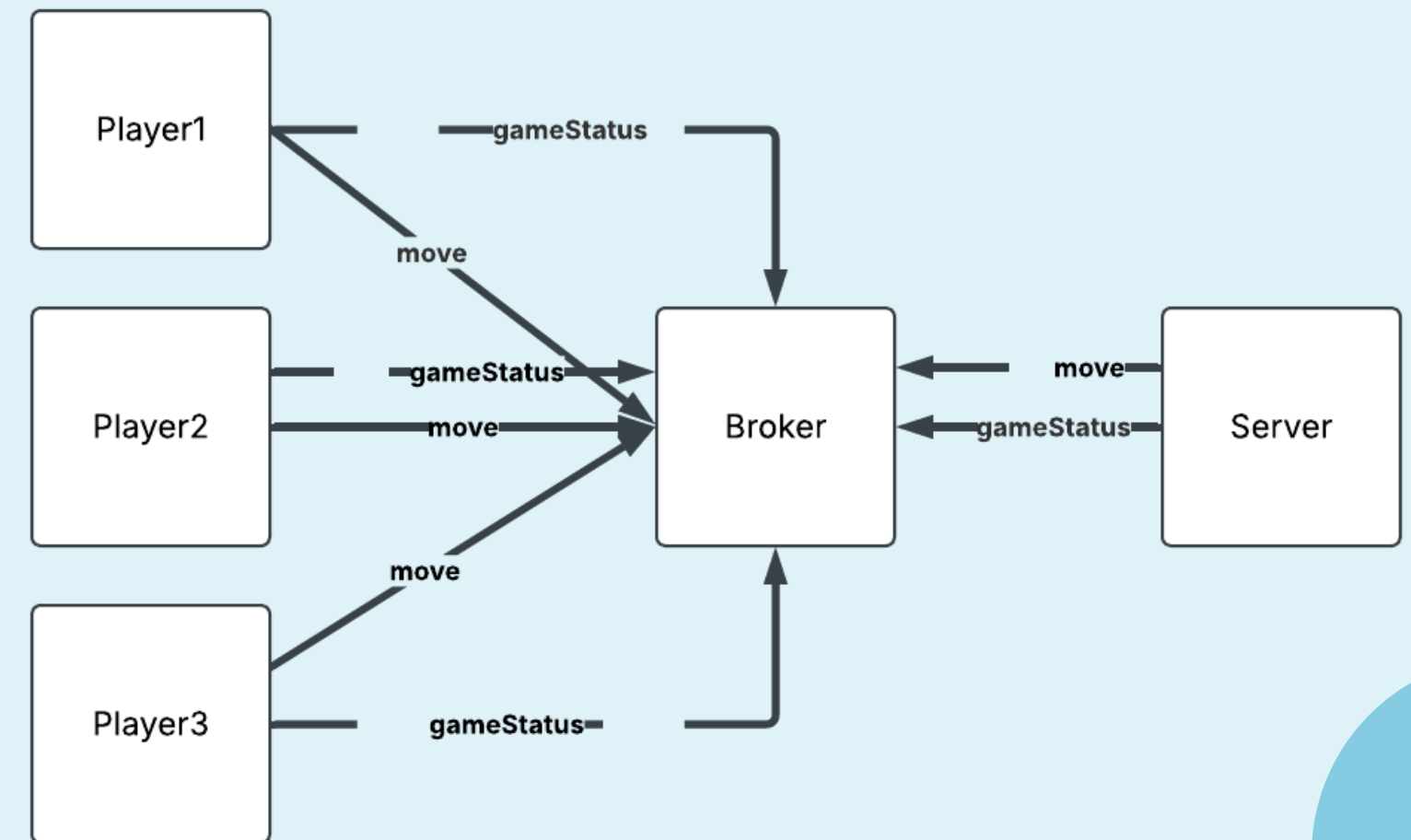
Rise The Stakes or Call Bullsh*t

Loser starts next round with one extra card

ARCHITECTURE & INFRASTRUCTURE

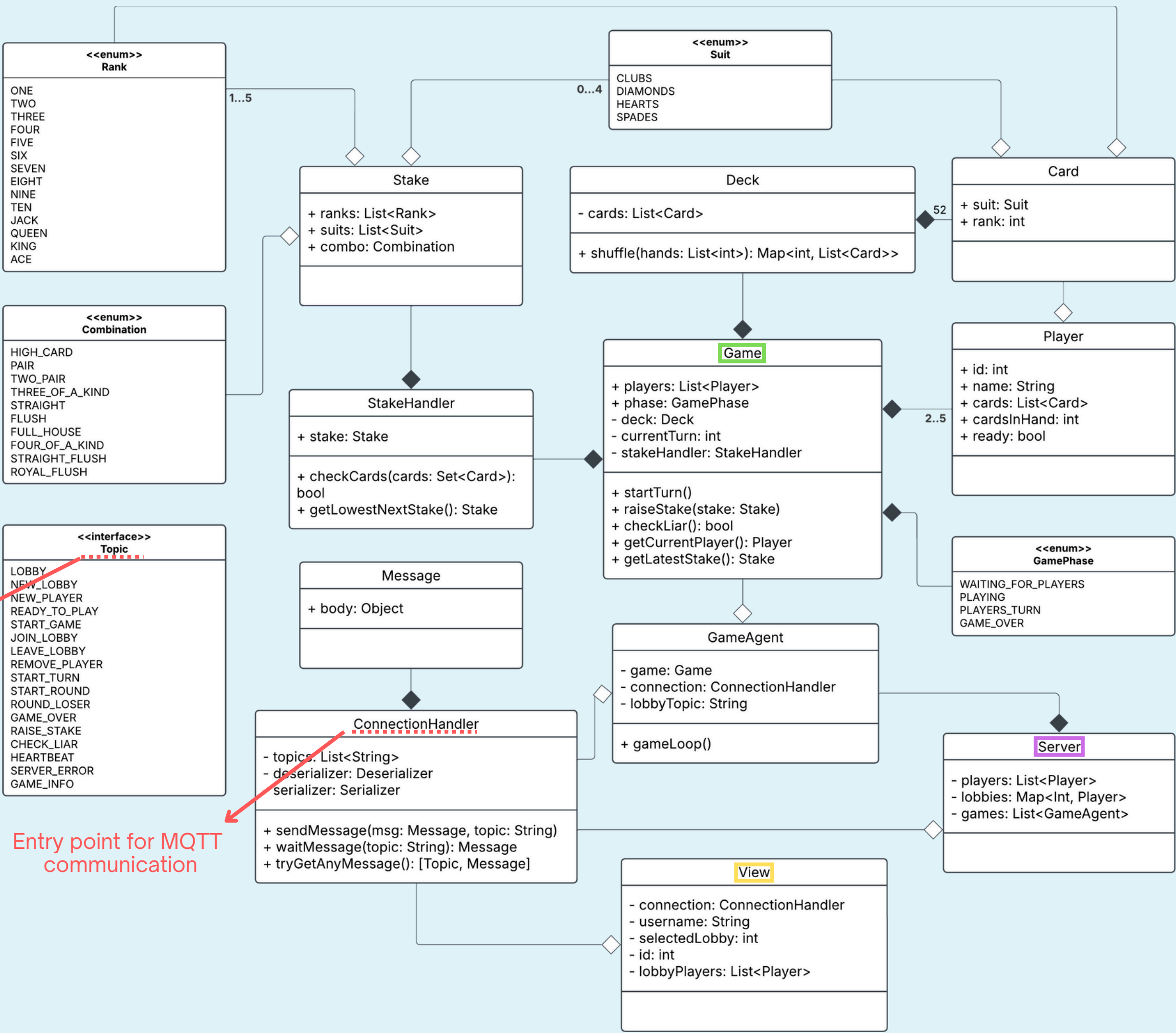
MVC

Client-Server
with
Pub-Sub pattern



MODELLING

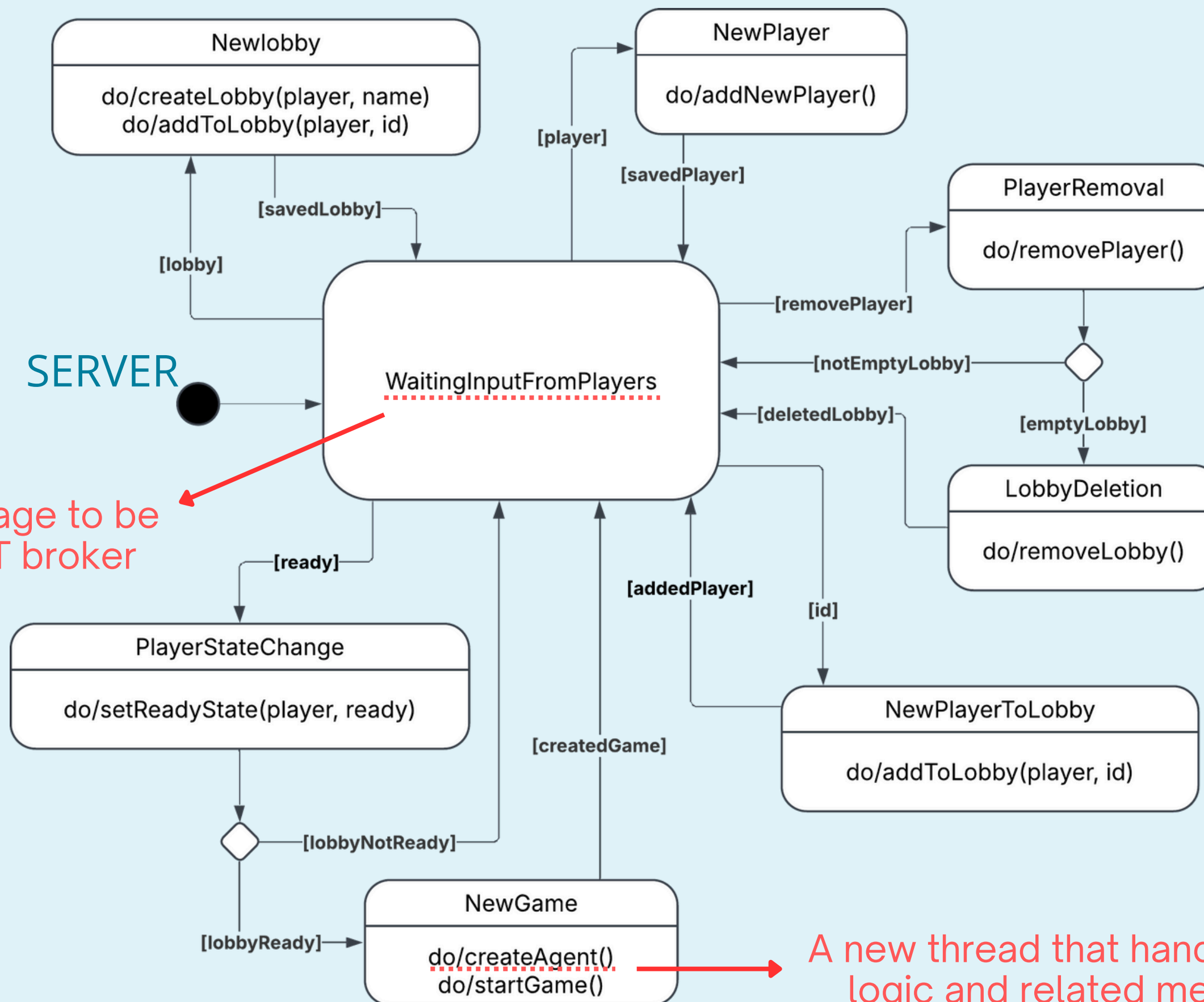
- Model
- View
- Controller



All the topics used across all communications. Can be prefaced by: lobby/<LOBBY ID>/...

Entry point for MQTT communication

BEHAVIOUR



INTERACTION & NETWORK PROTOCOLS

Pub-Sub using MQTT

MQTT topics with path-like structure

ConnectionHandler to manage connection

DATA REPRESENTATION

JSON format

```
""
{
  "body": {
    "stake": {
      "ranks": [
        { "name": "ONE", "$type": "Rank" },
        { "name": "TWO", "$type": "Rank" },
        { "name": "THREE", "$type": "Rank" },
        { "name": "FOUR", "$type": "Rank" },
        { "name": "FIVE", "$type": "Rank" }
      ],
      "suits": [
        { "name": "SPADES", "$type": "Suit" }
      ],
      "combo": {
        "name": "STRAIGHT_FLUSH",
        "$type": "Combination"
      },
      "$type": "Stake"
    },
    "$type": "Message"
  }
}
""
```

Serializer

Message
Object



JSON string

Deserializer

Message
Object



JSON string

adding type tag in serialization to be
used during deserialization for choosing
the correct object creation method



FAULT TOLERANCE

FAULT TOLERANCE BROKER

Redundancy

Bridge
connection

Always passing
through
ConnectionHandler

```
# PRIMARY CONFIGURATION  
listener 1883  
allow_anonymous true
```

```
connection backup-broker  
address 127.0.0.1:1884  
topic # both 0
```

```
# BACKUP CONFIGURATION  
listener 1884  
allow_anonymous true
```

FAULT TOLERANCE SERVER

Using redundancy a backup server is launched alongside the main one, only listening to incoming messages



if the backup server doesn't receive an **heartbeat** message every 2 seconds then promotes itself to the primary server

A copy of the current game state is kept in the secondary server, only leaving out information about ongoing games

FAULT TOLERANCE CLIENT

The system is designed to hide eventual client failures or disconnections



when it is a player's turn if the server doesn't receive any message within 1 minute, he eliminates the player and goes on with the match

The browser saves user's information in order to reconnect to the game in case of page reloading

VIDEO EXECUTION



VIDEO FAULT TOLERANCE





**THANK YOU
FOR THE
ATTENTION**