

Communication protocol

All the messages exchanged between the server and clients are coded in JSON and interpreted as events.

A ping message is sent asynchronously with the messages.

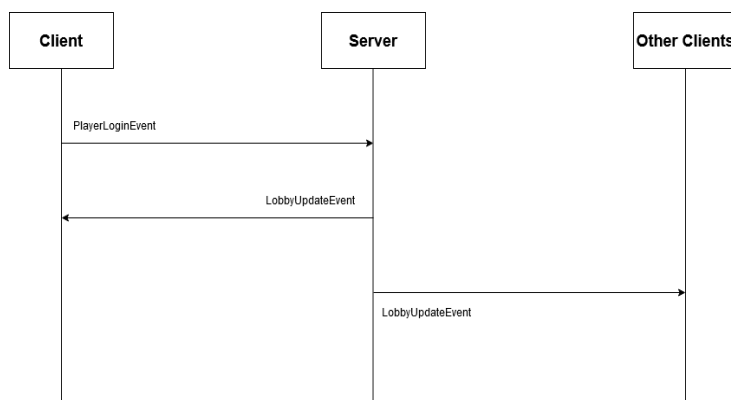
All JSON messages use the following scheme:

```
{
  "event": "EventName",
  "attributes":
  {
    // class attributes
  }
}
```

The communication can be divided in 9 different phases (described below)

Initialization

Login phase



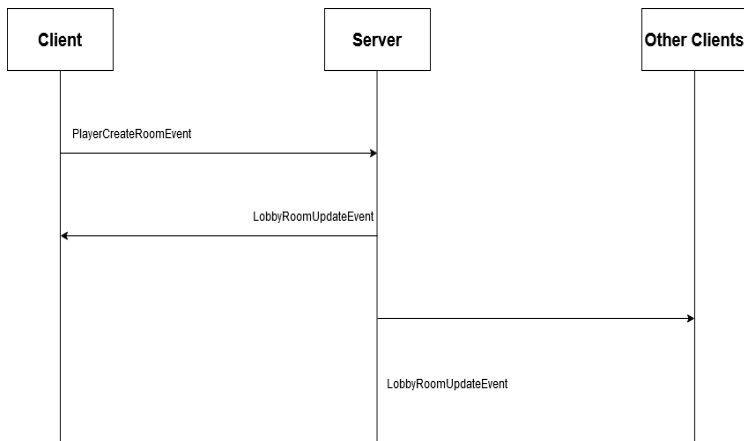
PlayerLoginEvent

```
{
  "event": "PlayerLoginEvent",
  "attributes":
  {
    player : player,
    age: age,
  }
}
```

LobbyUpdateEvent

```
{
  "event": "LobbyUpdateEvent",
  "attributes":
  {
    List of rooms,
  }
}
```

Create Room Phase



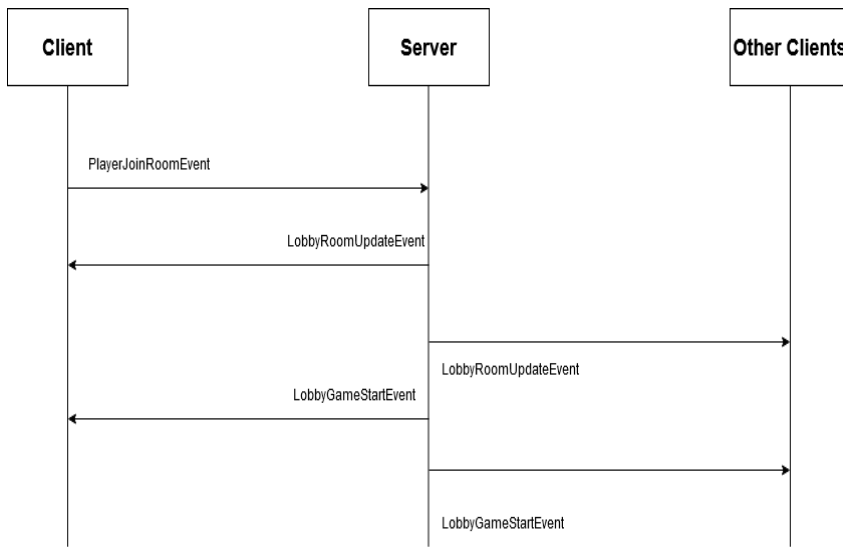
PlayerCreateRoomEvent

```
{
  "event" : "PlayerCreateRoomEvent",
  "attributes":
  {
    player : player,
    maxPlayers : maxPlayers
    simplegame : simplegame(Y/N)
  }
}
```

LobbyRoomUpdateEvent

```
{
  "event" : "LobbyRoomUpdateEvent",
  "attributes":
  {
    roomInfo : roomInfo, (Object that contains the room information)
  }
}
```

Join room phase



The game starts only if the max number of player is reached

PlayerJoinRoomEvent

```
{
  "event" : "PlayerJoinRoomEvent",
  "attributes" :
  {
    player : player,
    roomOwner: roomOwner,
  }
}
```

LobbyRoomUpdateEvent

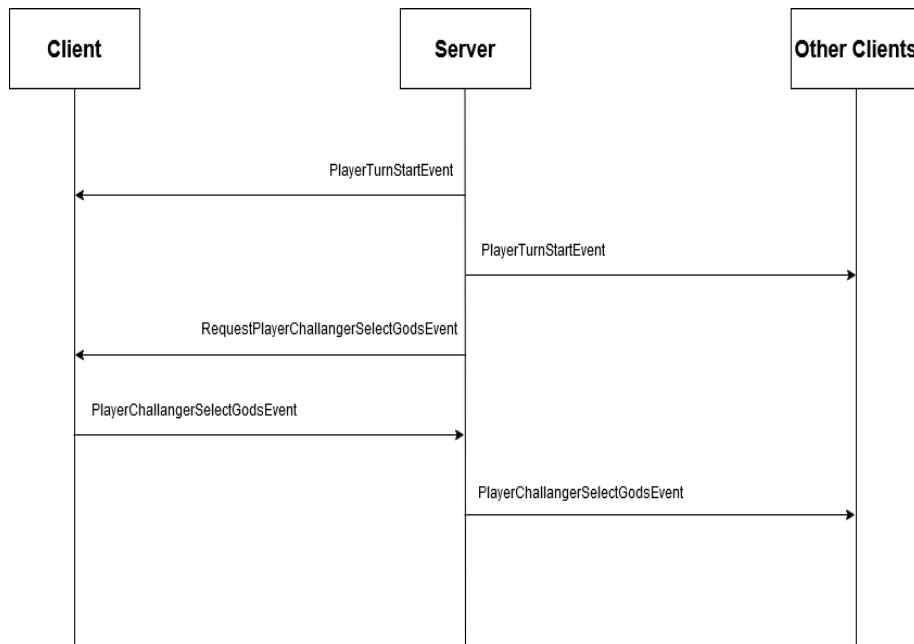
```
{
  "event" : "LobbyRoomUpdateEvent",
  "attributes":
  {
    roomInfo : roomInfo, (Object that contains the room information)
  }
}
```

LobbyGameStartEvent

```
{
  "event" : "LobbyGameStartEvent",
  "attributes" :
  {
    player : player,
    roomInfo : roomInfo,
  }
}
```

Game Phase

Challenger Select Gods Phase (optional)



PlayerTurnStartEvent

```
{
  "event" : "PlayerTurnStartEvent",
  "attributes":
  {
    player : player,
  }
}
```

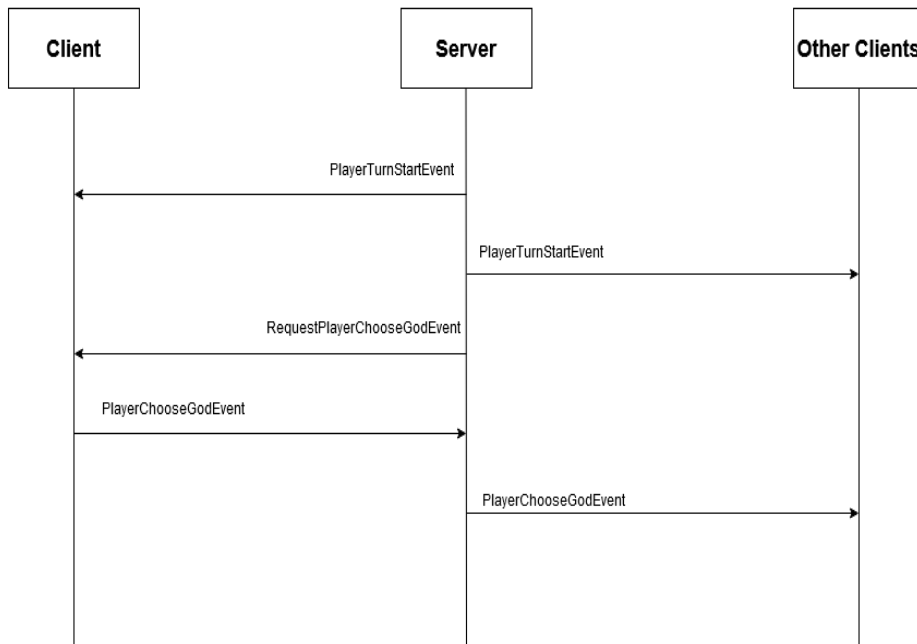
RequestPlayerChallangerSelectGodsEvent

```
{
  "event" : "RequestPlayerChallangerSelectGodsEvent"
  "attributes" :
  {
    player : player,
    List of gods,
    selcectedGodsCount (the number of gods to select),
  }
}
```

PlayerChallangerSelectGodsEvent

```
{
  "event" : "PlayerChallangerSelectGodsEvent",
  "attributes" :
  {
    List of gods (the selected gods)
  }
}
```

Select gods phase (optional)



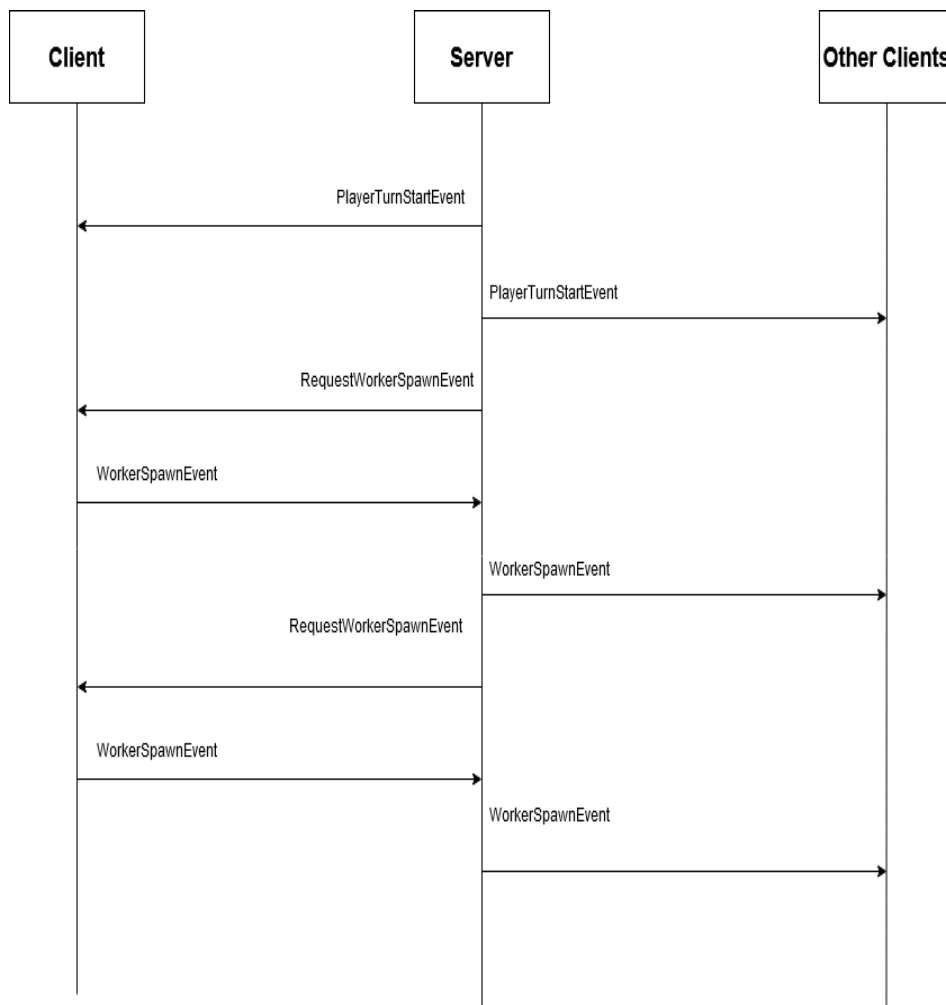
RequestPlayerChooseGodEvent

```
{
  "event" : "RequestPlayerChooseEvent",
  "attributes" :
  {
    player : player,
    List of avaiable gods,
  }
}
```

PlayerChooseGodEvent

```
{
  "event" : "PlayerChooseGodEvent",
  "attributes" :
  {
    player : player,
    god : god (the selected god),
  }
}
```

Place worker phase



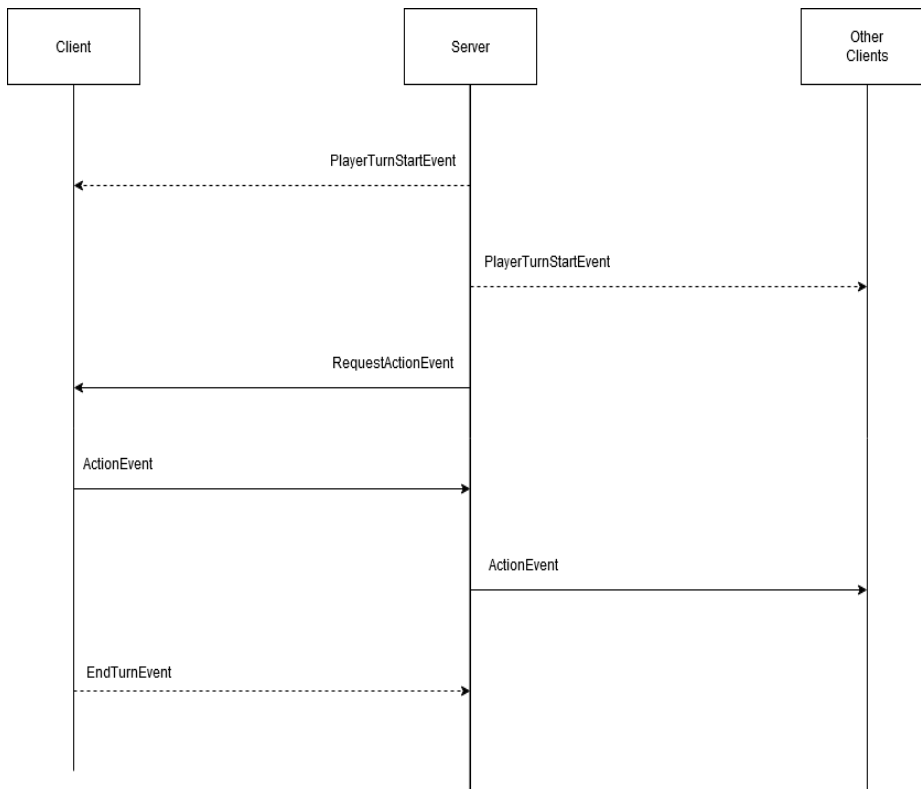
RequestWorkerSpawEvent

```
{
  "event" : "RquestWorkerSpawnEvent",
  "attributes" :
  {
    player : player,
    List of avaiable positions,
  }
}
```

WorkerSpawnEvent

```
{
  "event" : "WorkerSpawnEvent",
  "attributes" :
  {
    player : player,
    id : id (the worker id),
    position : position (coordinates),
  }
}
```

Action phase



This phase is the same for Move, BuildBlock, BuildDome and Force.

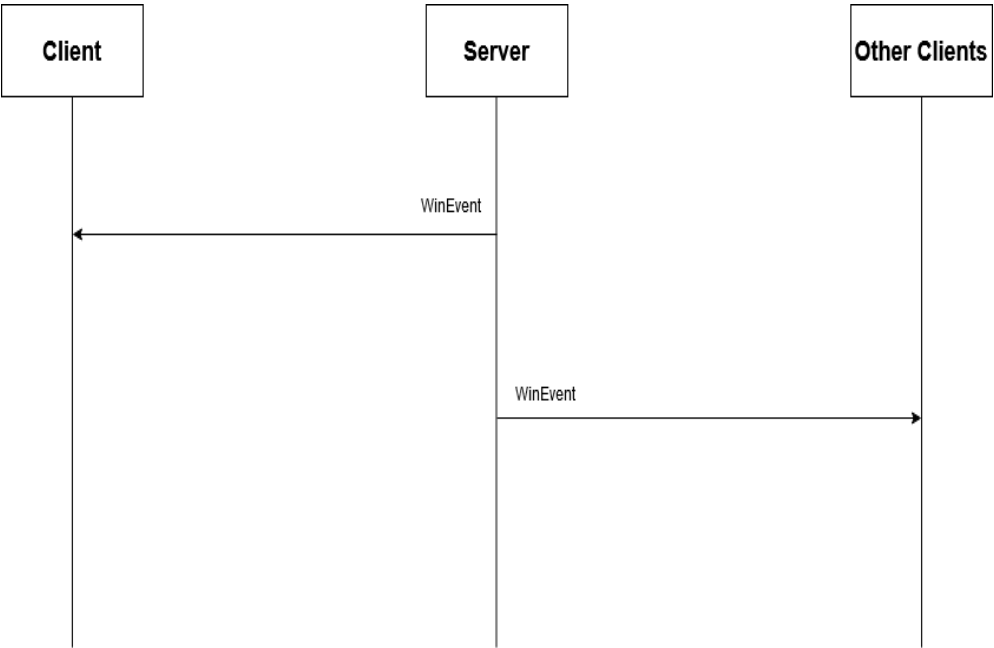
RequestActionEvent

```
{
  "event" : "RequestActionEvent",
  "attributes" :
  {
    player : player,
    worker : worker
    List of available target (only in the force move),
    List of available destinations,
  }
}
```

ActionEvent

```
{
  "event" : "ActionEvent",
  "attributes" :
  {
    player : player,
    worker : worker;
    target : target (only in the force move),
    destination : destination,
  }
}
```

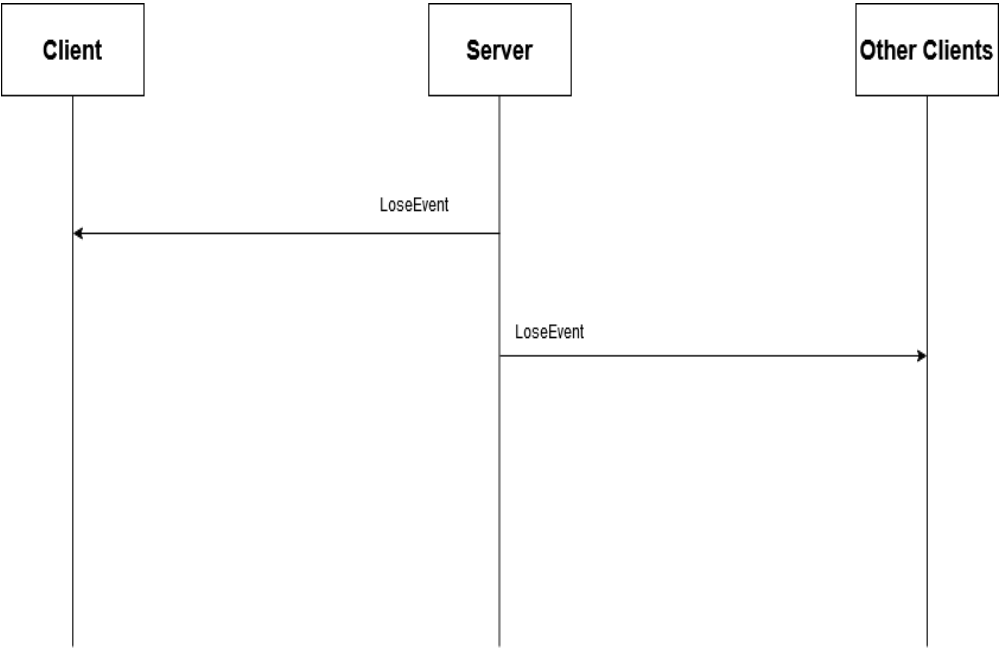
Win phase



WinEvent

```
{
  "event" : "WinEvent",
  "attributes" :
  {
    player : player,
  }
}
```


Lose phase



LoseEvent

```
{
  "event" : "LoseEvent",
  "attributes" :
  {
    player : player,
  }
}
```