

Cluedo

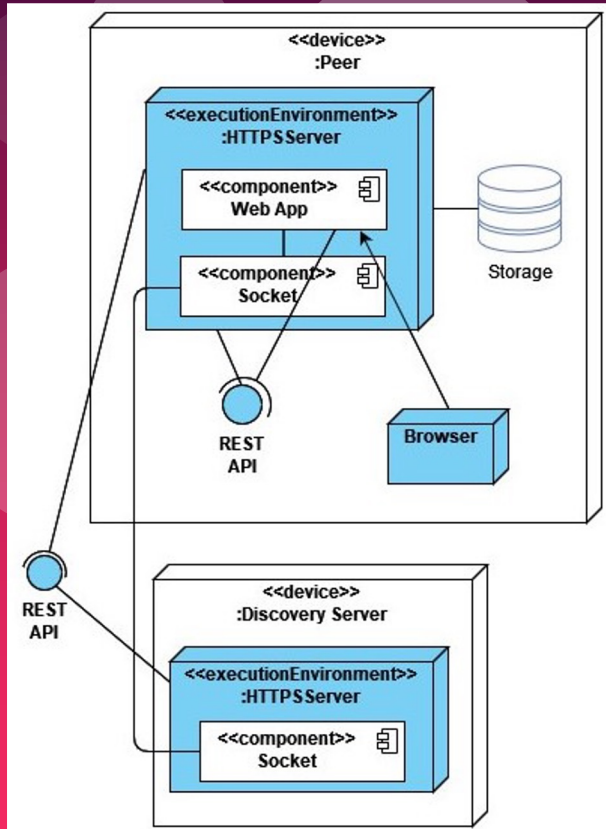
Sistemi Distribuiti

Marica Pasquali

0000982560

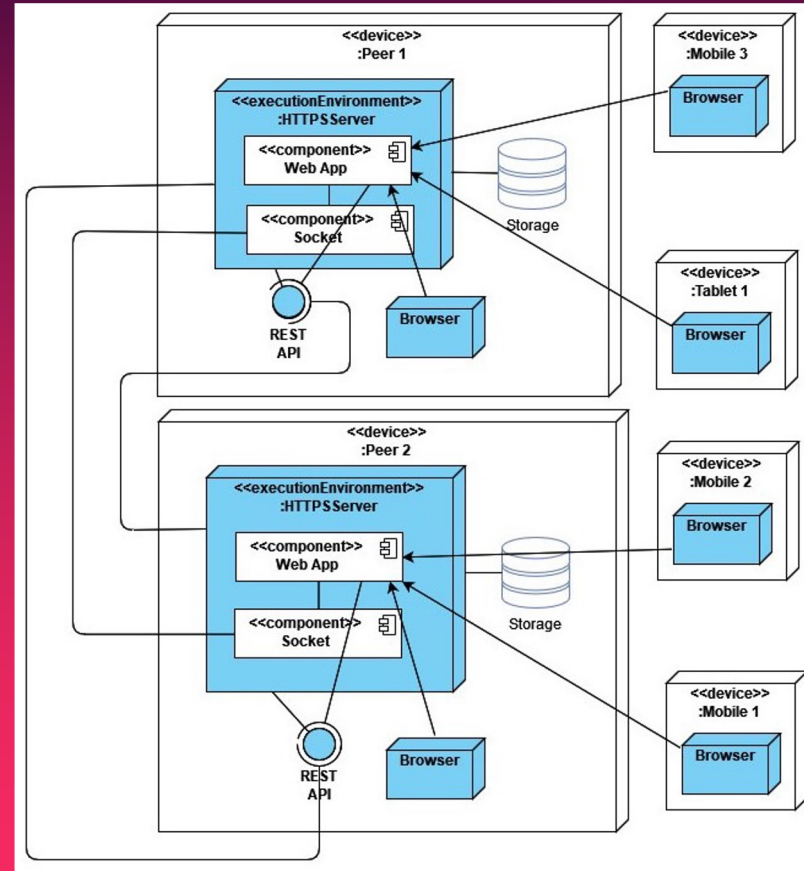
(marica.pasquali@studio.unibo.it)

Design – Struttura

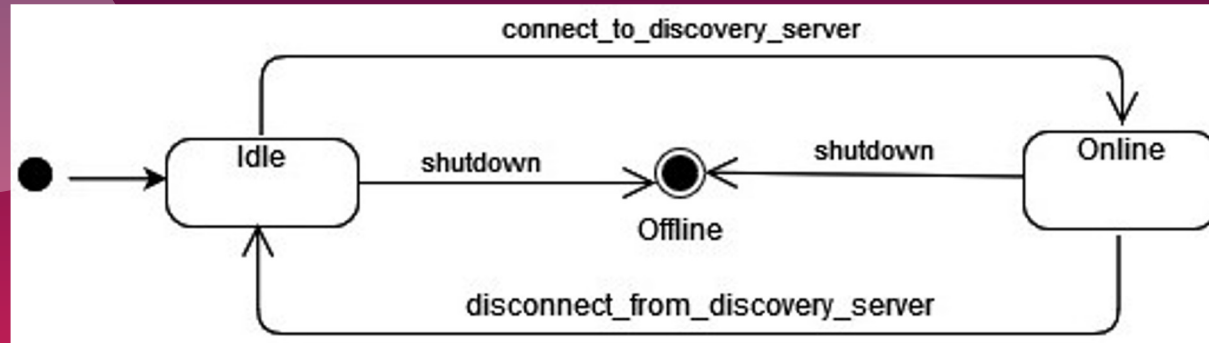


Peer (P)
implementata la
business logic del gioco
e ha la funzione di
backup server

Discovery Server (DS)
permette di “scoprire” gli
indirizzi dei peer in rete



Design – Comportamento (punto di vista del Peer)



STATI

- **Idle**: stato in cui è possibile utilizzare l'applicativo come *architettura Client-Server*
- **Online**: stato in cui è possibile raggiungere il Peer attraverso la rete (P2P)
- **Offline**: stato in cui non è possibile raggiungere il Peer tramite la rete

Design – Comportamento (punto di vista di una partita/utente)

STATI

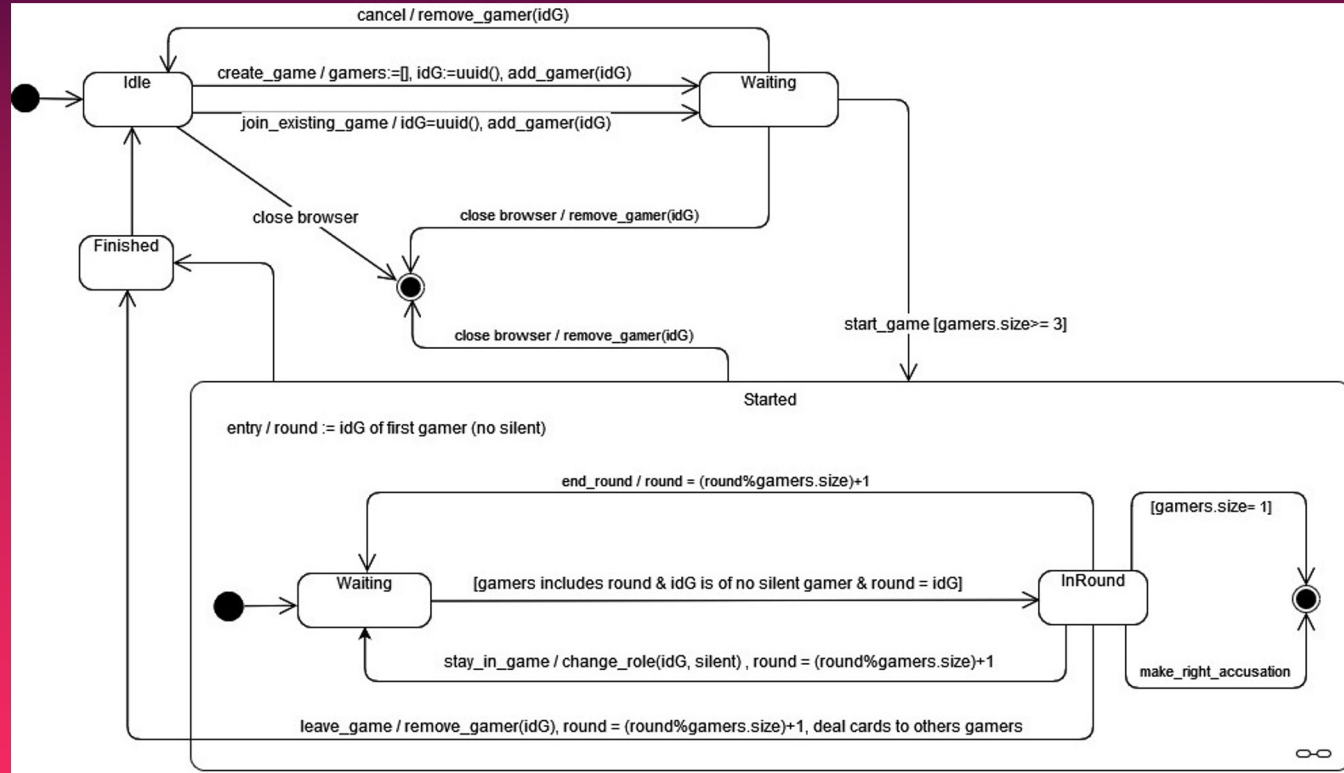
- **Idle**: stato in cui un'utente può interagire con l'applicativo tramite GUI

Partita/Utente

- **Waiting**: stato in cui l'utente deve aspettare l'entrata di altri giocatori
- **Started**: stato in cui è possibile incominciare a giocare la partita
- **Finished**: stato in cui la partita è finita (stato di transizione momentanea)

Giocatore in partita avviata

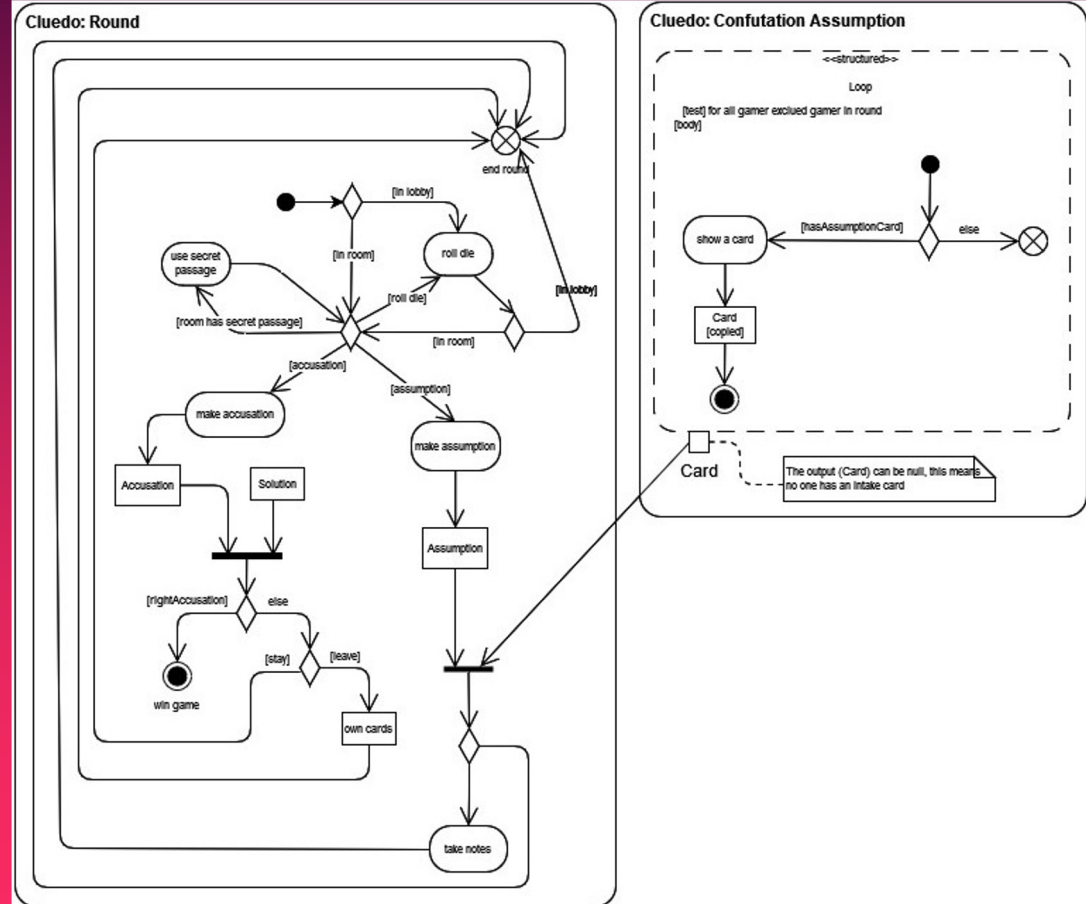
- **Waiting**: stato in cui si deve aspettare il proprio turno (ruolo utente: *participant*) oppure aspetta che la partita finisca (caso ruolo utente: *silent*)
- **InRound**: stato in cui il giocatore può iniziare a giocare



Design – Comportamento (punto di vista dei giocatori)

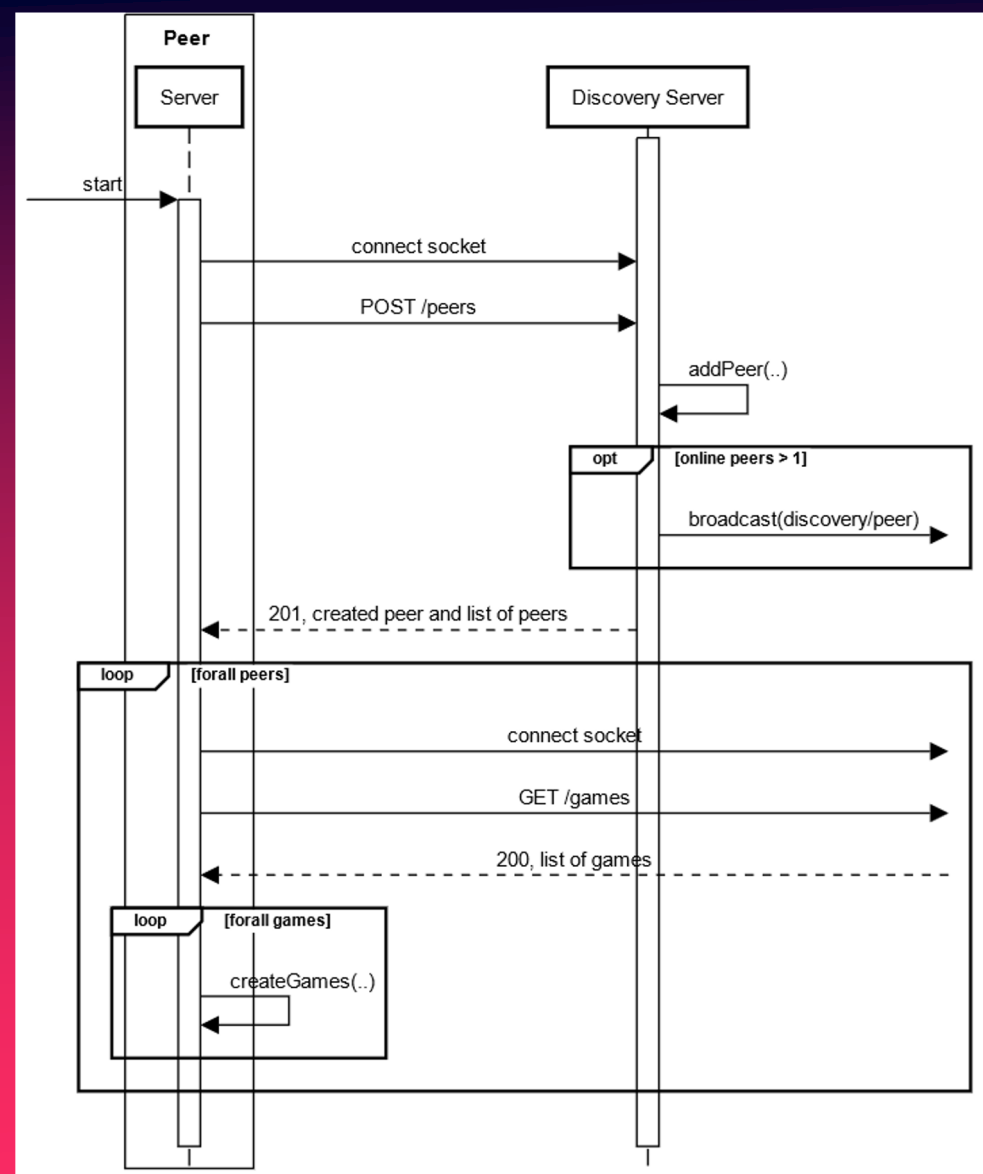
Durante il proprio turno, un giocatore può effettuare diverse attività/azioni.

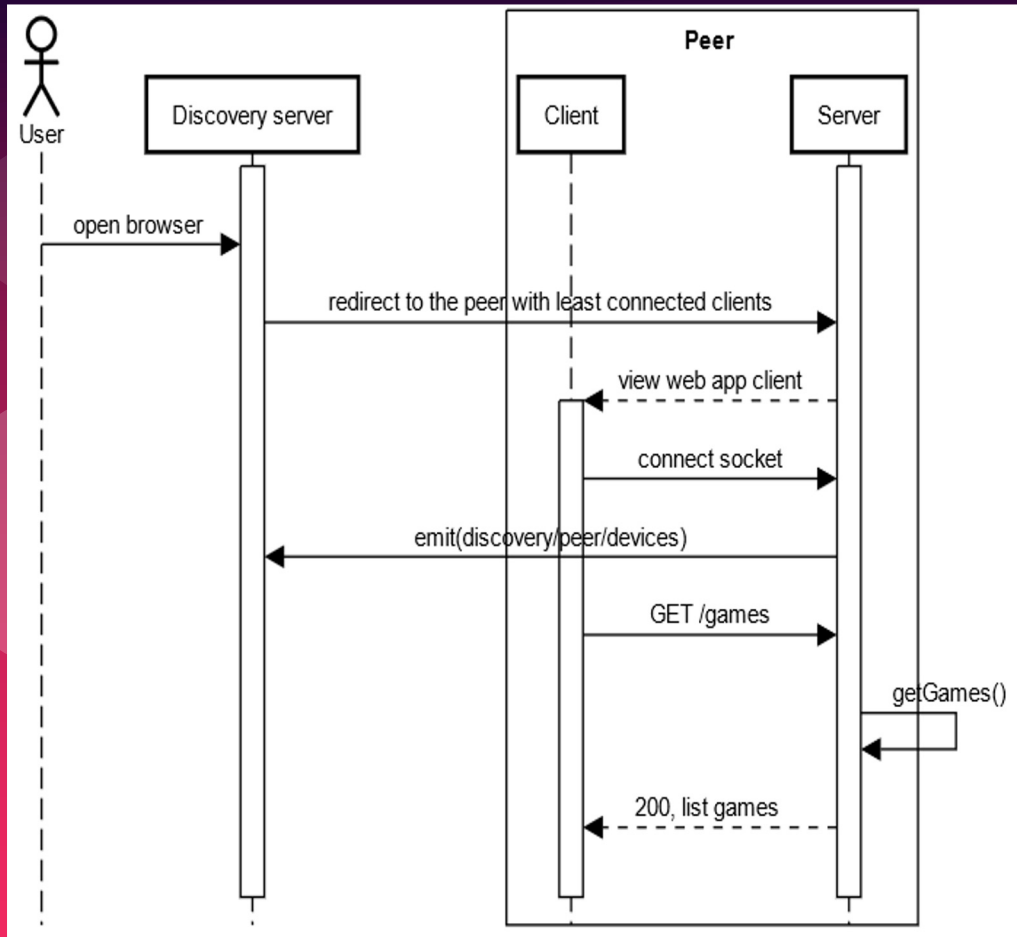
Attività critica: ‘fare un’ipotesi (*make assumption*)’ perché coinvolge anche gli altri giocatori attraverso la *confutazione dell’ipotesi*.



Design - *Interazione* (1)

- 1) Un Peer (P), in fase di avvio, cerca di stabilire una connessione socket con il DS.
- 2) Un Peer (P) dichiara la sua esistenza al Discovery Server (DS) attraverso una chiamata POST (endpoint: /peers).
- 3) Discovery Server elabora la richiesta e invia al Peer la lista degli altri peer in rete.
- 4) Il Peer stabilisce una connessione socket con gli altri Peer e richiede tutte le partite memorizzate tramite una chiamata GET (endpoint: /games),

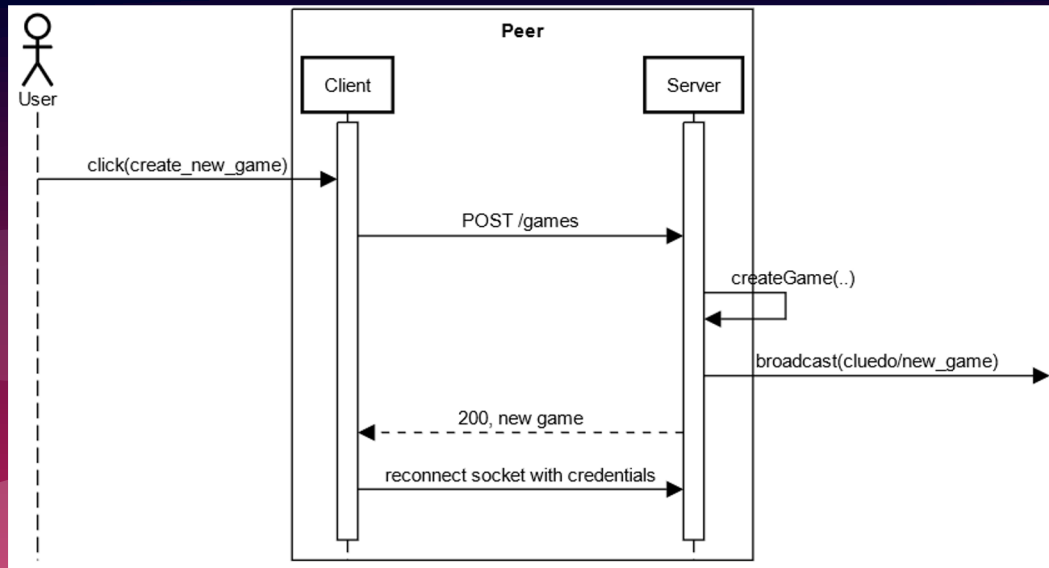




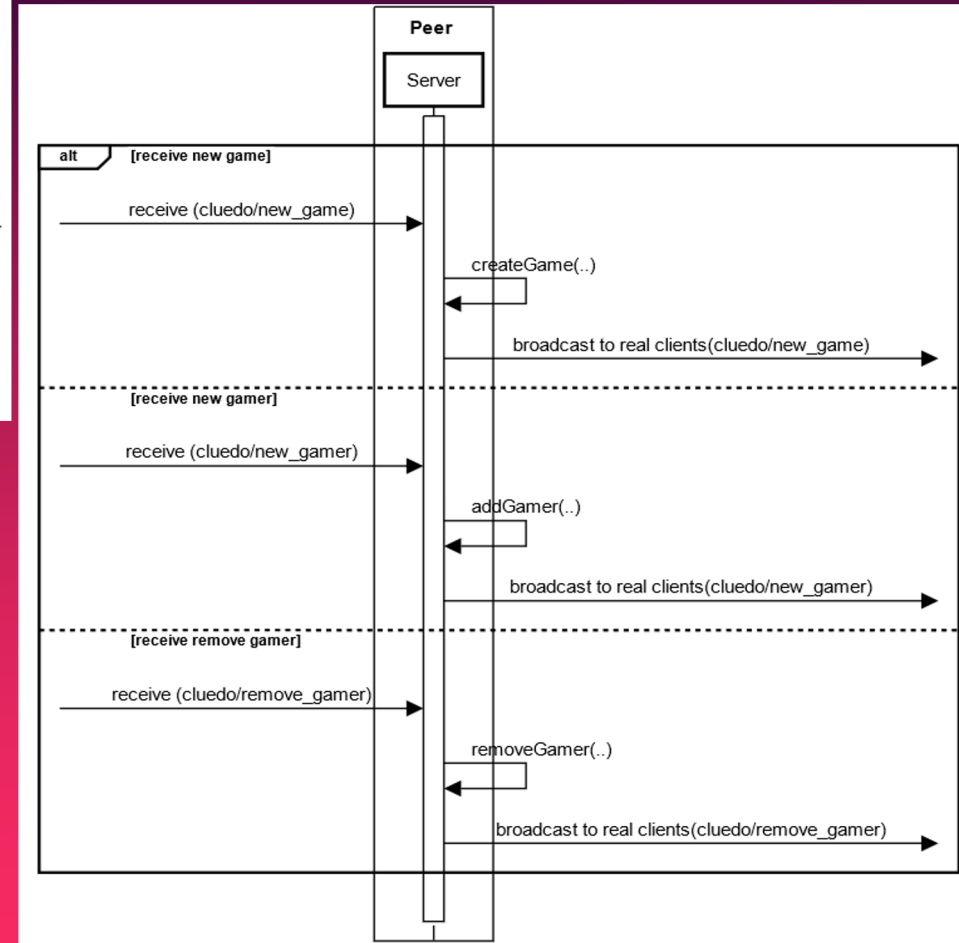
Design - *Interazione* (2)

→ **Discovery Server = load balancer**

→ **Peer Server = message broker**



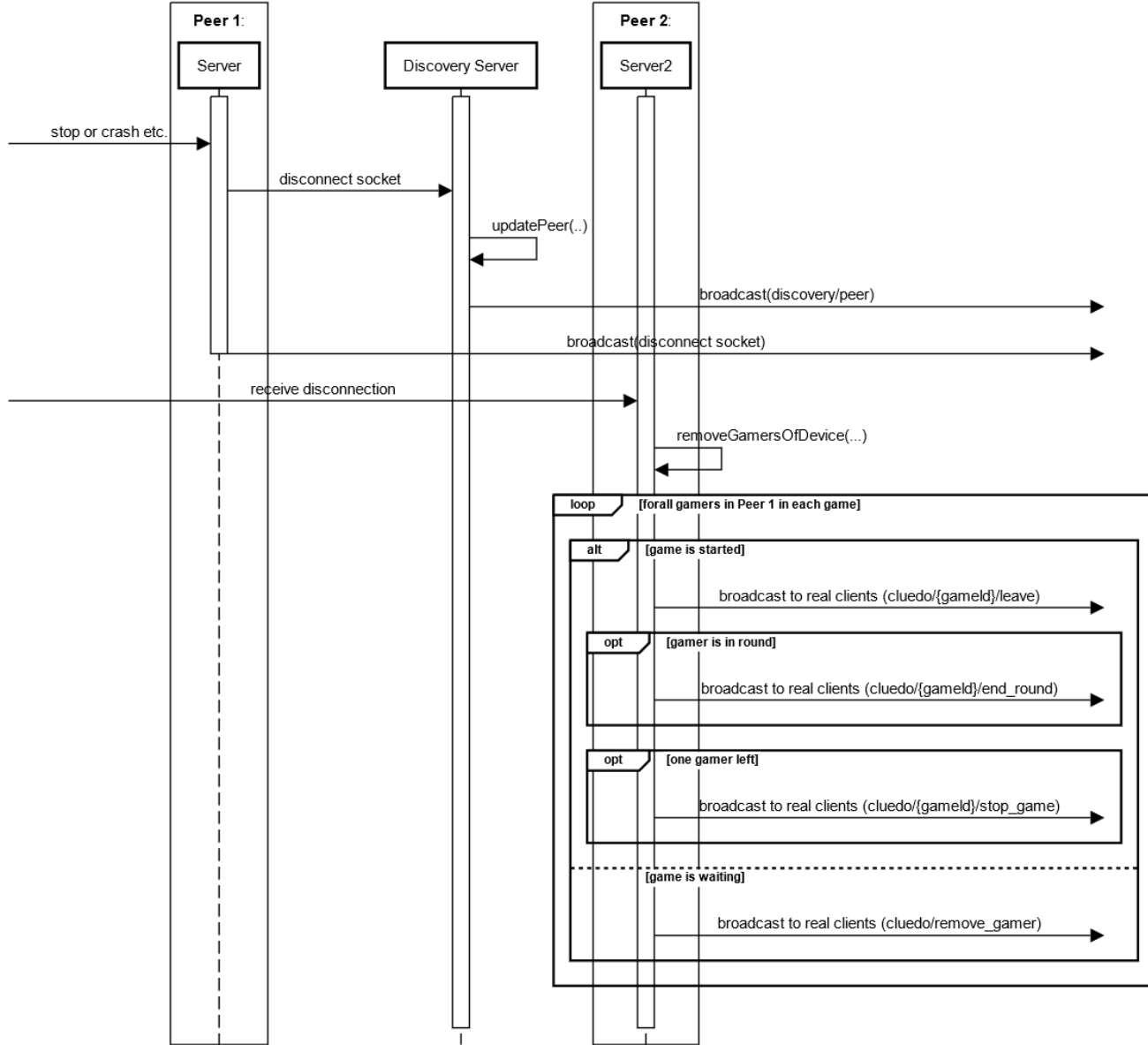
Design - *Interazione* (3)



Pattern di comunicazione P2P

- 1) Utente interagisce con la GUI → *invio richiesta REST al server side del Peer*
- 2) Peer (server) elabora la richiesta e invia in *broadcast tramite evento socket* le modifiche a tutti gli altri Peer connessi e successivamente *termina la richiesta REST*
- 3) Quando un Peer (server) riceve un evento socket, elabora le modifiche [e le invia in broadcast a [tutti] i client ad esso collegati (sempre tramite connessione socket)]

Design - *Interazione (4)*



Per mantenere la consistenza, quando un Peer termina (gracefully or crashed), tramite le connessioni socket viene intercettato e gestito in maniera adeguata.

Tecnologie

- *Node.JS + Express + Vue.JS*: Server + RESTFul API + Interfaccia Grafica
- *MongoDB*: storage orientato a documenti
- ***socket.io*: framework che utilizza Web Socket per la comunicazione real-time**
 - sceglie automaticamente la migliore opzione disponibile tra *HTTP long-polling*, *Web Socket* e *Web Transport*
 - meccanismo heartbeat (controllo periodico dello stato della connessione)
- *Docker*: simulare il deploy del sistema in production mode
- *Mocha*: framework per il testing. Usato un approccio Test-Driven Development (TDD).

Conclusioni

Possibili sviluppi futuri

- Miglioramento interfaccia grafica
- Chat o video chiamata per la comunicazione diretta tra gli utenti durante una partita o nella waiting room
- Verificare l'efficienza e il corretto funzionamento da parte dell'utilizzo intensivo di numerosi utenti
- Comunicazione tra reti differenti

Demo



The background is a vertical gradient from dark purple at the top to bright pink at the bottom. Several semi-transparent circles of various sizes are scattered across the frame, some overlapping each other.

GRAZIE PER L'ATTENZIONE