



Aplicație Singleplayer-Multiplayer de Șah

Absolvent: Căpățîna Răzvan Nicolae

Coordonator științific: Conf. dr. Rusu Cristian




Lucrarea are 2 scopuri principale:

- dezvoltarea unei aplicații de la zero ce permite utilizatorului să joace șah
- implementarea în cadrul aplicației a unui agent care să joace șah cu o performanță cât mai mare posibilă, fiind limitat de constrângerile hardware



2 moduri de joc:

- singleplayer, contra unui agent implementat în cadrul aplicației
- multiplayer, contra unui alt utilizator



Detalii implementare:


- proiect scris în C++, compilat pe 64-bit
- elementele grafice realizate cu OpenGL (afișarea butoanelor, pieselor și textului)
- biblioteca ENet pentru comunicarea Client-Server (necesară în cadrul sesiunii de multiplayer)
- arhitectură modulară, se folosesc principii ale programării orientate pe obiecte: moștenire, polimorfism
- șabloane de proiectare: singleton



Competele Aplicației



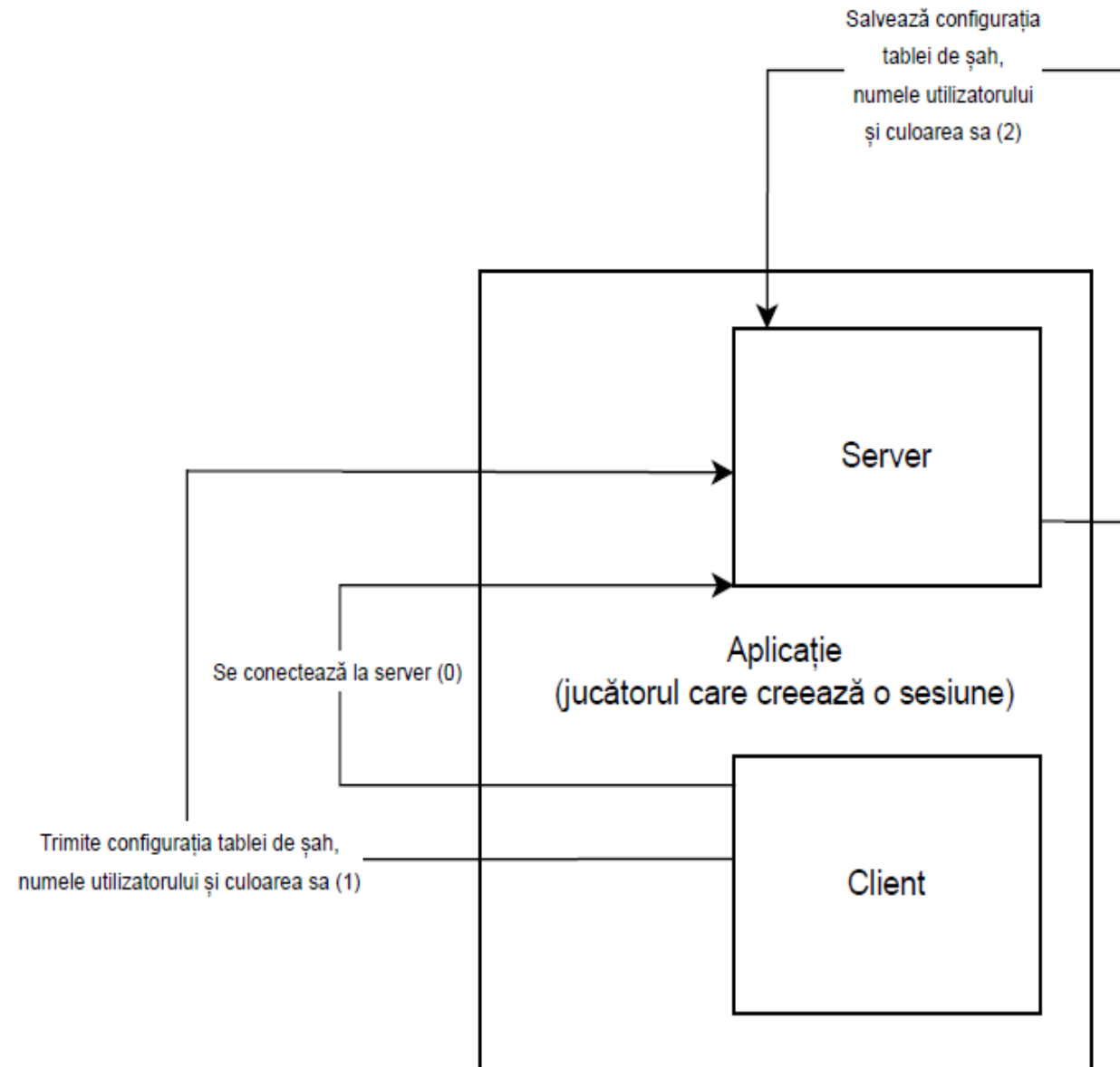
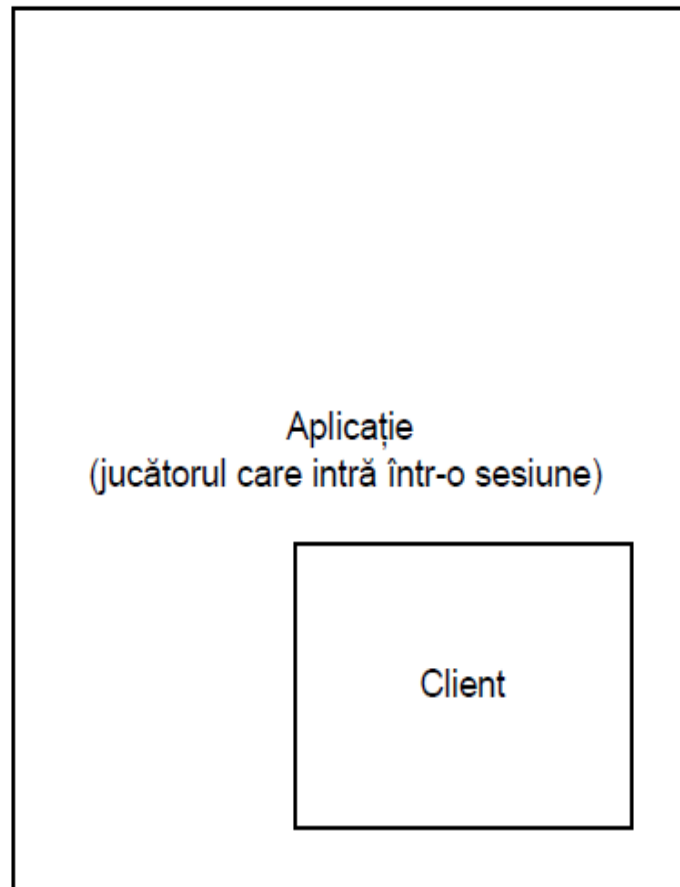
Comunicarea Client-Server



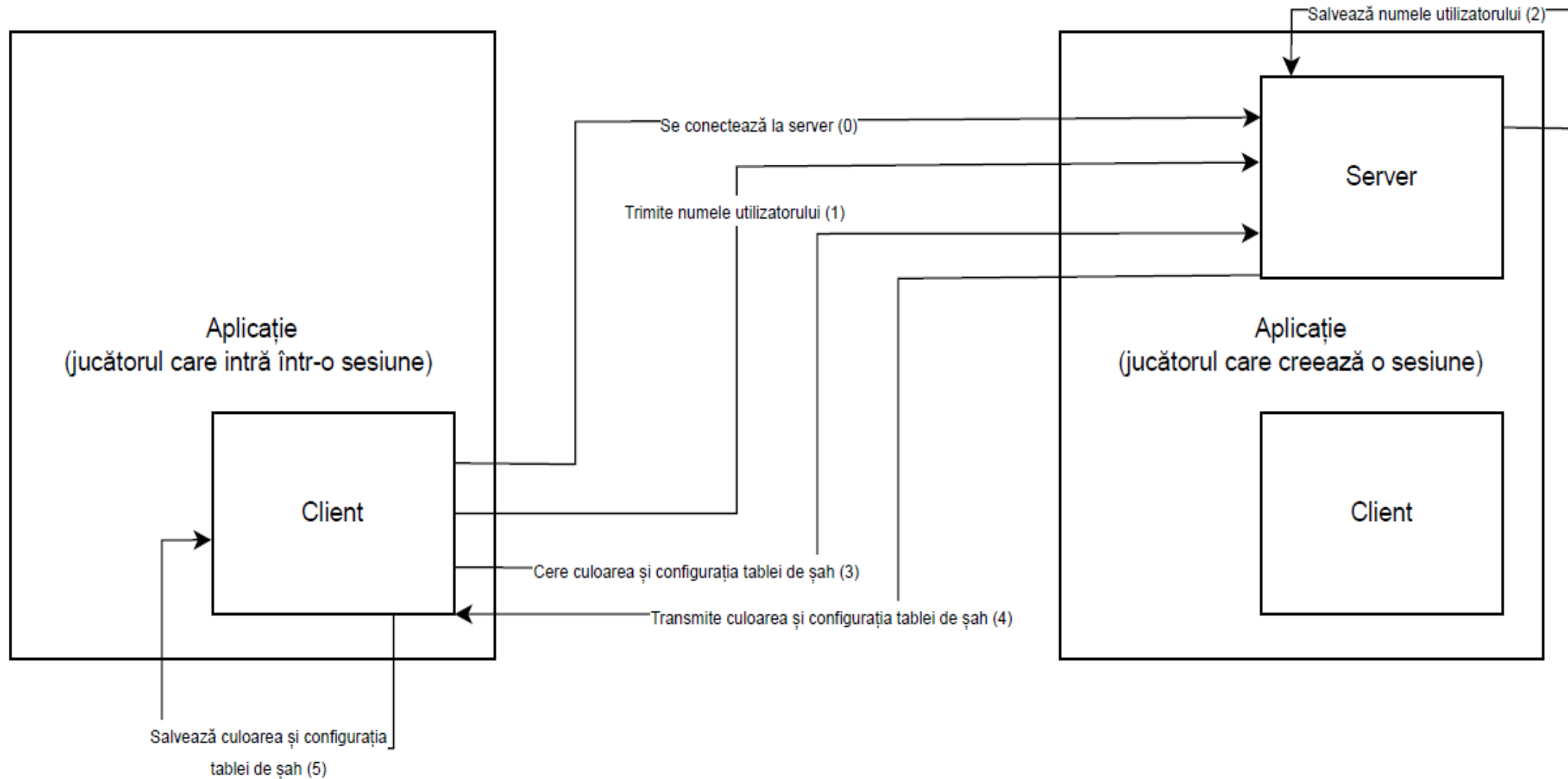
Comunicarea Client-Server

- are loc în timpul unui meci de Multiplayer
- este formată din 2 etape: conectarea (schimbul inițial de informații) și interacțiunea pe parcursul jocului (sistemul de pinging + transmiterea mutărilor de pe tablă)
- se folosește întotdeauna instanța de server a utilizatorului creator al sesiunii de joc
- utilizatorul ce nu a creat sesiunea de joc se va conecta la server-ul celui alt jucător
- prezintă mecanisme de așteptare și de pinging pentru a asigura menținerea conexiunii

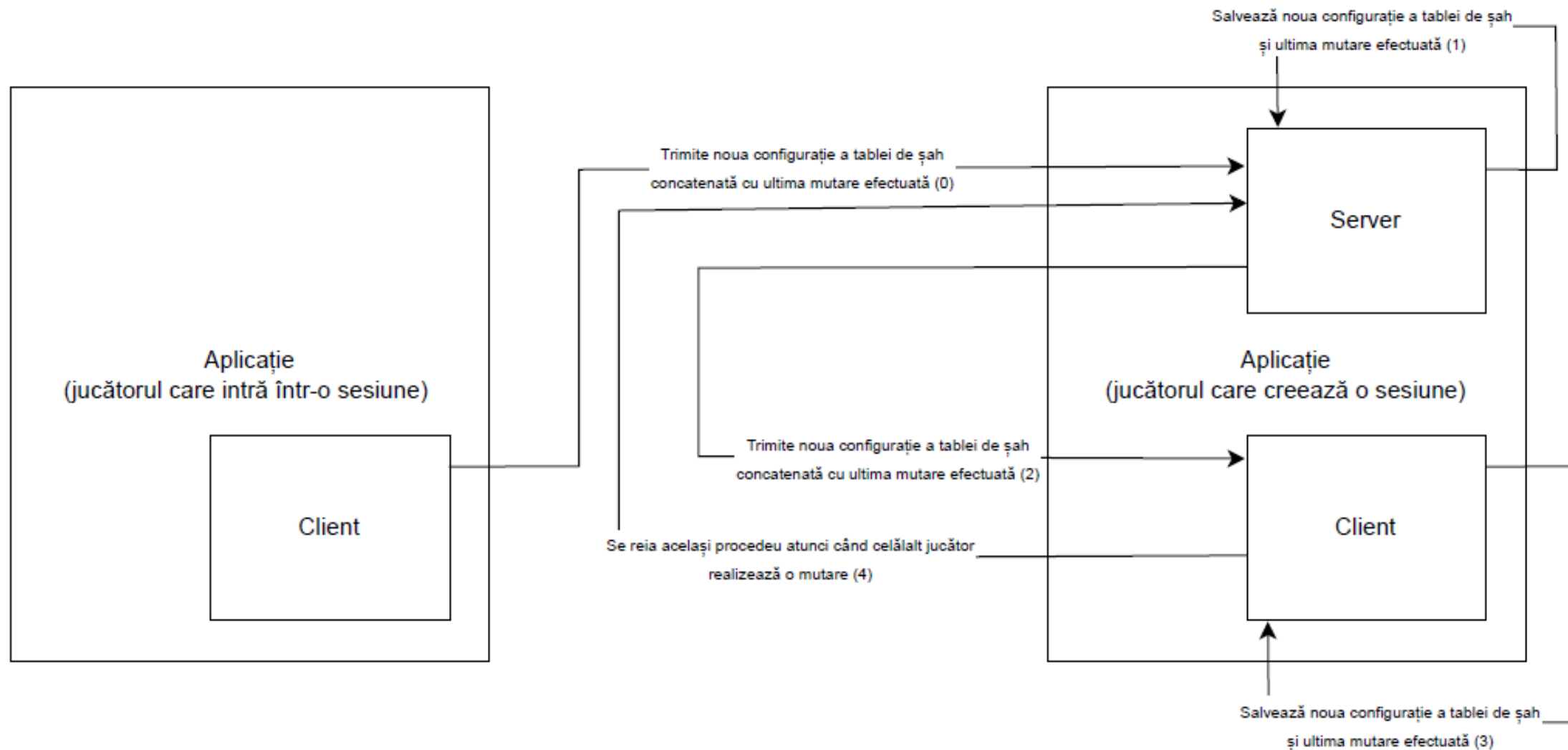
Comunicarea inițială dintre server și jucătorul creator al sesiunii



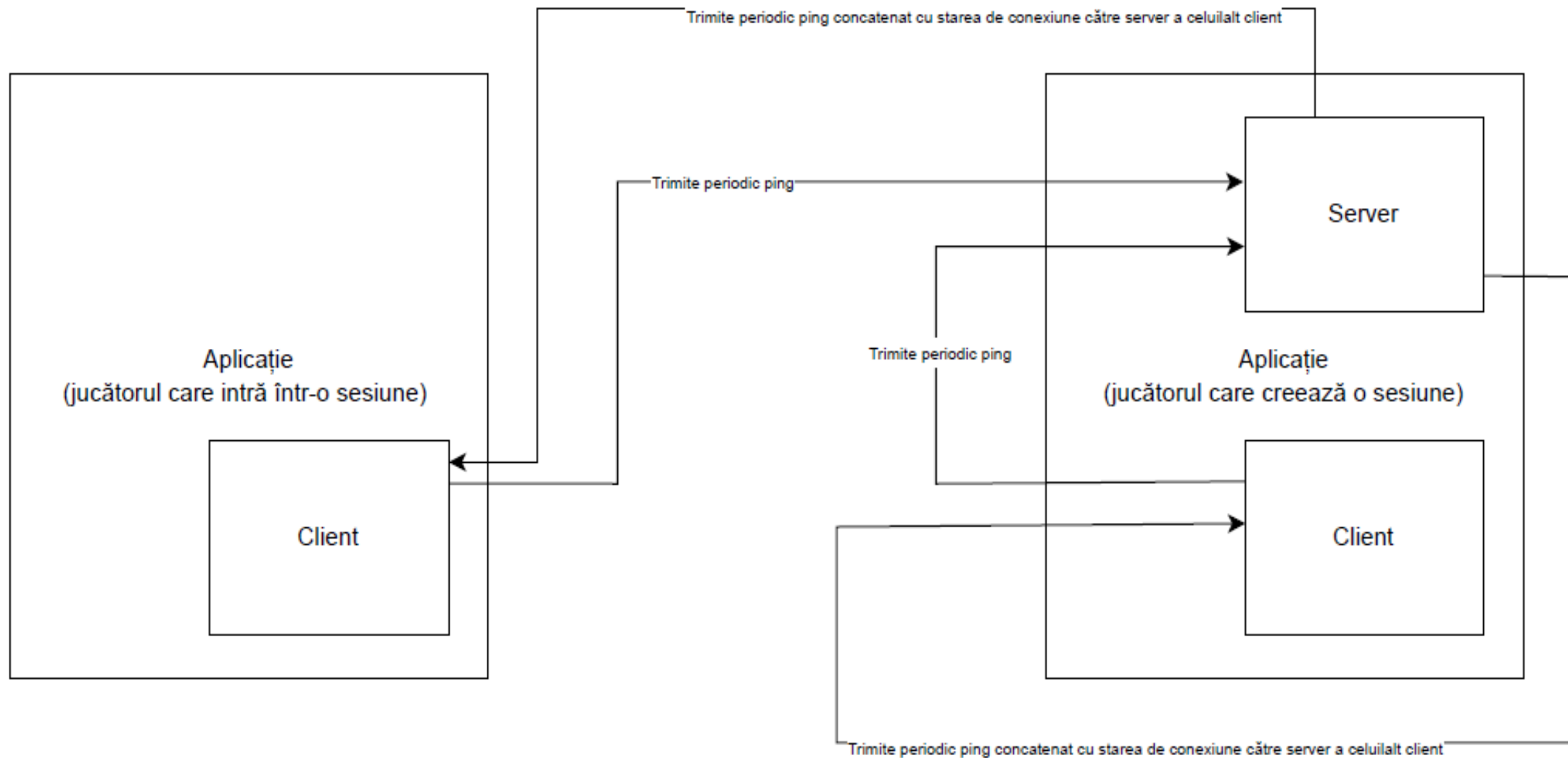
Comunicarea inițială dintre server și jucătorul ce intră în sesiune



Comunicarea dintre server și clienți în timpul jocului



Sistemul de pinging dintre server și clienți





Agentul



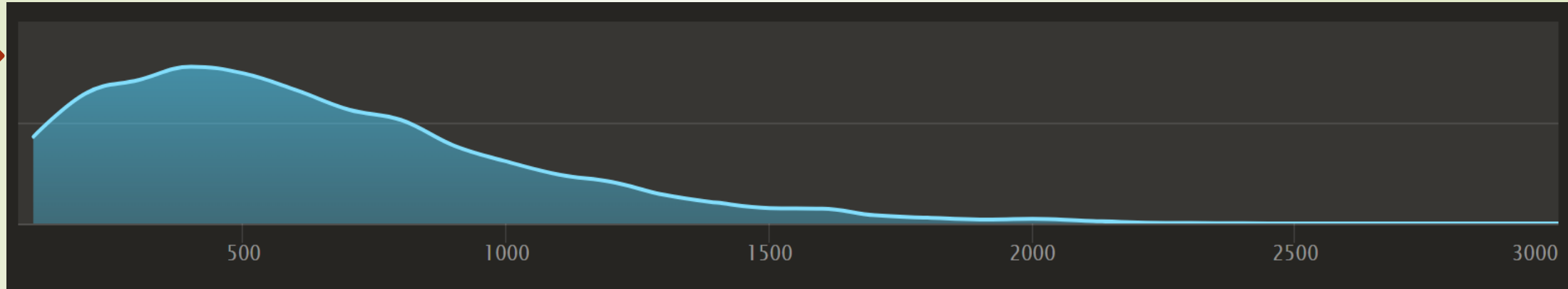
Agentul

- are la bază algoritmul Minimax și optimizarea Alpha-Beta Pruning
- a fost paralelizat, se creează câte un fir de execuție diferit pentru fiecare subarbore al rădăcinii
- folosește Zobrist Hashing pentru a codifica configurațiile tablei
- Zobrist Hashing este apoi folosit pentru a crea un sistem de caching per fir de execuție (memoizarea transpozițiilor)



Rezultate Obținute

Distribuția Rating-urilor pentru meciurile cu limită de 15 minute (Rapid) (Chess.com)



Rating Agent:
~2100-2200

Rating Oponent	Victorii	Înfrângeri	Remize	Total	Rată Victorie
1600 (Chess.com)	7	1	2	10	70%
1800 (Chess.com)	7	2	1	10	70%
2000 (Chess.com)	3	1	1	5	60%
2200 (Chess.com)	2	3	0	5	40%
2300 (Chess.com)	0	2	1	3	0%

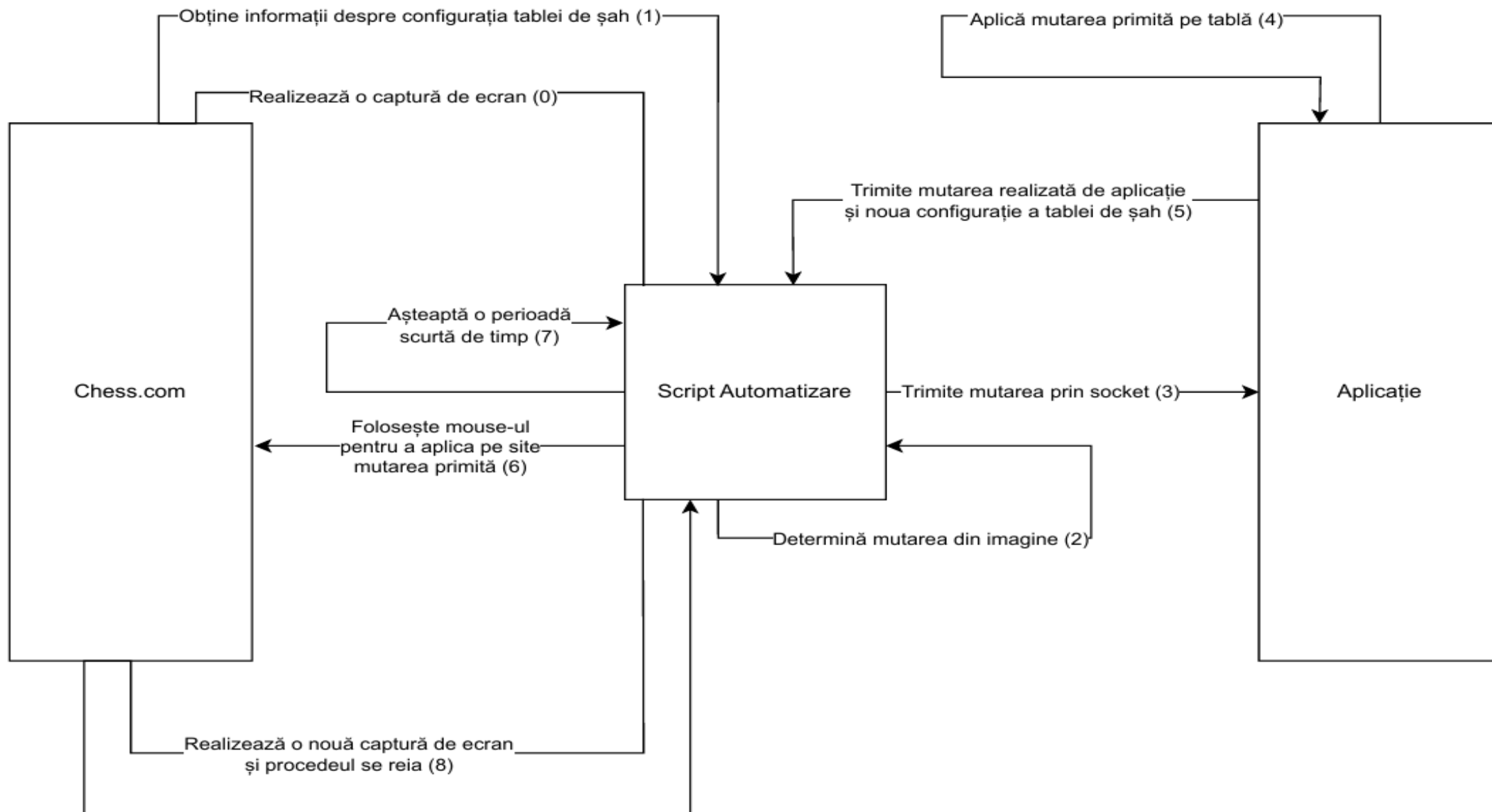
Oponent	Victorii	Înfrângeri	Remize	Total	Rată Victorie
Stockfish nivel 5/8 (Lichess.org)	5	1	1	7	71.4%
Stockfish nivel 6/8 (Lichess.org)	0	4	1	5	0%



Automatizarea Evaluării

- se dorește eliminarea necesității unui intermediar (persoană fizică) în interacțiunea dintre aplicație și site (Chess.com)
- s-a implementat un script în Python care preia această funcție de intermediar
- script-ul folosește modulele NumPy, PyAutoGUI (mișcare automată mouse) și Pillow (prelucrare de imagini) pentru a automatiza evaluarea
- script-ul folosește atât comunicare prin socket-uri, cât și creare de capturi de ecran și identificarea mutărilor realizate folosind prelucrări de imagini și analize asupra histogramelor RGB

Comunicare Script Automatizare





Concluzii:

- aplicația a îmbinat diverse subdomenii ale Informaticii, de la structuri de date eficiente și optimizări pe bits până la grafică, programare paralelă și rețelistică
- respectarea principiilor programării orientate pe obiecte facilitează construcția unei aplicații complexe și modulare
- este posibilă implementarea unui agent de șah performant care să fie în același supus unor constrângeri hardware
- deciziile arhitecturale luate stau la baza performanței agentului, trebuie ținut cont la fiecare decizie de existența tradeoff-ului dintre timp și memorie



Posibile îmbunătățiri:

- înlocuirea funcției euristice din frunzele arborelui Minimax cu o inferență la un model de învățare automată (exemplu: regresie folosind o rețea neuronală convoluțională)
- adăugarea unui nedeterminism (randomness) pentru agent