

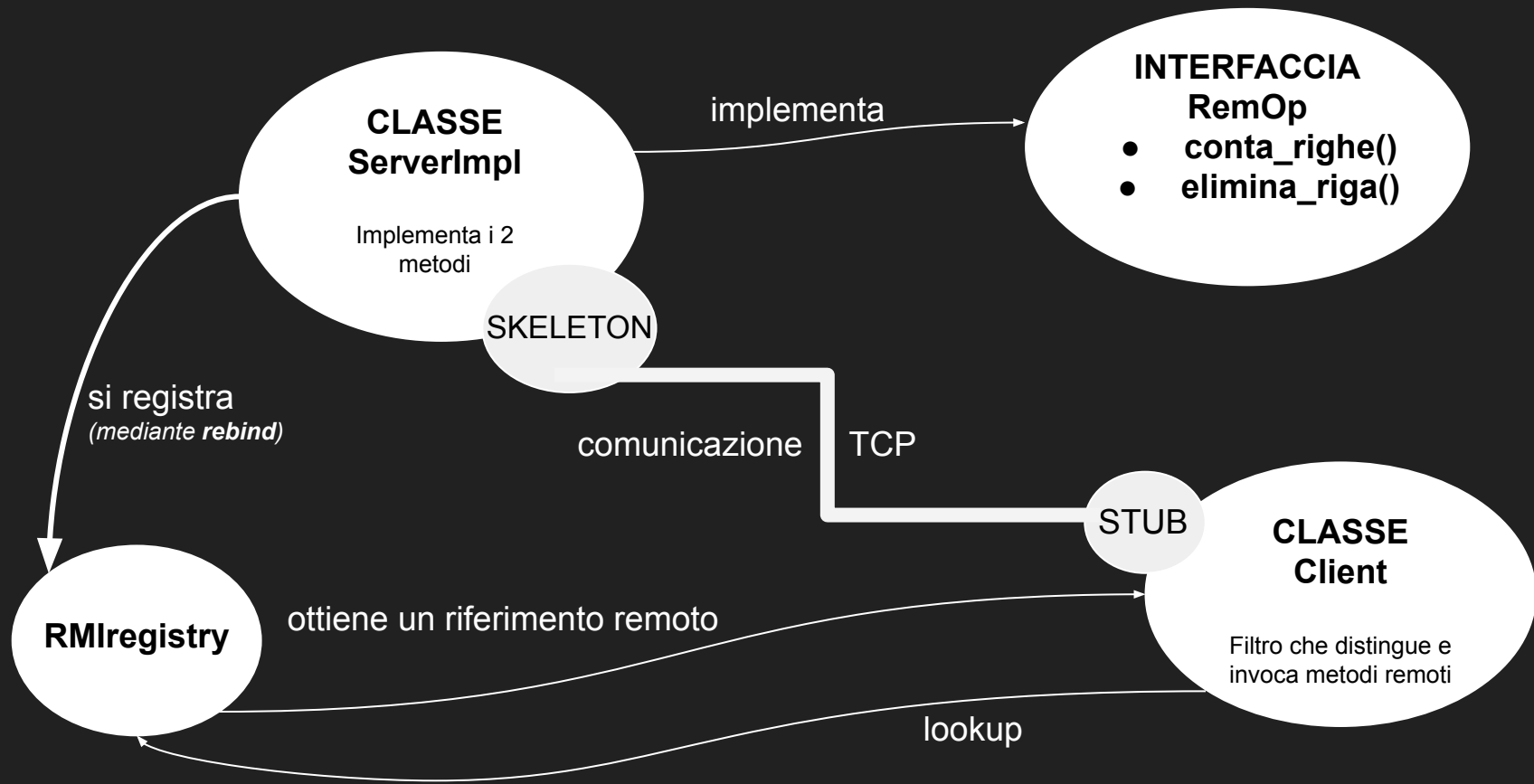
Proposta soluzione esercitazione 6 JAVA RMI

Mattia Innocenti - 825046
Federico Macchiavelli - 825621
Andrea Proia - 825784
Luca Bartolomei - 825005

Specifiche servizio

L'applicativo ha come obiettivo un'applicazione C/S che consenta di effettuare 2 operazioni remote su file testo:

1. un servizio per **contare le righe** che contengono un numero di parole superiore ad un intero espresso dal cliente
2. un servizio per **eliminare una riga** da un file remoto passando il nome del file e il numero di riga da rimuovere



Client - struttura

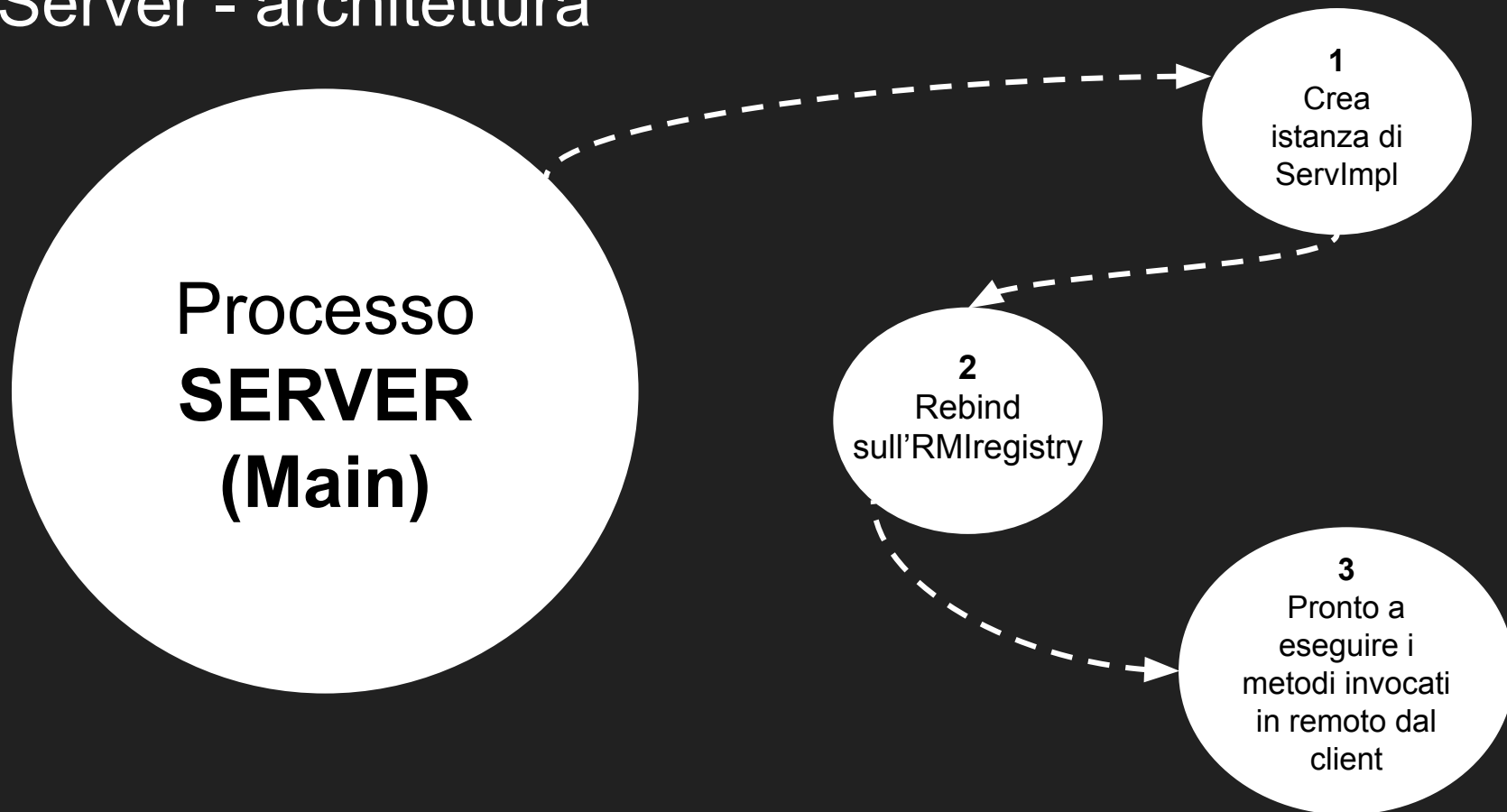
Operazioni preliminari:

- Controllo argomenti invocazione
- Invocazione metodo **lookup** per recuperare riferimento remoto memorizzandolo in una variabile di tipo interfaccia (RemOp)

Struttura ciclica (filtro):

1. Richiedo tipologia servizio (C/E)
2. Richiedo il nome del file sul quale invocare il servizio
3. Distinguo richiesta:
 - a. `conta_righe()`: richiedo soglia parole per considerare la riga come valida nel conteggio
 - b. `elimina_riga()`: richiedo il numero della linea da eliminare
4. Attendo esito e stampo a video

Server - architettura



Server - struttura conta_righe()

- **public int conta_righe(String fileName, int soglia)**
 - a. Controllo esistenza e leggibilità file
 - b. Ciclo di lettura A CARATTERE mediante *FileReader* fino a fine file ed eventuale incremento contatore ogni lettura di '\n' se la linea possiede più di “soglia” parole
 - c. Chiusura strutture e return

Server - struttura elimina_riga()

- **int elimina_riga(String fileName, int nRiga)**
 - a. Controllo esistenza e leggibilità file
 - b. Ciclo di lettura per controllare che: $nRiga \leq$ numero linee file
 - c. Creazione file temporaneo fileTmp
 - d. Ciclo di lettura A BUFFER mediante *BufferedReader* fino a fine file per scrivere tutte le linee su fileTmp (tranne quella da eliminare)
 - e. Sovrascrittura file originale con fileTmp
 - f. Chiusura strutture e return

Problemi riscontrati

I metodi messi a disposizione dal Server presentano entrambi dei possibili problemi di concorrenza:

1. La funzionalità del conteggio delle righe presenta criticità laddove un client tentasse contemporaneamente di eliminare una riga dallo stesso file, oggetto dell'operazione
2. La possibilità di eliminare una riga da un file, potrebbe essere causa di problemi di contention se eseguita contemporaneamente da più client

Conclusioni

Le entità in gioco sono necessariamente 3:

Il **Registry** che deve essere sempre in esecuzione (dopo la compilazione dell'interfaccia e del server)

Il **Server** che si deve registrare sul Registry per poter essere riferito correttamente dai clienti

Il **Client** che dopo aver recuperato dal registry il riferimento all'oggetto remoto, può invocare in remoto i metodi definiti sul Server

Tramite l'implementazione dell'interfaccia remotizzabile **RemOp** (condivisa da entrambe le entità C/S) è stato quindi possibile definire sul server i metodi invocabili in remoto dal client, e la comunicazione è stata resa possibile dai 2 proxy Stub (Client) e Skeleton (Server) che hanno reso possibile l'invocazione remota e la gestione delle chiamate sul Server.