

## Proff. De Simone, Della Corte

**Durata della prova: 120 minuti**

[illegible]

*Al termine della prova lo studente dovrà far verificare il funzionamento del programma ad un membro della Commissione.*

[illegible]

The diagram illustrates the architecture of the system. It consists of three main components: Client (Java), Manager (Java), and Registro Vendite (Python). The Client (Java) sends a request topic to the JMS/STOMP Topics. The Manager (Java) sends a request topic to the JMS/STOMP Topics and receives compra and vendi topics from the JMS/STOMP Topics. The Registro Vendite (Python) receives compra and vendi topics from the JMS/STOMP Topics and updates the vendite.csv file. The Registro Vendite (Python) also contains a lista of data.

```

graph TD
    Client[Client (Java)]
    Manager[Manager (Java)]
    Registro[Registro Vendite (Python)]
    Topics[JMS/STOMP Topics]
    File[vendite.csv]

    Client -.->|request topic| Topics
    Manager -.->|request topic| Topics
    Topics -.->|compra topic| Manager
    Topics -.->|vendi topic| Manager
    Topics -.->|compra topic| Registro
    Topics -.->|vendi topic| Registro
    Registro -.->|update| File
    
```

- **Client.** Componente **Java** che genera richieste di tipo *compra* o *vendi*, sotto forma di messaggi JMS. Client attende 2 secondi tra le richieste, e genera un totale di 20 richieste. Per ogni richiesta, Client genera un MapMessage composto da due campi: *tipo*, che conterrà il tipo di richiesta (cioè *compra* o *vendi*), e *valore*, che conterrà

- Il MapMessage è poi inviato sul topic *request*. Il tipo di richiesta viene fornito da terminale dall'utente. Ad esempio: **java nomepackage.Client compra**

- **Registro Vendite.** Componente **Python** che implementa la ricezione asincrona STOMP sul topic *compra* e sul topic *vendi*.

- **Alla ricezione di un TextMessage sul topic *compra***, il componente avvia un nuovo processo converte la stringa presente nel messaggio (e.g., “giggino-2500”) in un dictionary (e.g., {“giggino” : 2000}) e inserisce tale elemento in una lista.
- **Alla ricezione di un TextMessage sul topic *vendi***, il componente avvia un nuovo processo che consumerà il primo elemento dalla coda per il quale il budget associato ad un cliente (e.g., il budget del cliente *giggino* è *2500* nell’esempio precedente) è maggiore o uguale alla quotazione dell’auto riferita nel messaggio che ha appena ricevuto (nell’esempio di messaggio *AB000CD-2150*, il budget del client *giggino* è maggiore o uguale a *2150*). Nel caso di avvenuta consumazione, aggiungere ad un file *vendite.csv* con intestazione uguale a ***targa,utente,quotazione\_auto*** le informazioni relative alla vendita. Per esempio il file risultante sarà:

**targa,utente,quotazione\_auto**  
AB000CD,giggino,2150

**L'accesso alla lista deve essere gestito in mutua esclusione tra i processi generati.**

Lo studente dovrà sviluppare le applicazioni **Client**, **Manager** e **Registro Vendite**. Il sistema sarà testato da terminale con 2 Client (uno per *compra* e uno per *vendi*), 1 Manager, e 1 Registro Vendite.