Reti dei Calcolatori
Relazione Progetto Green Pass
Docenti: Alessio Ferone-Aniello Castiglione

Studente: Alfredo Scognamiglio
Matricola: 0124001647
Anno Accademico: 2021/2022

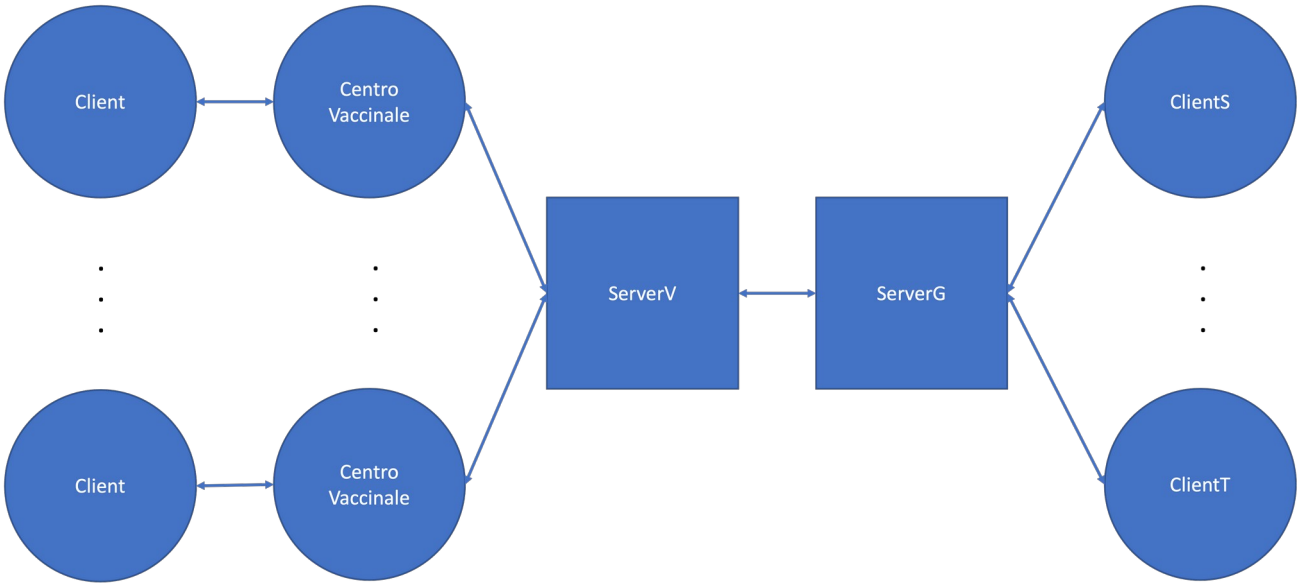
Sommario

Descrizione del progetto

Progettare ed implementare un servizio di gestione dei green pass secondo le seguenti specifiche. Un utente, una volta effettuata la vaccinazione, tramite un client si collega ad un centro vaccinale e comunica il codice della propria tessera sanitaria. Il centro vaccinale comunica al ServerV il codice ricevuto dal client ed il periodo di validità del green pass. Un ClientS, per verificare se un green pass è valido, invia il codice di una tessera sanitaria al ServerG il quale richiede al ServerV il controllo della validità. Un ClientT, inoltre, può invalidare o ripristinare la validità di un green pass comunicando al ServerG il contagio o la guarigione di una persona attraverso il codice della tessera sanitaria.

Descrizione del modello architetturale e schema dell'architettura

Viene ora spiegato il modello architetturale sulla quale si basa il progetto del Green Pass. Il ServerV e il ServerG hanno la funzione di gestire quello che è il flusso di informazioni che include il Client, il ClientS e il ClientT. Inoltre abbiamo il Centro Vaccinale ed il ServerG che fungono da Peer in quanto ricevono richieste e forniscono servizi.Qui di seguito è riportata l immagine che riassumo lo schema architetturale del lavoro:



Descrizione e schemi del protocollo applicazione

Il protocollo utilizzato in questo progetto è il socket TCP.
Tramite il socket è possibile inviare e ricevere dati e stabilire una comunicazione tra un host mittente e un host destinatario.
E' stata utilizzata la libreria 'pthread' che permette di generare più thread in grado di effettuare lavori diversi ed indipendenti.
Per poter quindi trasmettere dati bisogna stabilire una connessione tra le entità in questione. Queste ultime sono il server che riceve richieste, le elabora e fornisce una risposta ed i client che richiedono un servizio.
Nell'applicazione del Green Pass abbiamo anche il Peer, ovvero un entità che è in grado di essere sia client che server contemporaneamente. Le entità che in questo caso fungono da Peer sono il Centro Vaccinale, che riceve la tessera sanitaria dal Client e la invia al ServerV comunicando anche il periodo di validità della tessera, ed il ServerG che gestisce le richieste dei ClientS e T e comunica quest ultime al ServerV.
E' stato usato una 'strcpy' per andare a differenziare chi sta contattando chi prima di ogni scambio di dati fra le varie entità. In questo modo ad esempio quando il ServerV riceverà una richiesta dal ServerG andrà ad effettuare una FullRead ricevendo un buffer contenente 'ServerG'.

Dettagli implementativi del client

1. *Client*
   Tramite il Client, l utente immette il codice della propria tessera sanitaria.Inoltre stabilisce la connessione con il Centro Vaccinale sulla porta 8080.
2. *CentroVaccinale*
   Ora è il turno del Centro Vaccinale. Abbiamo detto che questa entità è un Peer quindi funge sia da client che da server. Rimane in ascolto per le richieste da parte del Client e,una volta ricevuti i dati,si collega col ServerV sulla porta 8081.Una volta creato il thread indipendente per ogni connessione, il Centro Vaccinale effettua la strcpy per riempire il buffer con la propria identità in questo modo:
   - strcpy(buffer,"CentroVaccinale");
   cosi facendo il ServerV capirà che sta comunicando con il CentroVaccinale piuttosto che con altre entità.
3. ClientS
   Per quanto riguarda il ClientS, esso permette di verificare se un green pass è valido andando ad inviare il codice di una tessera sanitaria al ServerG.Viene utilizzata la porta 8082 per la comunicazione.Dopo aver effettuato il controllo il ClientS riceverà l esito negativo o positivo del controllo della validità della tessera sanitaria.
4. ClientT
   Quest ultimo client utilizza la porta 8082 per la comunicazione.Prende in input il codice della tessera sanitaria e la scelta effettuata dall utente di invalidare o ripristinare quella determinata tessera.La modifica avverrà se la tessera è già presente all interno del ServerV(che fa da DB), in caso contrario il ClientT riceverà un responso negativo.

Dettagli implementativi del server

1. *ServerV*
   Il serverV comunica sia col CentroVaccinale sia col ServerG tramite la porta 8081.Tutti i controlli e le comunicazioni passano per questa entità.Il ServerV utilizza la strcmp per andare a capire chi lo sta contattando e agire di conseguenza.All’ interno del ServerV vengono riempiti i campi di una struct che conterrà quelle che sono le informazioni dell’utente.Infine il ServerV,oltre ad effettuare i controlli riguardanti i codici delle tessere sanitarie,avrà il compito anche di rispondere a queste richieste facendo quindi da mediatore tra il CentroVaccinale ed il ServerG.
2. *ServerG*
   Il ServerG,cosi come il CentroVaccinale, è un Peer che offre servizio al ClientS e T.Utilizza la porta 8082 per comunicare con i due client e la porta 8081 per comunicare con il ServerV.Il compito del ServerG è quello di fare da tramite ed inviare tutti i messaggi,provenienti dai ClientS e T, al ServerV.

Manuale Utente

Per avviare il programma bisogna avviare il ServerV.In seguito avviare il CentroVaccinale e restare in attesa di una connessione da parte del Client.Successivamente si può avviare il ServerG ed i ClientS e T.

Istruzioni per la compilazione

- ServerV
  Per compilare il ServerV bisogna andare nella cartella ‘ServerV’,avviare il terminale e digitare il comando ‘make’ in modo tale che il makefile, al cui interno è presente il comando ‘gcc serverV.c -o serverV’,venga eseguito automaticamente.

- CentroVaccinale
  Per compilare il CentroVaccinale bisogna andare nella cartella ‘CentroVaccinale’,avviare il terminale e digitare il comando ‘make’ in modo tale che il makefile, al cui interno è presente il comando ‘gcc centro_vaccinale.c -o centro_vaccinale’,venga eseguito automaticamente.

- Client
  Per compilare il Client bisogna andare nella cartella ‘Client’,avviare il terminale e digitare il comando ‘make’ in modo tale che il makefile, al cui interno è presente il comando ‘gcc client.c -o client’,venga eseguito automaticamente.

- ServerG
  Per compilare il ServerG bisogna andare nella cartella ‘ServerG’,avviare il terminale e digitare il comando ‘make’ in modo tale che il makefile, al cui interno è presente il comando ‘gcc serverG.c -o serverG’,venga eseguito automaticamente.

- ClientS
  Per compilare il ClientS bisogna andare nella cartella ‘ClientS’,avviare il terminale e digitare il comando ‘make’ in modo tale che il makefile, al cui interno è presente il comando ‘gcc clientS.c -o clientS’,venga eseguito automaticamente.

- ClientT
  Per compilare il ClientT bisogna andare nella cartella ‘ClientT’,avviare il terminale e digitare il comando ‘make’ in modo tale che il makefile, al cui interno è presente il comando ‘gcc clientT.c -o clientT’,venga eseguito automaticamente.

Istruzioni per l'esecuzione

- *ServerV*
  Per eseguire il ServerV, dopo essere andati nella cartella 'ServerV' e aver aperto il terminale, bisogna dare il comando: './serverV'

- *CentroVaccinale*
  Per eseguire il CentroVaccinale, dopo essere andati nella cartella 'CentroVaccinale' e aver aperto il terminale, bisogna dare il comando: './centro_vaccinale'

- *Client*
  Per eseguire il Client, dopo essere andati nella cartella 'Client' e aver aperto il terminale, bisogna dare il comando: './client'

- *ServerG*
  Per eseguire il ServerG, dopo essere andati nella cartella 'ServerG' e aver aperto il terminale, bisogna dare il comando: './serverG'

- *ClientS*
  Per eseguire il ClientS, dopo essere andati nella cartella 'ClientS' e aver aperto il terminale, bisogna dare il comando: './clientS'

- *ClientT*
  Per eseguire il ClientT, dopo essere andati nella cartella 'ClientT' e aver aperto il terminale, bisogna dare il comando: './clientT'