

# **Università degli studi di Napoli Parthenope**

**Informatica**



## **Progetto Laboratorio Reti di Calcolatori Green Pass**

Componenti:

Anastasio Ilaria 0124002185

Panico Gennaro 0124002115

Selvaggio Christian 0124002164

# Indice

Descrizione del progetto .....	3
Descrizione dell'architettura .....	4
Utente .....	4
Centro Vaccinale .....	5
ServerV .....	5
ServerG .....	6
ClientS .....	6
ClientT .....	6
Descrizione del modello .....	7
Descrizione del protocollo .....	8
Utente -> Centro Vaccinale .....	9
Centro Vaccinale -> ServerV .....	9
ClientS -> ServerV .....	10
ServerG -> ServerV .....	10
ClientT -> ServerG .....	11
Guida alla compilazione .....	12
Guida all'esecuzione .....	12
Esempi di funzionamento .....	12

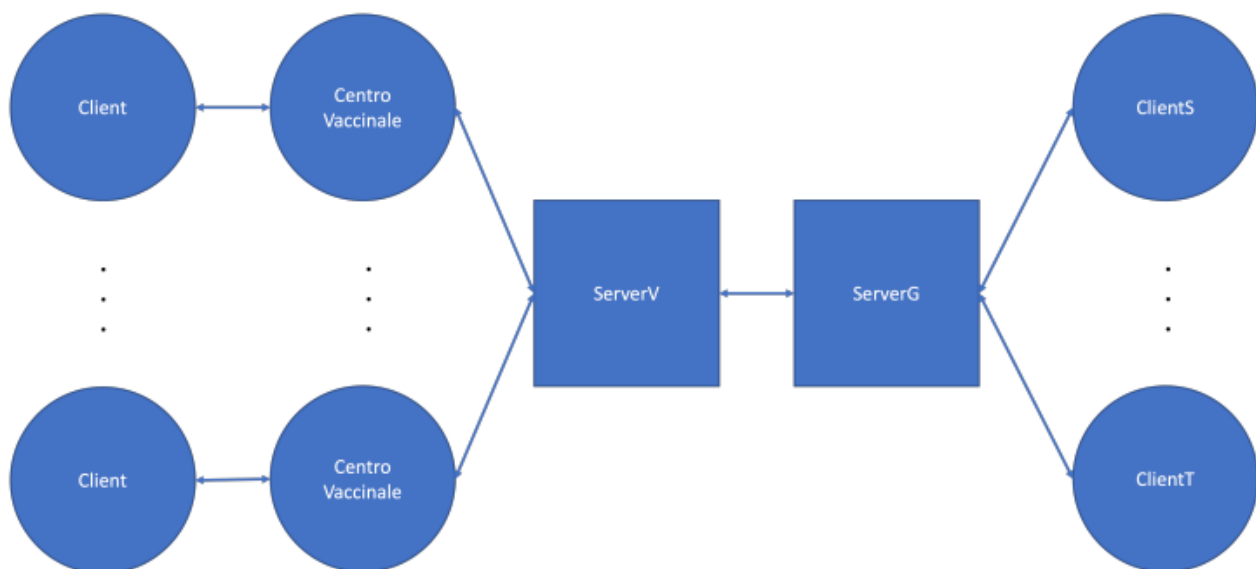
## Descrizione del progetto

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 dell'architettura

Il progetto in questione riguarda la creazione e la gestione dei Green Pass.

Un **Utente**, effettuata la vaccinazione, comunica i propri dati anagrafici ed il codice fiscale della tessera sanitaria ad un **Centro Vaccinale**. Quest'ultimo, in primis informerà l'Utente dell'avvenuta ricezione o meno dei dati e successivamente invierà il codice fiscale della tessera sanitaria, seguito dalla data di inizio e di fine validità ad un **ServerV** la cui funzione è quella di memorizzare tutti i Green Pass in un filesystem.

In seguito, un **ClientS**, per verificare se un Green Pass è valido o meno, invia un codice fiscale ad un **ServerG**, il quale richiede al ServerV di inviargli il certificato verde corrispondente. Infine, un **ClientT** ha la funzione di poter invalidare o meno un Green Pass effettuando la comunicazione della guarigione o del contagio di una persona al ServerG attraverso il codice fiscale della tessera sanitaria.

## Utente

Abbiamo definito un pacchetto **VACCINAZIONE** che l'Utente invia al Centro Vaccinale per la creazione del Green Pass.

```
typedef struct {  
    char nome[BUFF_MAX_SIZE];  
    char cognome[BUFF_MAX_SIZE];  
    char cod_fisc[COD_SIZE];  
} VACCINAZIONE;
```

# Centro Vaccinale

Ricevuto il pacchetto dall'Utente, genera un certificato verde al quale associa un periodo di validità (nel nostro caso 4 mesi). Infine, invia un pacchetto avente il codice fiscale della tessera sanitaria e la data di inizio e di fine validità del Green Pass al ServerV.

Abbiamo definito due pacchetti:

1. Pacchetto **Date**:

```
typedef struct {  
    int giorno;  
    int mese;  
    int anno;  
} DATE;
```

2. Pacchetto **GP**:

```
typedef struct {  
    char cod_fisc[COD_SIZE];  
    DATE data_inizio;  
    DATE data_fine;  
} GP
```

# ServerV

Dopo aver ricevuto i dati da parte del Centro Vaccinale, crea un file univoco per ciascun Green Pass avente come nome il codice fiscale della tessera sanitaria, e all'interno del quale troveremo la data di inizio e di fine validità.

Inoltre, Il ServerV viene contattato dal ServerG, per la richiesta di un certificato verde specifico. Per gestire le diverse comunicazioni, viene inviato ad esso un bit di riconoscimento che gli permetterà di identificare chi ha richiesto la connessione.

Se il bit è uguale a **1** il ServerV effettuerà la connessione con il **Centro Vaccinale**.

Se il bit è uguale a **0** il ServerV effettuerà la connessione con il **ServerG**.

L'ultima funzione del ServerV è quella di invalidare o ripristinare il Green Pass attenendosi alle disposizioni del ClientT.

## ServerG

Attende la ricezione di un codice fiscale di una tessera sanitaria associato ad un Green Pass da parte del ClientS. Ricevuti i dati, effettuerà una connessione con il ServerV ed invierà il codice fiscale della tessera. Quest'ultimo gli invierà il certificato verde richiesto (se esiste), per testarne la validità. Infine, il ServerG notificherà al ClientS se il Green Pass è valido o meno.

## ClientS

Invia un codice fiscale di una tessera sanitaria associata ad un Green Pass al ServerG per testarne la validità. Il ClientS rimane in attesa di una risposta da parte del ServerG.

## ClientT

Ha la funzione di invalidare o ripristinare un Green Pass a seconda dell'esito del tampone associato al codice della tessera sanitaria. Dunque, comunica l'esito di un referto, contenuto nel pacchetto **REPORT**, al ServerG, il quale si connette con il ServerV per la modifica del file d'interesse.

Se invia **1**, il Green Pass sarà **ripristinato**.

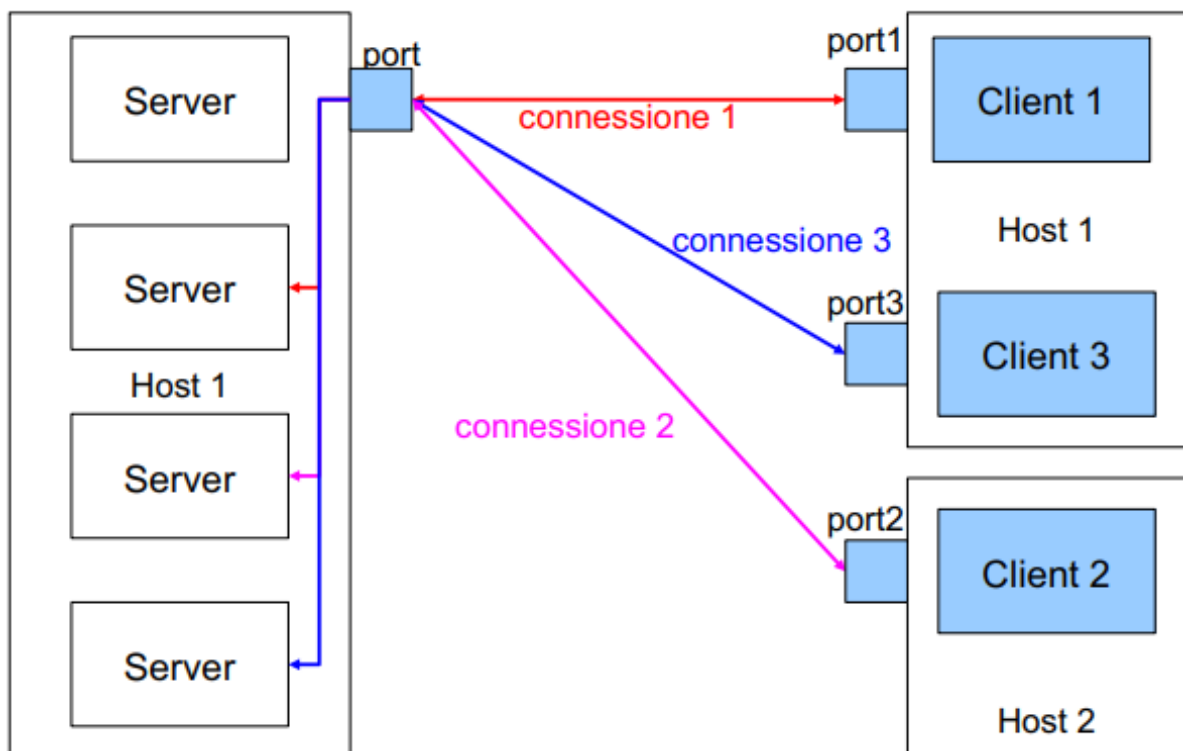
Se invia **0**, il Green Pass sarà **invalidato**.

```
typedef struct {  
    char cod_fisc[COD_SIZE];  
    char report;  
} REPORT;
```

# Descrizione del modello

Il modello di comunicazione utilizzato è quello **client-server**.

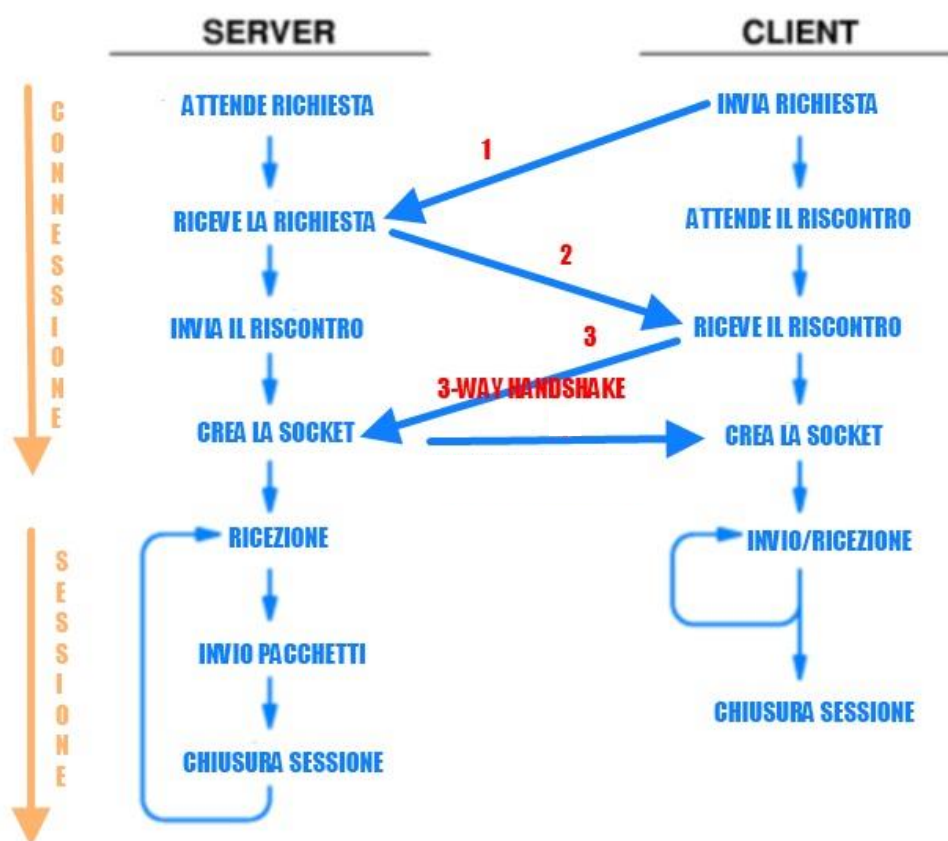
Il server utilizzato è il **server concorrente**, il quale può gestire più client contemporaneamente. L'implementazione classica prevede che ad ogni richiesta client venga generato un processo figlio che la gestisca. Ciò è realizzato attraverso la systemcall **fork()**.



# Descrizione del protocollo

Il protocollo utilizzato è **TCP/IP** le cui funzionalità sono svolte da un elemento fondamentale che rappresenta il “cervello” del protocollo: la **Socket TCP**. Essa è il mezzo attraverso il quale un pacchetto di destinazione viene dotato di tutte le informazioni necessarie per partire alla volta dell’host destinatario, oltre che il principale responsabile nello stabilire la connessione tra due host e mantenere la sessione per poi rigenerare la connessione all’invio di ulteriori pacchetti. Quindi quando due host comunicano tra loro, in realtà sono proprio le loro Socket a comunicare.

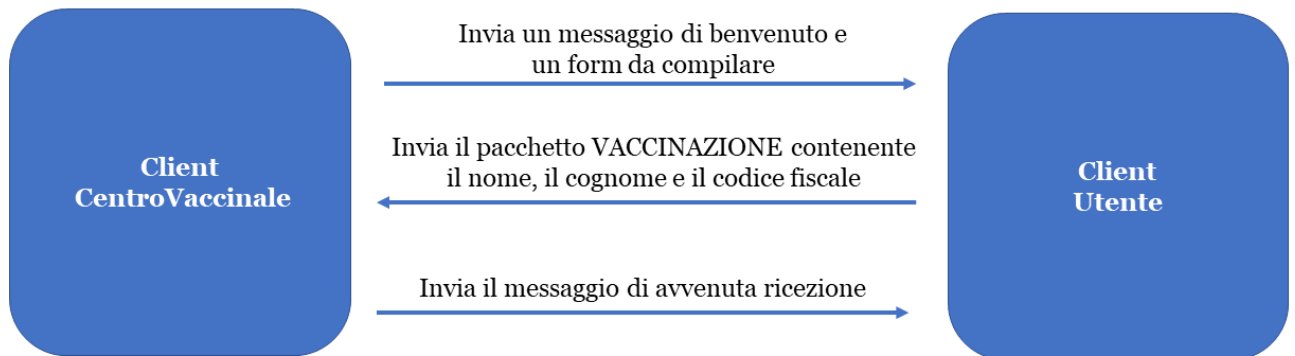
Ogni Socket Client, identificata con l’identificativo della connessione TCP a cui appartiene, è quindi accoppiata con una Socket Server e la loro comunicazione avviene attraverso la seguente procedura:



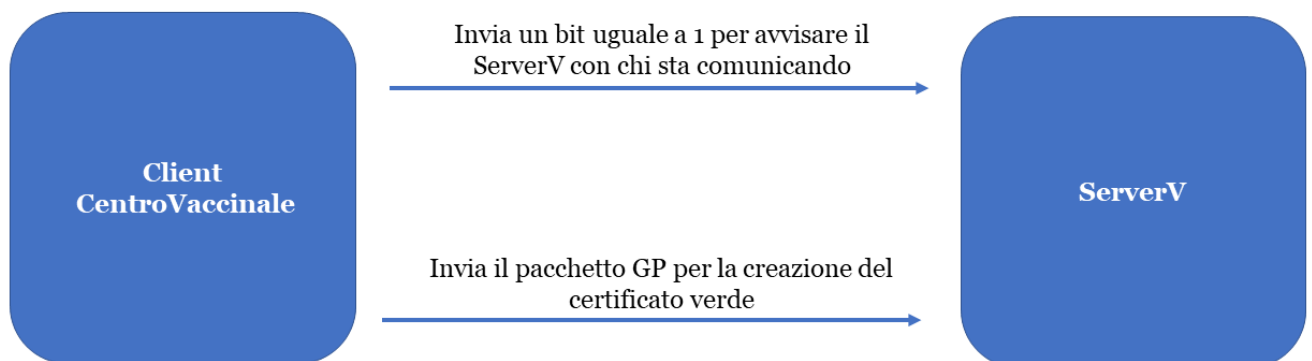


Di seguito sono riportati gli schemi di funzionamento dei client e dei server.

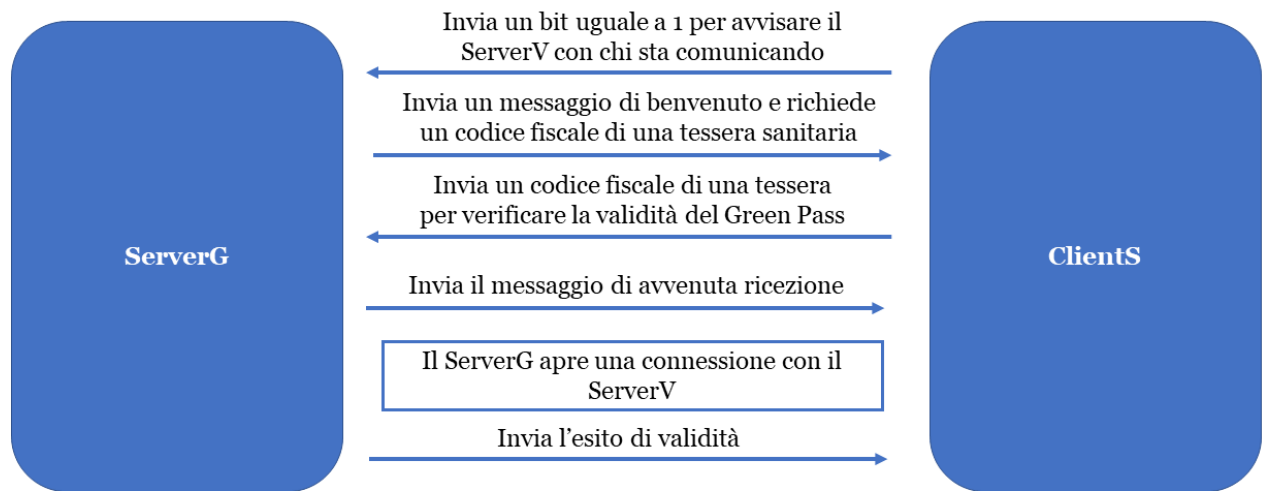
## Utente -> Centro Vaccinale



## Centro Vaccinale -> ServerV

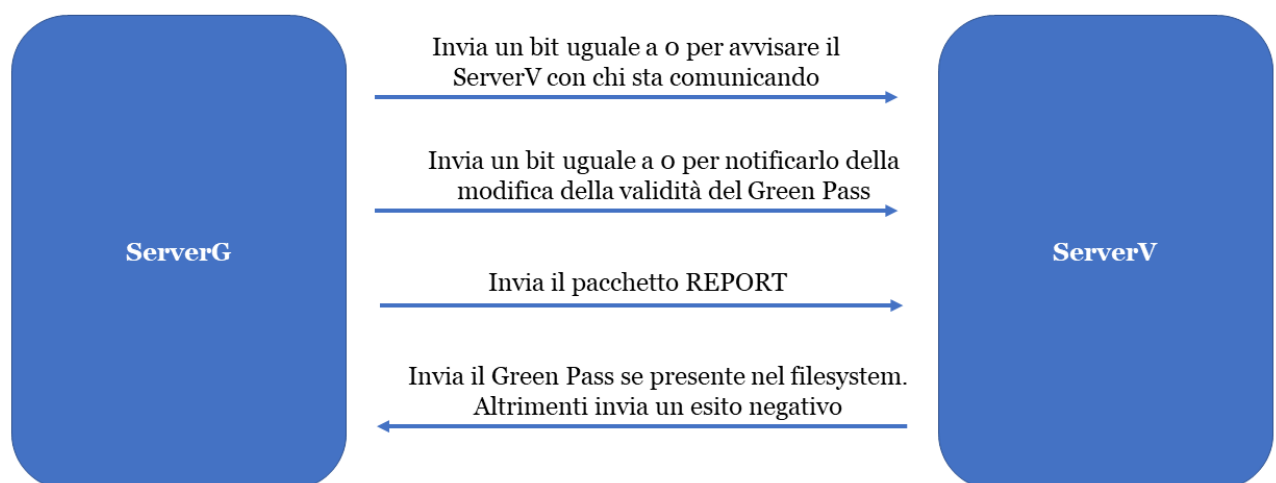


## ClientS -> ServerG

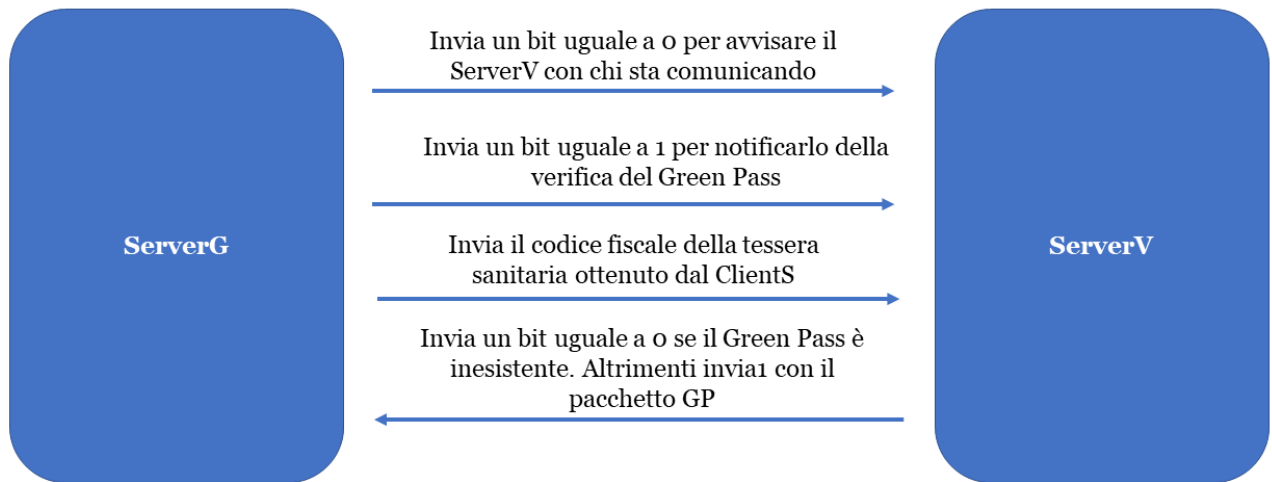


## ServerG -> ServerV

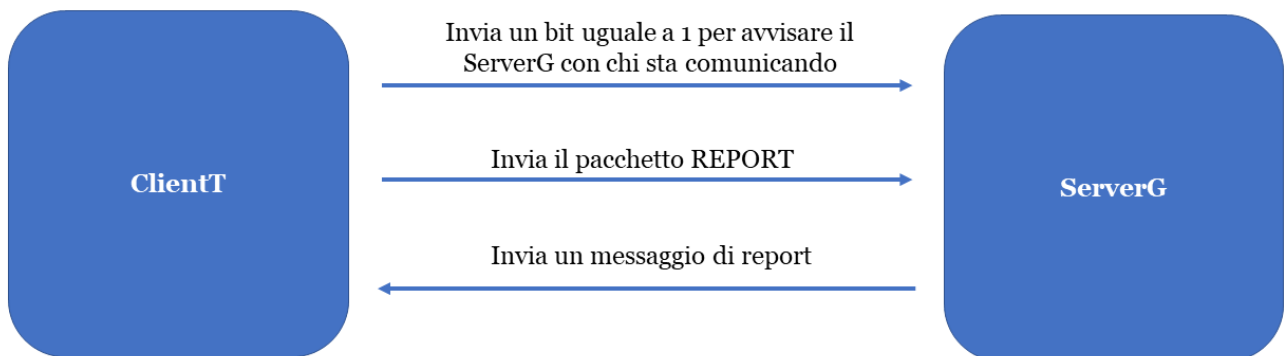
Operazione con il ClientT



Operazione con il ClientS



**ClientT -> ServerG**



# Guida alla compilazione

- gcc -o CentroVaccinale CentroVaccinale.c
- gcc -o Utente Utente.c
- gcc -o ClientS ClientS.c
- gcc -o ClientT ClientT.c
- gcc -o ServerV ServerV.c
- gcc -o ServerG ServerG.c

# Guida all'esecuzione

Per una corretta esecuzione si consiglia tale ordine:

- ./ServerV
- ./CentroVaccinale
- ./ServerG
- ./Utente localhost
- ./ClientS
- ./ClientT

# Esempi di funzionamento

Inserimento dei dati da parte dell'Utente.

```
chappys@DESKTOP-2BK5J17:/mnt/c/Users/Christian_Selvaggio/Desktop/ProgettoRetiDefinitivo$ ./Utente localhost
--- Benvenuto nel centro vaccinale ---
Immettere nome, cognome e codice fiscale della tessera sanitaria per inserirli sulla piattaforma.

Inserisci il nome: ilaria
Inserisci il cognome: anastasio
Inserisci il codice della tessera sanitaria. Attenzione! Devi inserire esattamente 16 caratteri: nstlri00p54f839p
Inserimento dei dati avvenuto con successo
```

Generazione del Green Pass con il corrispondente periodo di validità.

```
In attesa di nuove domande per la vaccinazione

I dati ricevuti sono:
Nome: ilaria
Cognome: anastasio
Codice Fiscale Tessera Sanitaria: nstlri00p54f839p

La data di inizio validità del green pass e': 15:01:2023
La data di fine validità del Green Pass e': 15:05:2023
```

Verifica della validità del Green Pass appena generato.

```
chappys@DESKTOP-2BKSJI7:/mnt/c/Users/Christian_Selvaggio/Desktop/ProgettoRetiDefinitivo$ ./ClientS
--- Benvenuto nel ServerG ---
Inserire il codice fiscale della tessera per verificarne la validità

Inserisci codice tessera sanitaria [N.B. immettere esattamente 16 caratteri]: nstlri00p54f839p

I dati sono stati ricevuti correttamente!

Caricamento in corso...

Il Green Pass è valido, operazione terminata!
```

Invalidazione del Green Pass.

```
chappys@DESKTOP-2BKSJI7:/mnt/c/Users/Christian_Selvaggio/Desktop/ProgettoRetiDefinitivo$ ./ClientT
--- Client T ---
Inserire il codice di tessera sanitaria ed il referto del tampone per l'invalidazione o il ripristino del Green Pass
Inserisci codice tessera sanitaria [N.B. immettere esattamente 16 caratteri]: nstlri00p54f839p
Se il Green Pass non è valido, immettere 0
Se il Green Pass è valido, immettere 1
DIGITA: 0

Inoltre richiesta di invalidazione del Green Pass!
--- Operazione conclusa con successo ---
```

Verifica del Green Pass appena invalidato.

```
chappys@DESKTOP-2BKSJI7:/mnt/c/Users/Christian_Selvaggio/Desktop/ProgettoRetiDefinitivo$ ./ClientS
--- Benvenuto nel ServerG ---
Inserire il codice fiscale della tessera per verificarne la validità

Inserisci codice tessera sanitaria [N.B. immettere esattamente 16 caratteri]: nstlri00p54f839p

I dati sono stati ricevuti correttamente!

Caricamento in corso...

Il Green Pass non è valido, operazione terminata!
```