

# PROGETTO RETI DI CALCOLATORI



**Realizzato da:**

Maurizio Speranza 0124002054

Luigi Malvone 0124002200

**Anno Accademico:**

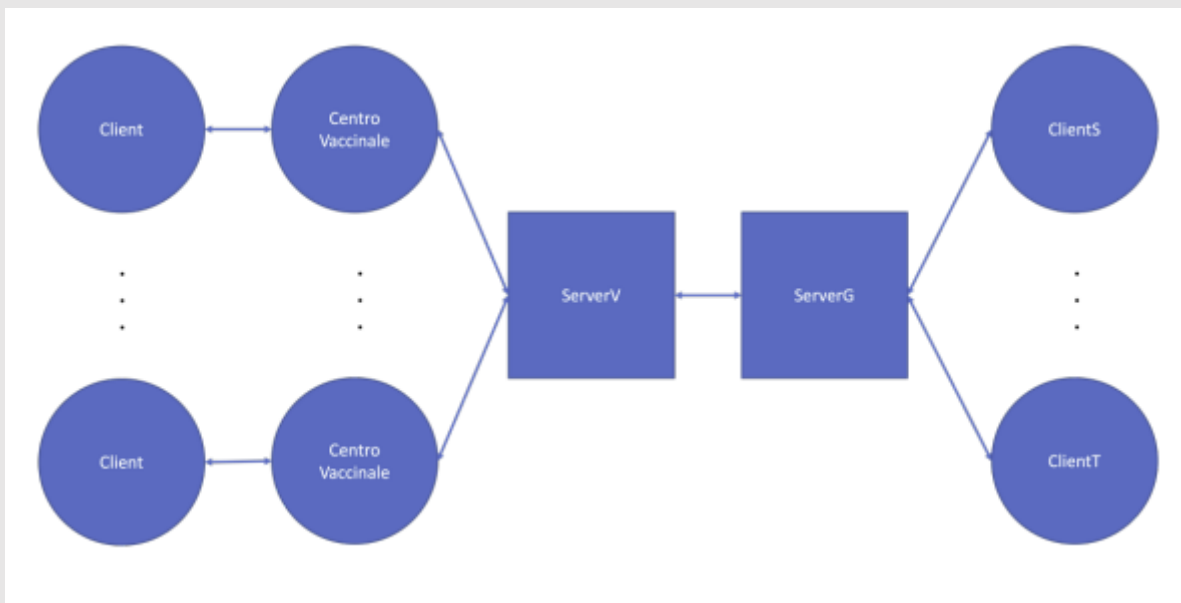
2022 - 2023

## Indice

<b>Descrizione del progetto .....</b>	<b>3</b>
<b>Client – Utente .....</b>	<b>4</b>
<b>Client – Centro Vaccinale .....</b>	<b>4</b>
<b>ServerV – Server Vaccinale.....</b>	<b>5</b>
<b>ServerG – Server Verifica.....</b>	<b>6</b>
<b>ClientS – App Verifica .....</b>	<b>6</b>
<b>ClientT – ASL .....</b>	<b>6</b>
<b>Descrizione Architettura.....</b>	<b>7</b>
<b>Descrizione del protocollo applicazione.....</b>	<b>8</b>
<b>Comunicazione Utente -&gt; CentroVaccinale .....</b>	<b>8</b>
<b>Comunicazione CentroVaccinale -&gt; ServerVaccinale.....</b>	<b>8</b>
<b>Comunicazione AppVerifica -&gt; ServerVerifica .....</b>	<b>9</b>
<b>Comunicazione ServerVerifica -&gt; ServerVaccinale .....</b>	<b>9</b>
<b>Comunicazione ASL -&gt; ServerVerifica.....</b>	<b>10</b>
<b>Manuale Utente – Compilazione .....</b>	<b>11</b>
<b>Manuale Utente – Esecuzione .....</b>	<b>11</b>
<b>Simulazione .....</b>	<b>11</b>

## Descrizione del progetto

Il progetto proposto è utilizzato per la gestione dei certificati vaccinali, noto come Green Pass. Un utente, dopo aver effettuato la vaccinazione, comunica i suoi dati anagrafici ed il numero di tessera sanitaria al Centro Vaccinale, il quale comunicherà i dati al cliente e invierà il codice della tessera sanitaria, con il periodo di validità del Green Pass al ServerVaccinale, che svolge un ruolo di database salvando in un filesystem tutti i certificati verdi ricevuti. Il ClientS può essere visto come l'app che scansiona i Green Pass, che invia un codice di una tessera sanitaria al ServerG che a sua volta chiede al ServerV di inviargli un Green Pass; così facendo il ServerVerifica effettuerà l'operazione di scansione per poi comunicarlo al ClientS. Infine, abbiamo un ClientT, identificabile come l'ASL, che può invalidare o ripristinare la validità di un Green Pass comunicando il contagio o la guarigione di una persona al ServerG tramite il codice della tessera.



## Client – Utente

Il client utente si connette al centro vaccinale, e dopo aver stabilito la connessione, riceve un messaggio di avvenuta connessione. Successivamente, il centro chiede all'utente di compilare un modulo con dati anagrafici e numero di tessera sanitaria (deve essere esattamente 10 caratteri). Il client verifica il form compilato dall'utente e lo invia al centro vaccinale il quale notificherà l'avvenuta ricezione.

Definiamo un pacchetto applicazione chiamato VAX\_REQUEST usato dall'utente e inviato al centro vaccinale per generare il corrispondente certificato verde. VAX\_REQUEST è una struct contenente 3 campi : nome, cognome, e numero di tessera sanitaria. La dimensione massima (MAX\_SIZE) del nome e cognome è di 1024 byte; mentre la dimensione del numero di tessera (ID\_SIZE) è di 11 byte poiché viene contato anche il carattere terminatore.

```
typedef struct {  
    char name[MAX_SIZE];  
    char surname[MAX_SIZE];  
    char ID[ID_SIZE];  
} VAX_REQUEST;
```

---

## Client – Centro Vaccinale

Il client Centro Vaccinale dopo aver stabilito una connessione con il client Utente invia un messaggio di benvenuto e richiede all'utente la compilazione del modulo con i dati anagrafici e il numero della tessera sanitaria; dopo la verifica di quest'ultimi, il centro vaccinale notifica all'utente l'avvenuta ricezione dei dati. Successivamente il centro vaccinale, con i dati acquisiti dall'utente, genera un GP da inviare al ServerVaccinale. Il centro vaccinale prende la data di inizio e aggiunge 3 mesi creando, in questo modo, il periodo di validità del GP. Successivamente il centro vaccinale invia un pacchetto al server che associa la data di validità del green pass tramite il numero della tessera sanitaria dell'utente.

Per quanto riguarda il pacchetto applicazione, usato dal centro vaccinale, creiamo una struct DATE che consente di salvare una data; quest'ultima è necessaria per creare il pacchetto applicazione che il centro vaccinale invierà al server vaccinale per generare il Green Pass, associando una data di inizio e di fine che indicherà il periodo di validità.

```
typedef struct {  
    int day;  
    int month;  
    int year;  
} DATE;
```

Quindi verrà definita la struct GP\_REQUEST che rappresenta il pacchetto inviato dal centro vaccinale al server vaccinale, contenente il numero della tessera sanitaria dell'utente e data di inizio e fine del GP.

```
typedef struct {  
    char ID[ID_SIZE];  
    DATE start_date;  
    DATE expire_date;  
} GP_REQUEST;
```

---

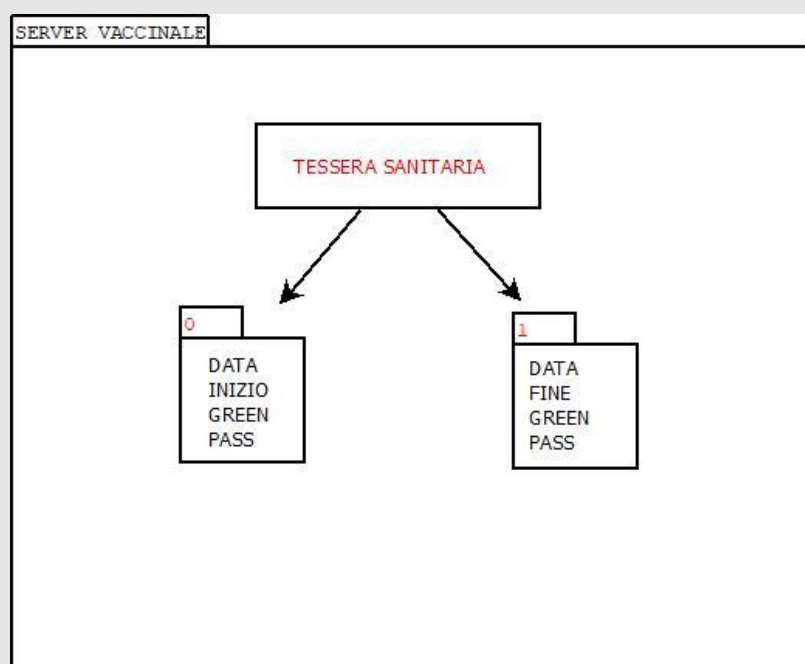
## ServerV – Server Vaccinale

Il server attende le connessioni da parte del centro vaccinale, una volta stabilita, attende la ricezione dei dati. Una volta ottenuti questi ultimi, il server crea un file unico per ogni Green Pass, rinominandolo con il numero di tessera sanitaria ricevuto dal centro vaccinale e contenente il periodo di validità del green pass.

Poiché il server vaccinale ha anche funzione di database, esso fornisce un supporto molto importante per la verifica della validità di un Green Pass. Il server vaccinale viene contattato dal server di verifica che gli richiede un Green Pass specifico; dunque, il server vaccinale eseguirà una query per trovare il Green Pass richiesto tramite numero di tessera sanitaria. Se il file esiste, viene inviato al server di verifica.

La comunicazione con il centro vaccinale e con il server di verifica viene gestita in questo modo: al momento della connessione con il server vaccinale, sia il centro vaccinale che il server di verifica inviano un bit di riconoscimento in modo che il server vaccinale saprà chi ha richiesto la connessione.

- Se il bit inviato è 1 allora il server sarà connesso con il centro vaccinale
- Se il bit inviato è 0 allora il server sarà connesso con il server verifica



## ServerG – Server Verifica

Il server Verifica resta in attesa di connessioni da parte dell'app di scansione (ClientS), dopo aver ottenuto la connessione invierà il form e, una volta ottenuti i dati, il server di verifica si conatterà al server vaccinale inviandogli il numero della tessera sanitaria. Fatto ciò, il server invierà il Green Pass richiesto al server di verifica in modo da testarne la validità.

---

## ClientS – App Verifica

Applicazione per la scansione del Green Pass per testarne la validità. ClientS si connette al server di verifica e dopo aver inserito il codice della tessera sanitaria, l'app attende che il server di verifica e quello vaccinale facciano le loro operazioni e riceve un esito da parte del server di verifica (positivo o negativo).

---

## ClientT – ASL

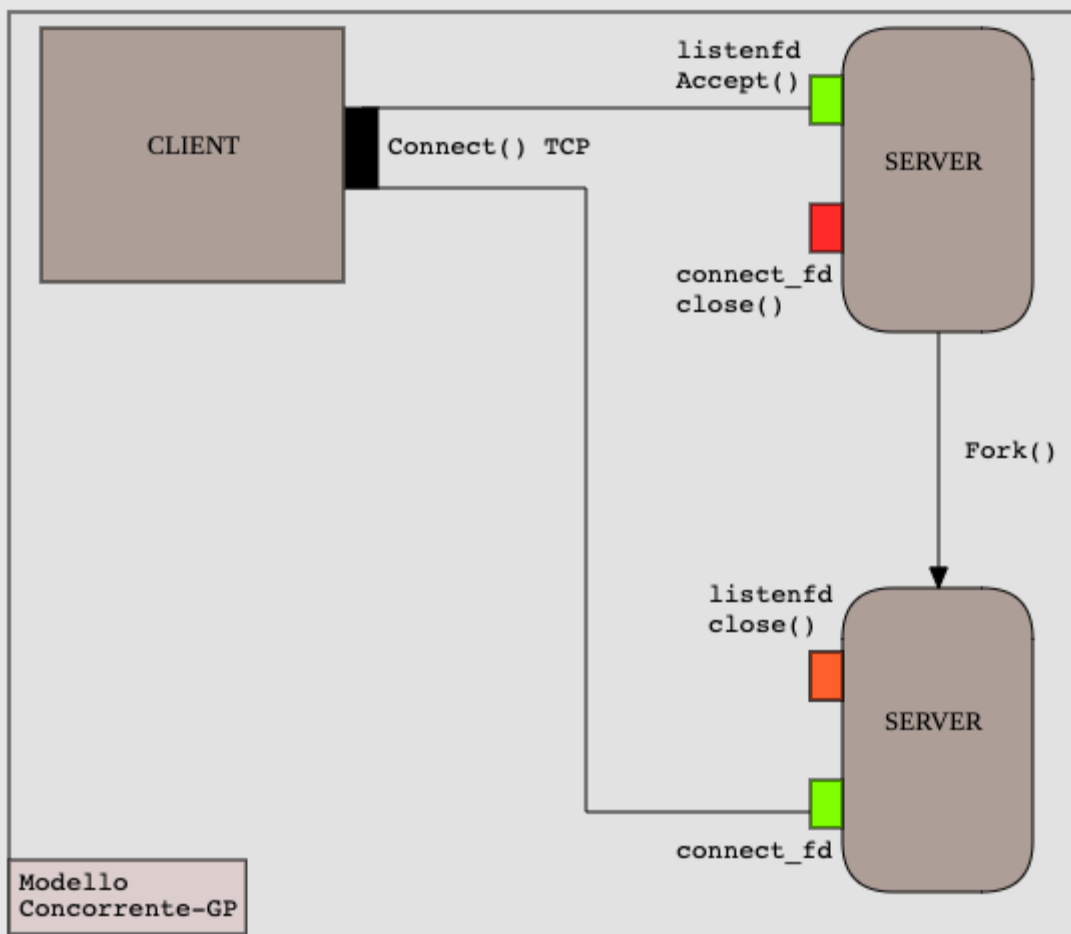
Il clientT rappresenta l'ASL, la quale svolge la funzione di ripristino o invalidamento del Green Pass tramite il codice della tessera sanitaria associata ad un utente in base al referto del tampone. Il clientT comunica al server verifica l'esito del tampone e il server di verifica, a sua volta, comunicherà con il server vaccinale riguardo la modifica del file interessato, in particolare tramite il pacchetto applicazione REPORT, il quale verrà inviato dall'ASL al server verifica contenente il codice della tessera sanitaria e il referto associato al Green Pass.

```
typedef struct {  
    char ID[ID_SIZE];  
    char report;  
} REPORT;
```

---

## Descrizione Architettura

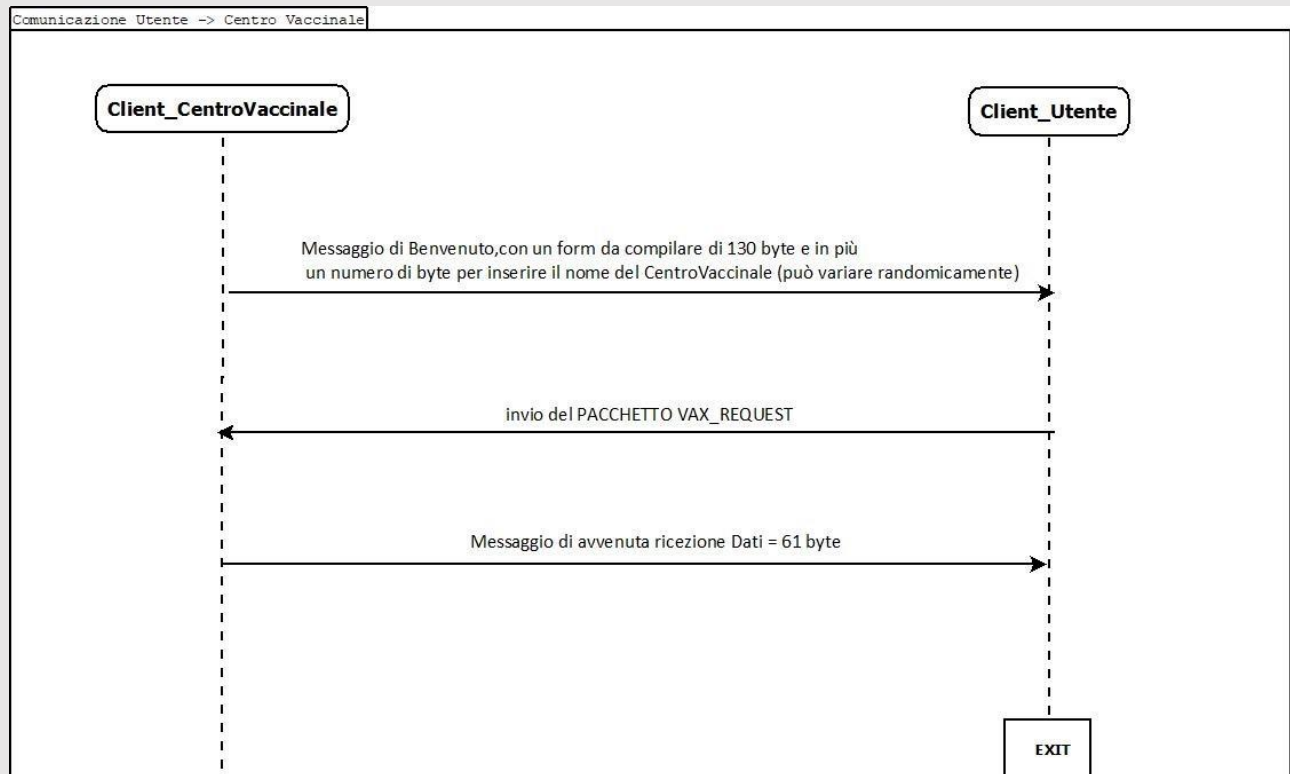
Il modello di programmazione scelto per la realizzazione del progetto è quello **Client-Server**. Il server utilizzato è concorrente, cioè è in grado di fornire servizi a più client contemporaneamente; esso è realizzato tramite l'uso della systemcall *fork()* che crea N processi figli tanti quante sono le connessioni dei client che vengono accettate dai server. Il padre gestirà il descrittore *listenfd* per accettare le richieste di connessione, mentre un figlio gestirà il descrittore restituito dalla *connect()*, cioè *connfd* per fornire servizi al client connesso.



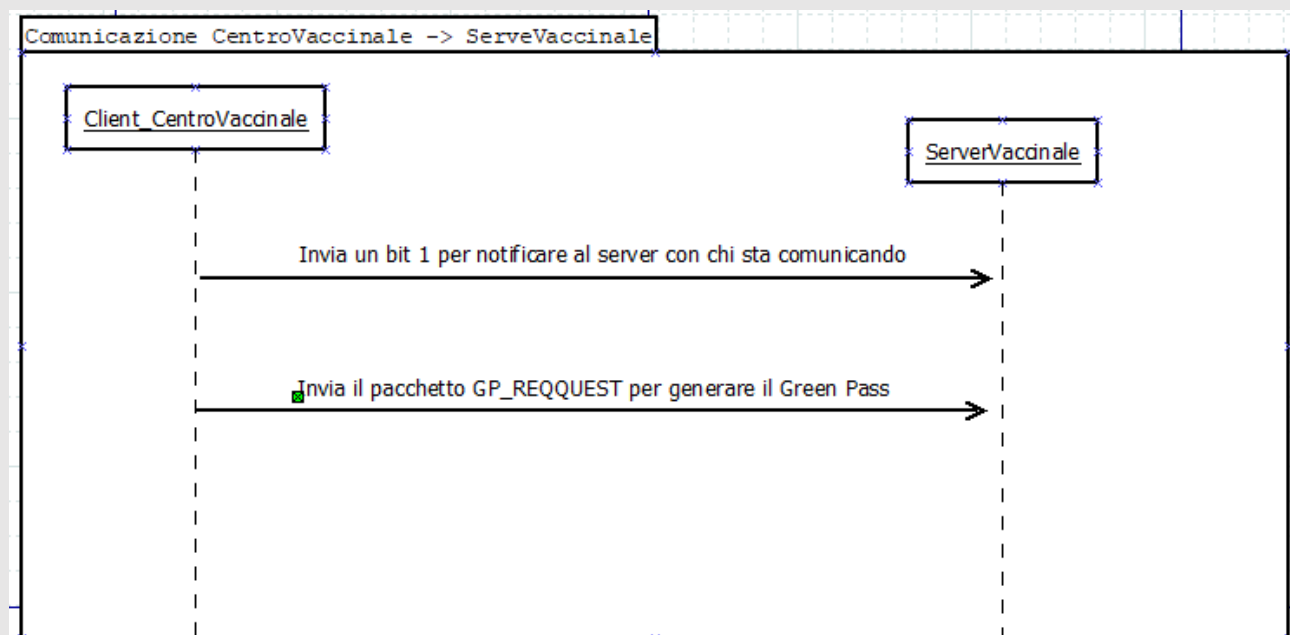
## Descrizione del protocollo applicazione

Il protocollo utilizzato è il TCP Socket, prima di trasmettere dati deve stabilire una connessione.

### Comunicazione Utente -> CentroVaccinale

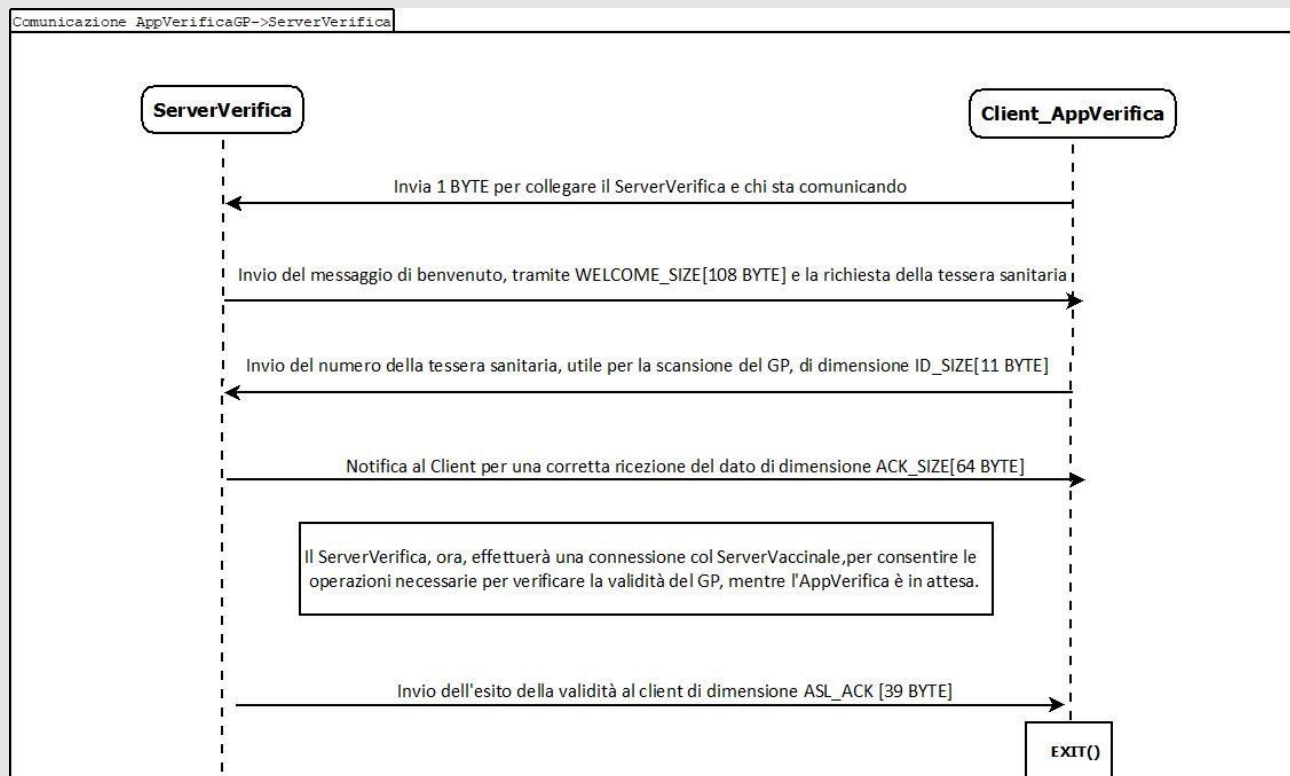


### Comunicazione CentroVaccinale -> ServerVaccinale



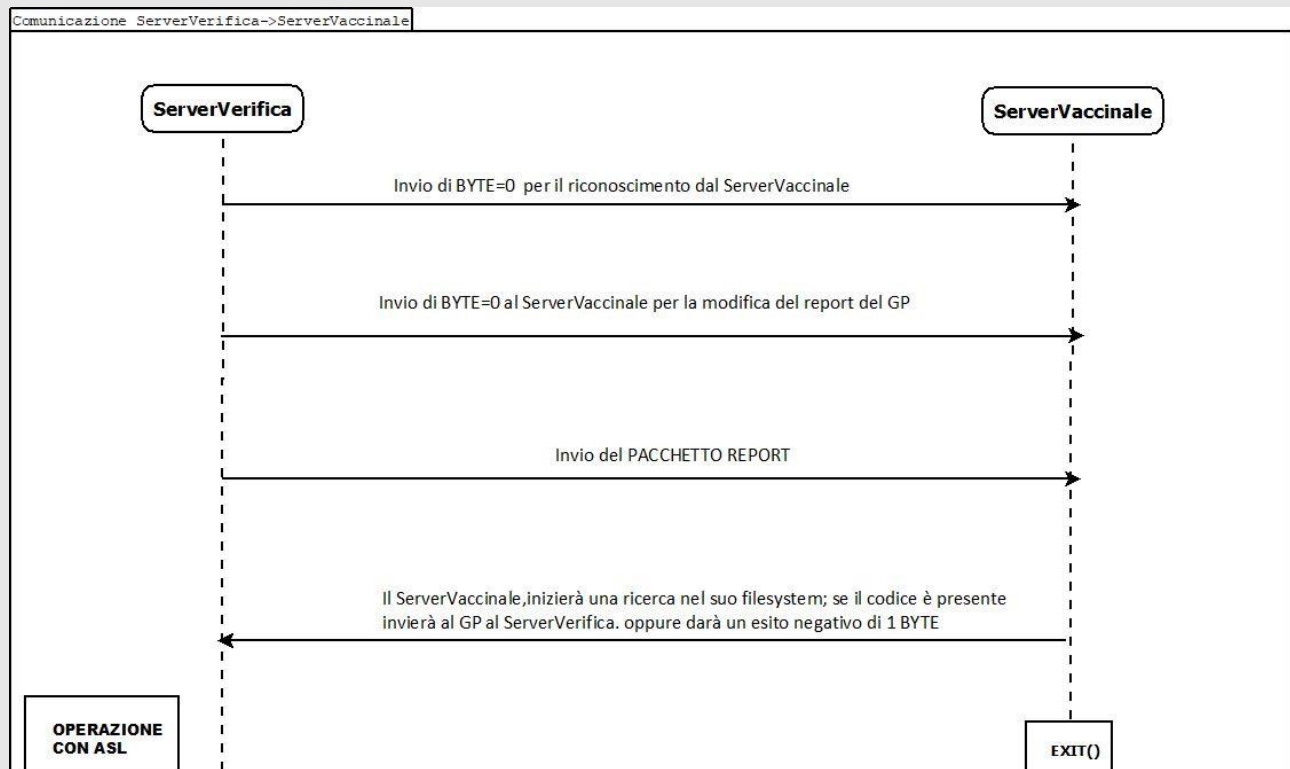


## Comunicazione AppVerifica -> ServerVerifica

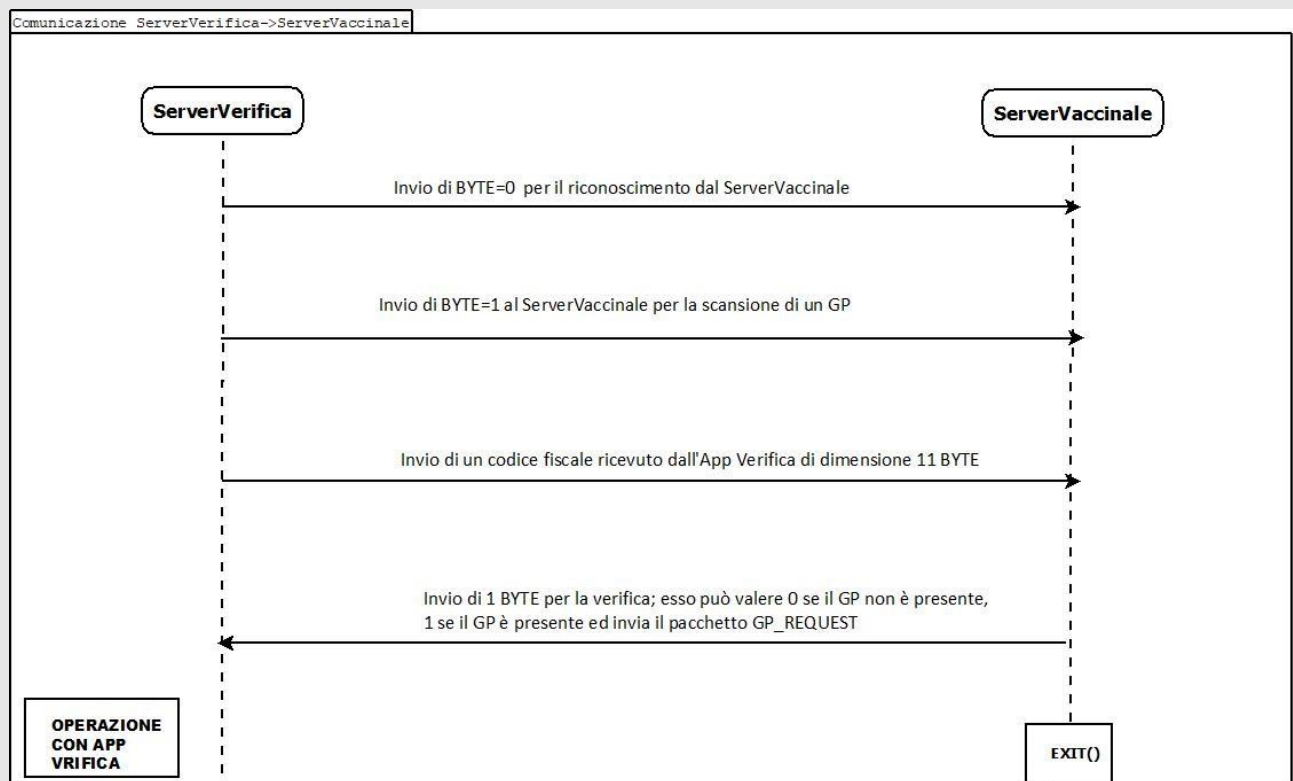


## Comunicazione ServerVerifica -> ServerVaccinale

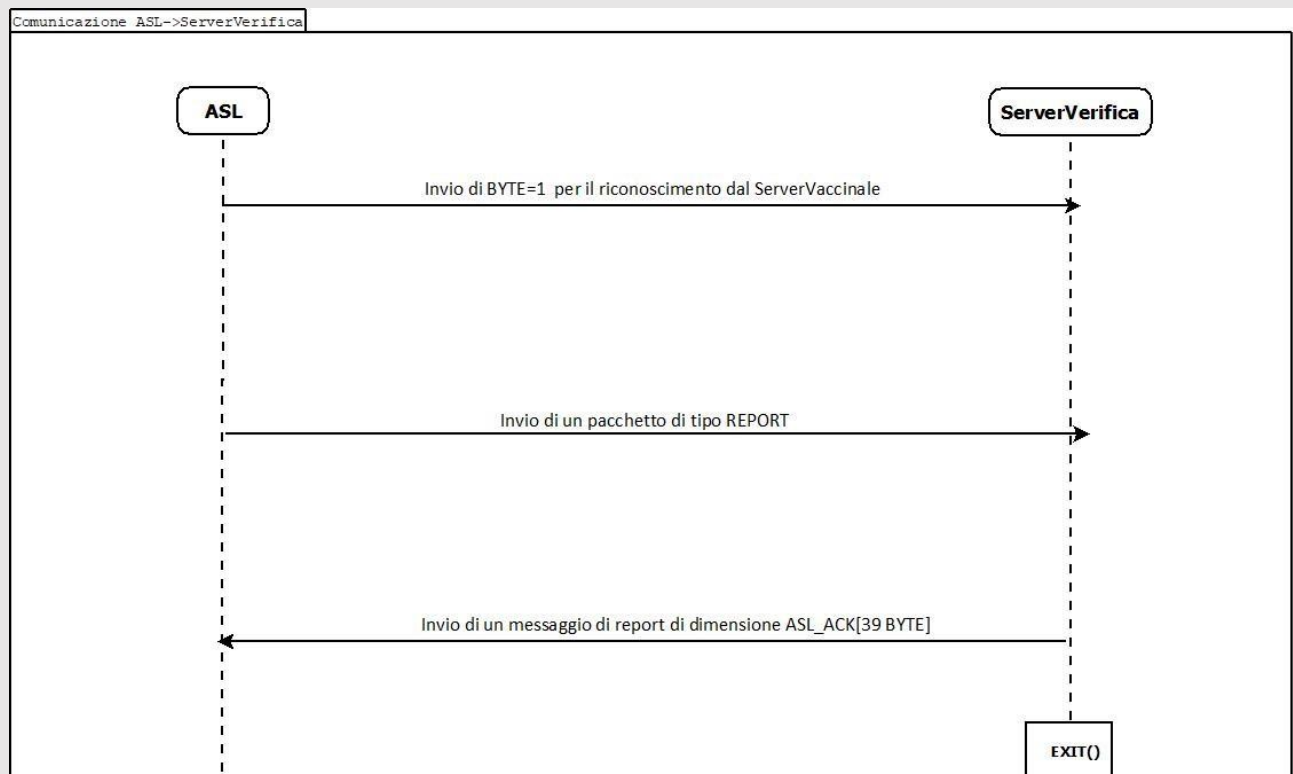
ASL



## APP VERIFICA



## Comunicazione ASL -> ServerVerifica



## Manuale Utente – Compilazione

- Per compilare Utente, entrare nel terminale e dirigersi nella directory del sorgente e digitare : **gcc -o Utente Utente.c**
- Per compilare CentroVaccinale, entrare nel terminale e dirigersi nella directory del sorgente e digitare : **gcc -o CentroVaccinale CentroVaccinale.c**
- Per compilare ServerVaccinale, entrare nel terminale e dirigersi nella directory del sorgente e digitare : **gcc -o ServerVaccinale ServerVaccinale.c**
- Per compilare ServerVerifica, entrare nel terminale e dirigersi nella directory del sorgente e digitare : **gcc -o ServerVerifica ServerVerifica.c**
- Per compilare AppVerifica e, entrare nel terminale e dirigersi nella directory del sorgente e digitare : **gcc -o AppVerifica AppVerifica.c**
- Per compilare ASL, entrare nel terminale e dirigersi nella directory del sorgente e digitare : **gcc -o ASL ASL.c**

---

## Manuale Utente – Esecuzione

Per un corretto ordine di esecuzione si consiglia l'avvio nel seguente ordine:

- Per eseguire Utente, digitare: **./ServerVaccinale**
- Per eseguire Utente, digitare: **./CentroVaccinale**
- Per eseguire Utente, digitare: **./ServerVerifica**
- Per eseguire Utente, digitare: **./Utente localhost**
- Per eseguire Utente, digitare: **./AppScansione**
- Per eseguire Utente, digitare: **./ASL**

---

## Simulazione

Per la simulazione del progetto useremo 6 terminali. Per prima cosa lanciamo il CentroVaccinale, ServerVerifica e ServerVaccinale che resteranno sempre in comunicazione tra loro per accettare e gestire le connessioni e i servizi. Nella primo screen lanciamo il programma UTENTE, che si conatterà al CentroVaccinale, compilerà un modulo e lo invierà a quest'ultimo. Successivamente il CentroVaccinale invierà il pacchetto al ServerVaccinale, che creerà un file che ha come nome il numero di tessera sanitaria dell'utente e al suo interno vi è il periodo di validità del Green Pass.

```
aulainfo@aulainfo-VirtualBox:~/Scrivania/ProgettoRetiSperanzaMaurizio$ ./Utente localhost
*Benvenuto nel centro vaccinale di Torino***
Inserisci nome, cognome e numero di tessera sanitaria.

Inserisci nome: Maurizio
Inserisci cognome: Speranza
Inserisci codice tessera sanitaria: 12345ABCDE
I tuoi dati sono stati correttamente inseriti in piattaforma
```

Nel secondo screen abbiamo la finestra del CentroVaccinale che è in attesa di richieste di vaccinazione. Quando ne riceve una, invia il modulo e stampa il contenuto di questo, inoltre mostrerà il periodo di validità del GreenPass.

```
aulainfo@aulainfo-VirtualBox:~/Scrivania/ProgettoRetiSperanzaMaurizio$ ./CentroVaccinale
In attesa di nuove richieste di vaccinazione
In attesa di nuove richieste di vaccinazione

Dati ricevuti
Nome: Maurizio
Cognome: Speranza
Numero Tessera Sanitaria: 12345ABCDE

La data di inizio validità del green pass e': 26:04:2023
La data di scadenza del green pass e': 26:07:2023
```

Dall'altro lato abbiamo l'APP per scansionare il Green Pass:

```
aulainfo@aulainfo-VirtualBox:~/Scrivania/ProgettoRetiSperanzaMaurizio$ ./AppVerifica
*Benvenuto nel server di verifica*
Inserisci il numero di tessera sanitaria per verificare la sua validità
Inserisci codice tessera sanitaria: 12345ABCDE
numero di tessera correttamente ricevuto
Convalida in corso..
GP valido
```

Dalla scansione precedente si comprende che il certificato verde è valido. Proviamo adesso a vedere come opera l'ASL: rendiamo il Green Pass creato invalido, comunicando al ServerVerifica la positività dell'utente al covid-19.

```
aulainfo@aulainfo-VirtualBox:~/Scrivania/ProgettoRetiSperanzaMaurizio$ ./ASL
*ASL*
Immettere un numero di tessera sanitaria ed il referto di un tampone per invalidare o ripristinare un GP
Inserisci codice tessera sanitaria: 12345ABCDE
Inserire 0 [GP Non Valido]
Inserire 1 [GP Valido]
: 0

Invio richiesta di sospensione GP
*Operazione avvenuta*
```

Proviamo adesso a vedere se l'operazione è andata a buon fine, scansionando il GP con l'APP di verifica:

```
aulainfo@aulainfo-VirtualBox:~/Scrivania/ProgettoRetiSperanzaMaurizio$ ./AppVerifica
*Benvenuto nel server di verifica*
Inserisci il numero di tessera sanitaria per verificare la sua validità
Inserisci codice tessera sanitaria: 12345ABCDE
numero di tessera correttamente ricevuto
Convalida in corso..
GP non valido, uscita
```

In questo caso noteremo che l'operazione è andata a buon fine e il Green Pass è stato invalidato.