Stefano Leonardi:

Punto 1) Propose a description of the system architecture:

Tutto il servizio è basato su un server scritto in node.js, questo grazie al framework express, effettua il routing di alcuni url. Sulla loro base, invoca delle funzioni scritte in javascript che interagiscono con il database di mongoDB. Tutto il servizio ha host su heroku, che usando https per la comunicazione garantisce la segretezza dei dati scambiati con l’utente.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Punto 4) Describe how you would handle the authentication and the authorization to access data:

Prima di tutto andrei a creare una tabella di utenti, in cui ognuno si può inserire inviando tramite un API, le proprie credenziali, siano queste nome utente, password, e altri dati circostanziali utili, anche al fine di un eventuale recupero informazioni perse da parte dell’utente.

In risposta viene inviato un codice creato al momento della richiesta che identifica univocamente l’utente. Questo codice dovrà essere in seguito usato dall’utente per garantire la propria identità e per tanto sarà sua cura mantenerlo segreto. Inoltre ad ogni utente verrà assegnata anche un etichetta che dirà a quali servizi ha accesso, questa non è univoca, ma al contrario ve ne sono poche.

Tale etichetta altro non è che l’identificatore di una riga della tabella dei permessi, in questa tabella si vedono sulle rige tutte le etichette, mentre sulle colonne tutti i possibili permessi come, possibilità di caricare un tweet, ricerca di testi contenenti parole, ricerca di tweet privati di altri utenti e tutto ciò che può venire in mente. All’incrocio vi è un semplice valore booleano che dice se tale utente ha quel permesso o meno.

Ogni volta che un utente vuole fare qualcosa si verifica se è autenticato, e sin caso si provvede a chiedergli l’identificatore, dopo di che si controlla se ha i permessi necessari per fare ciò che chiede.

Ogni volta lo scambio del codice univoco rimane segreto, in quanto il server lavora con https, e quindi non vi è il rischio che il codice venga letto e compreso da estranei.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_