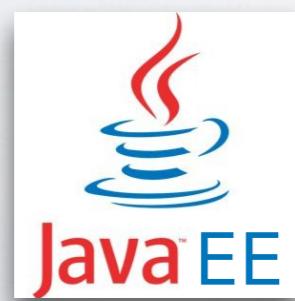
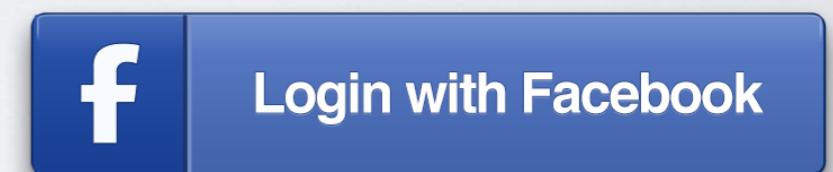


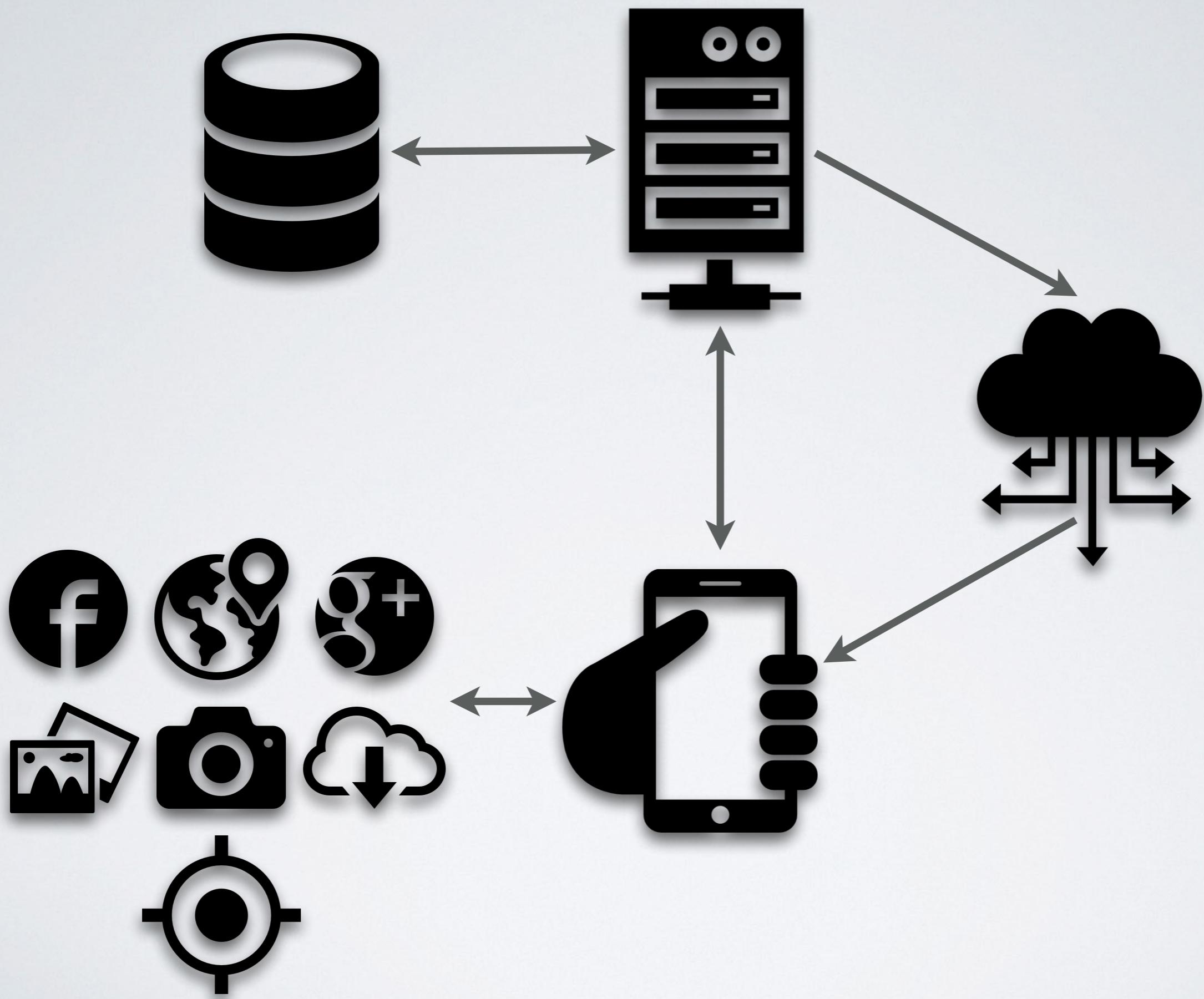


NATTER



{ JSON }



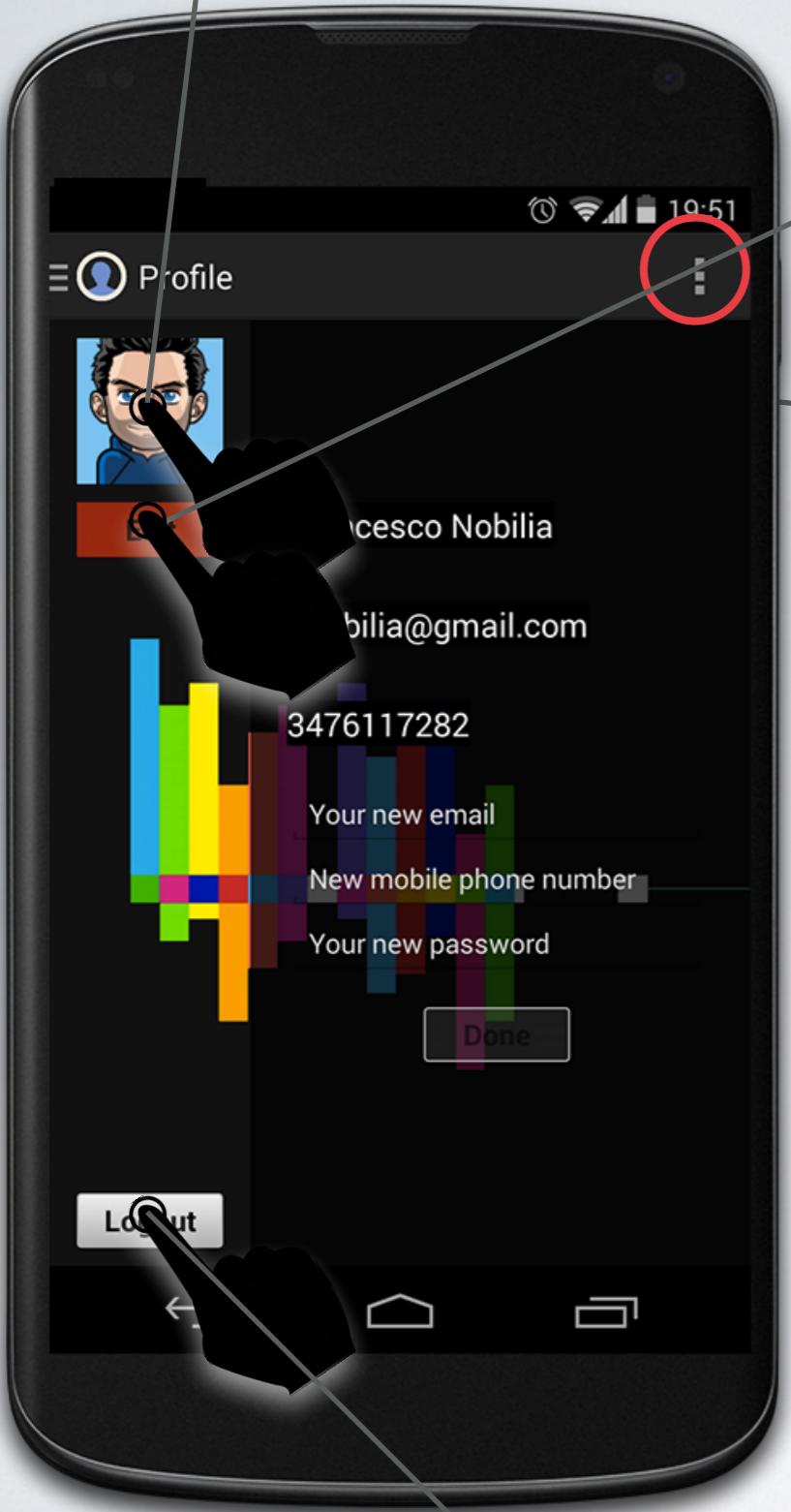


At first boot there are two **Services** that start

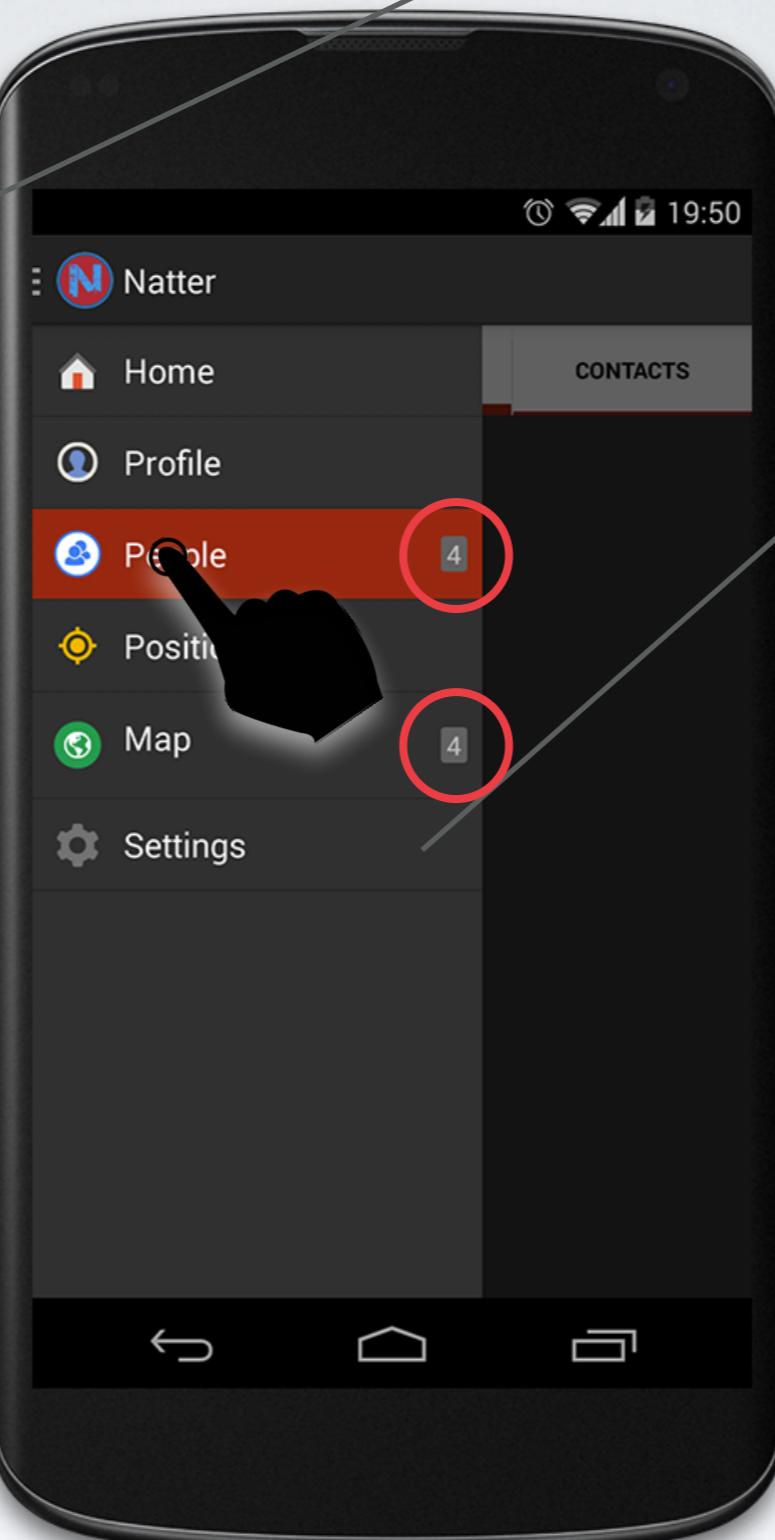
- the *first* one is a one shot service to align position and contacts
- the second one reacts to **Action_Boot_Complited** to interact with the server during all the operation's time of the phone



By clicking the **image** is possible to change the profile image

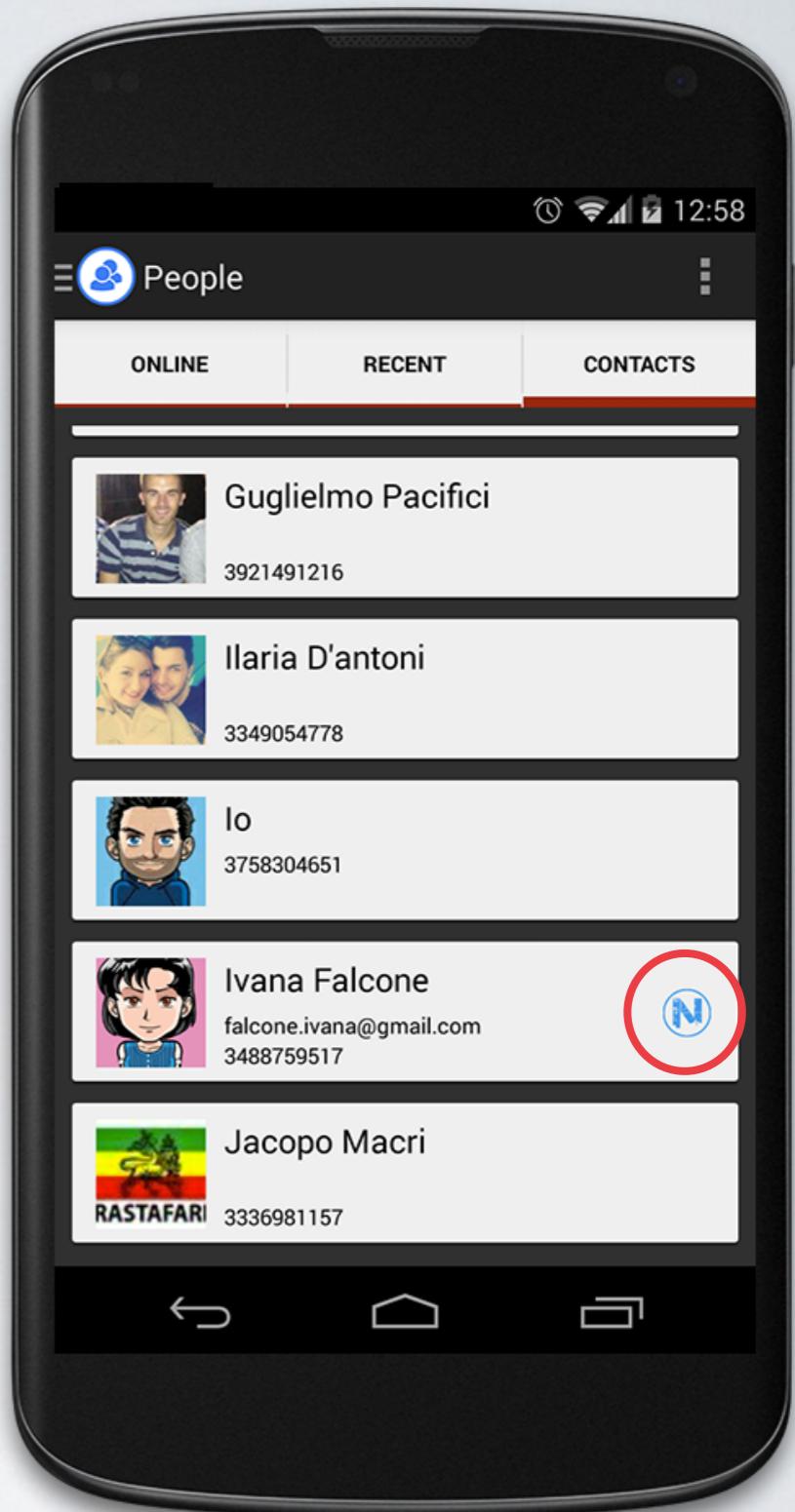
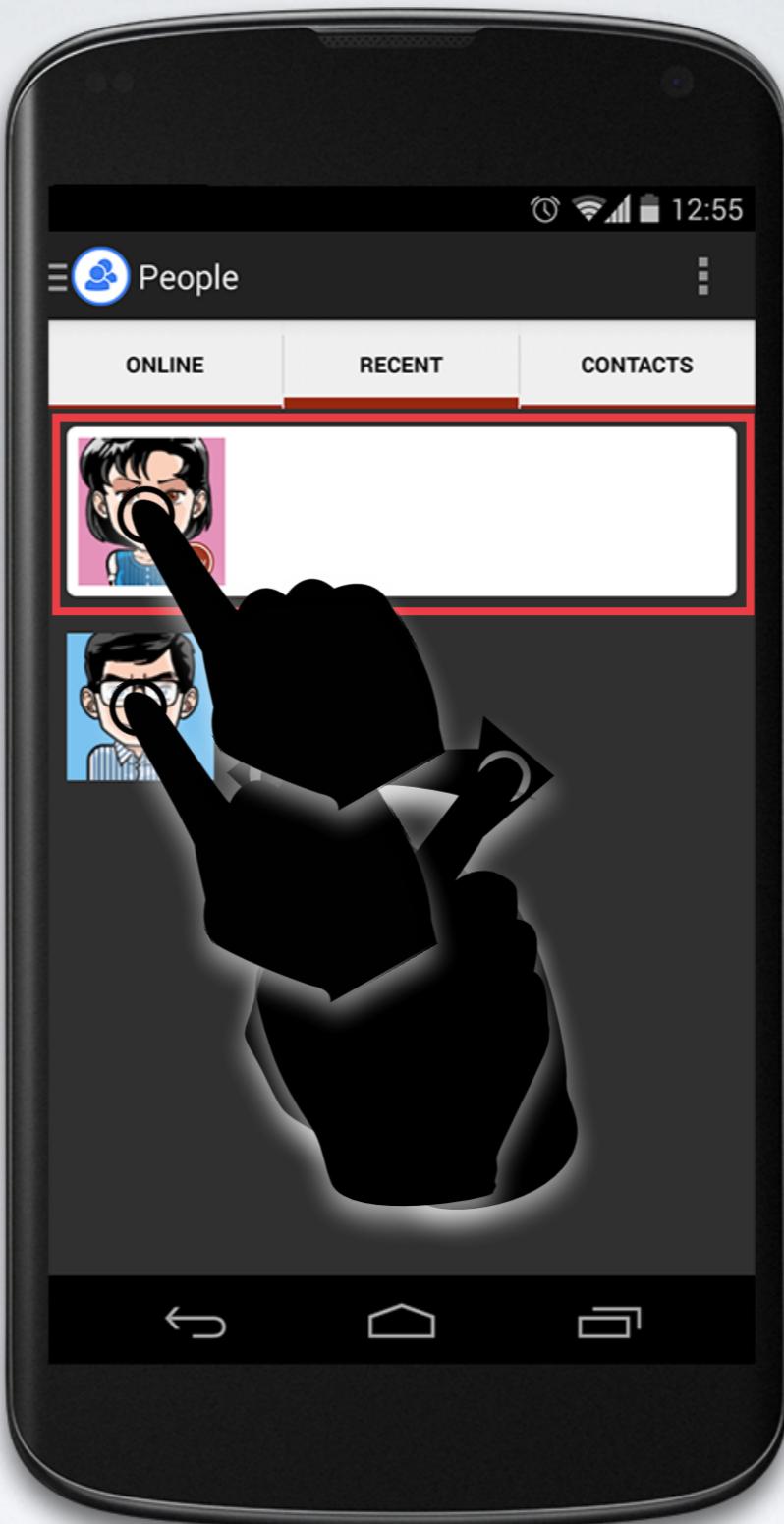
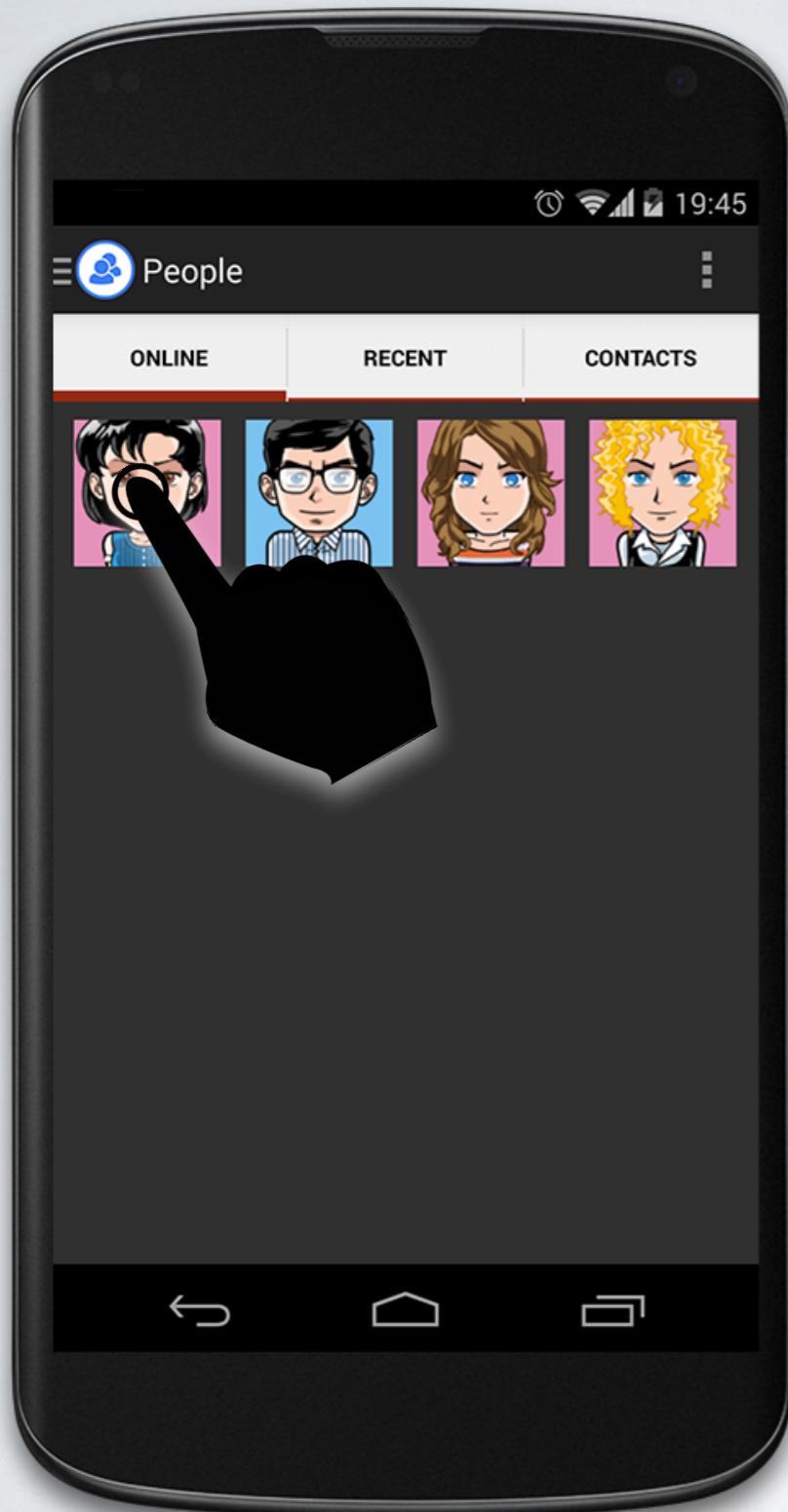


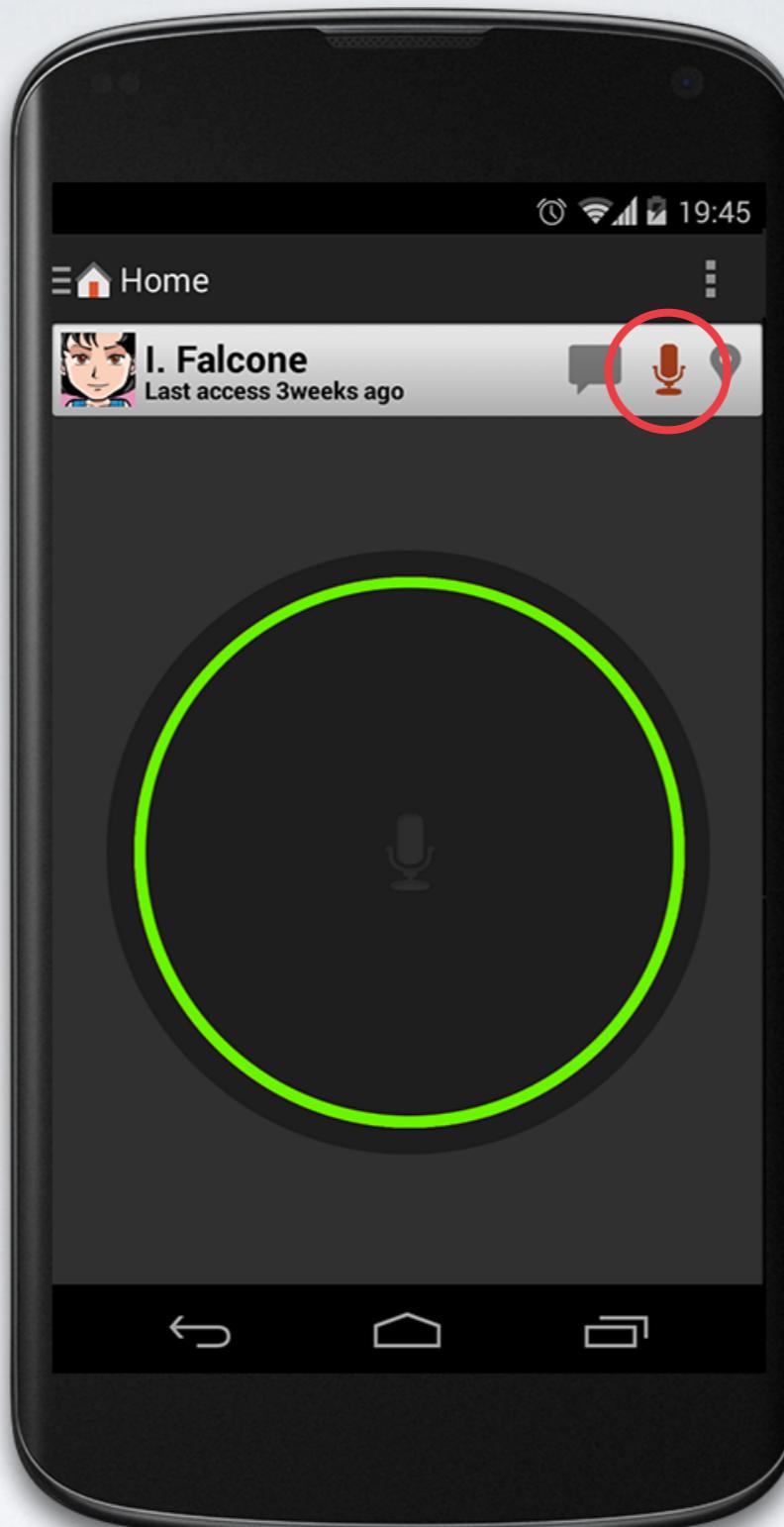
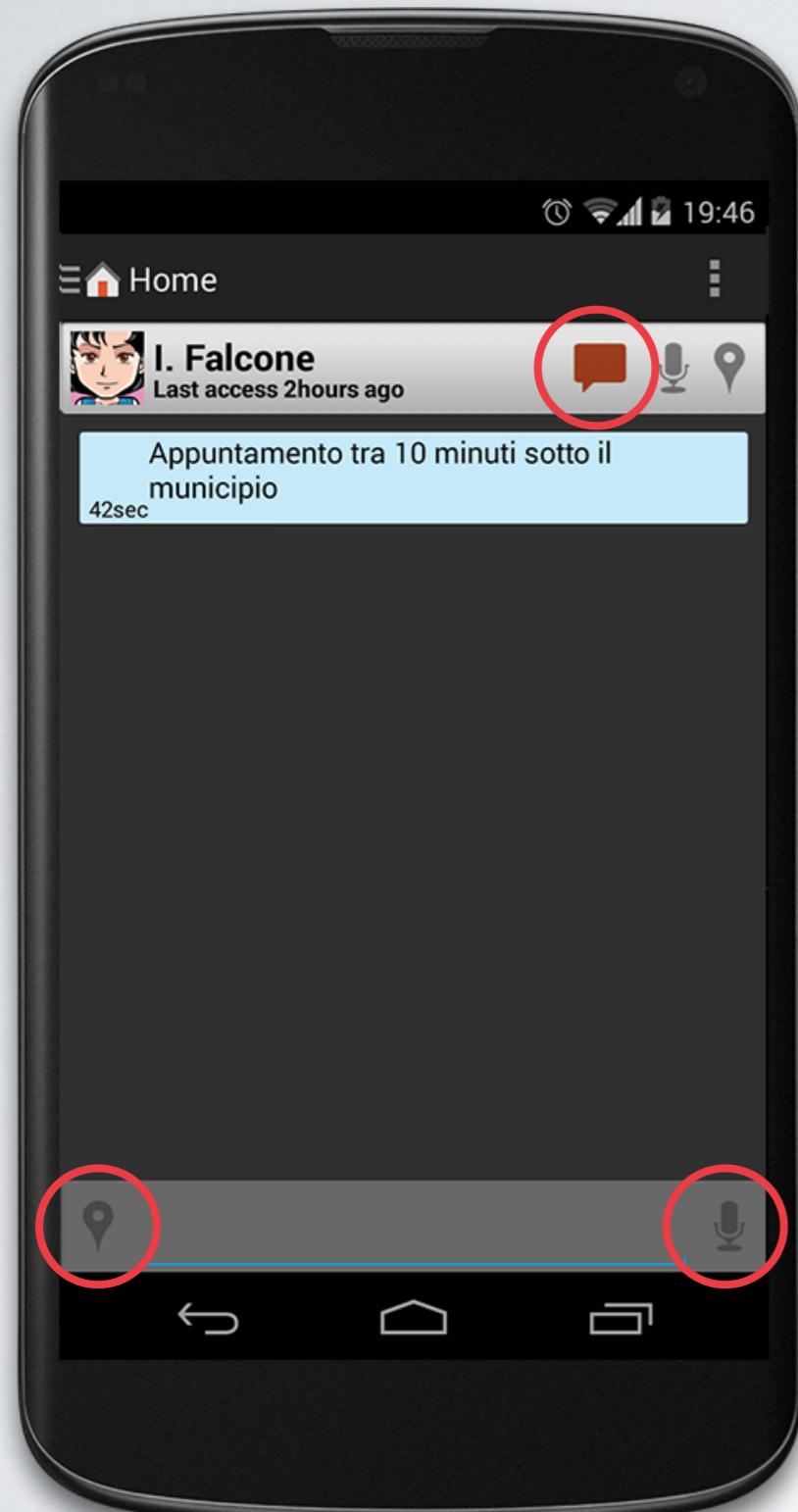
By clicking **Edit** is possible to change the profile info



By clicking **Logout**, app deletes some information from **local DB** and from Server

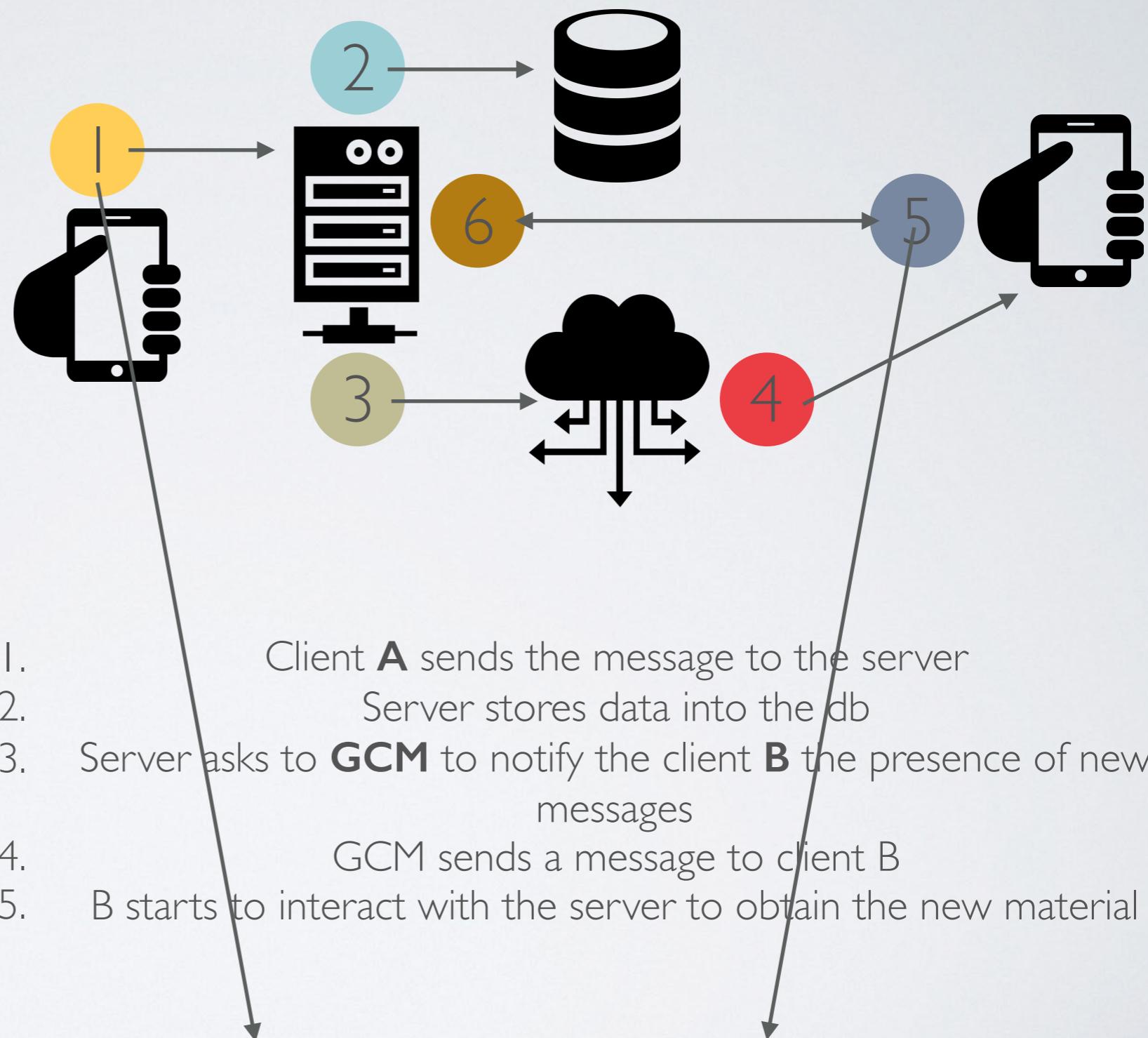






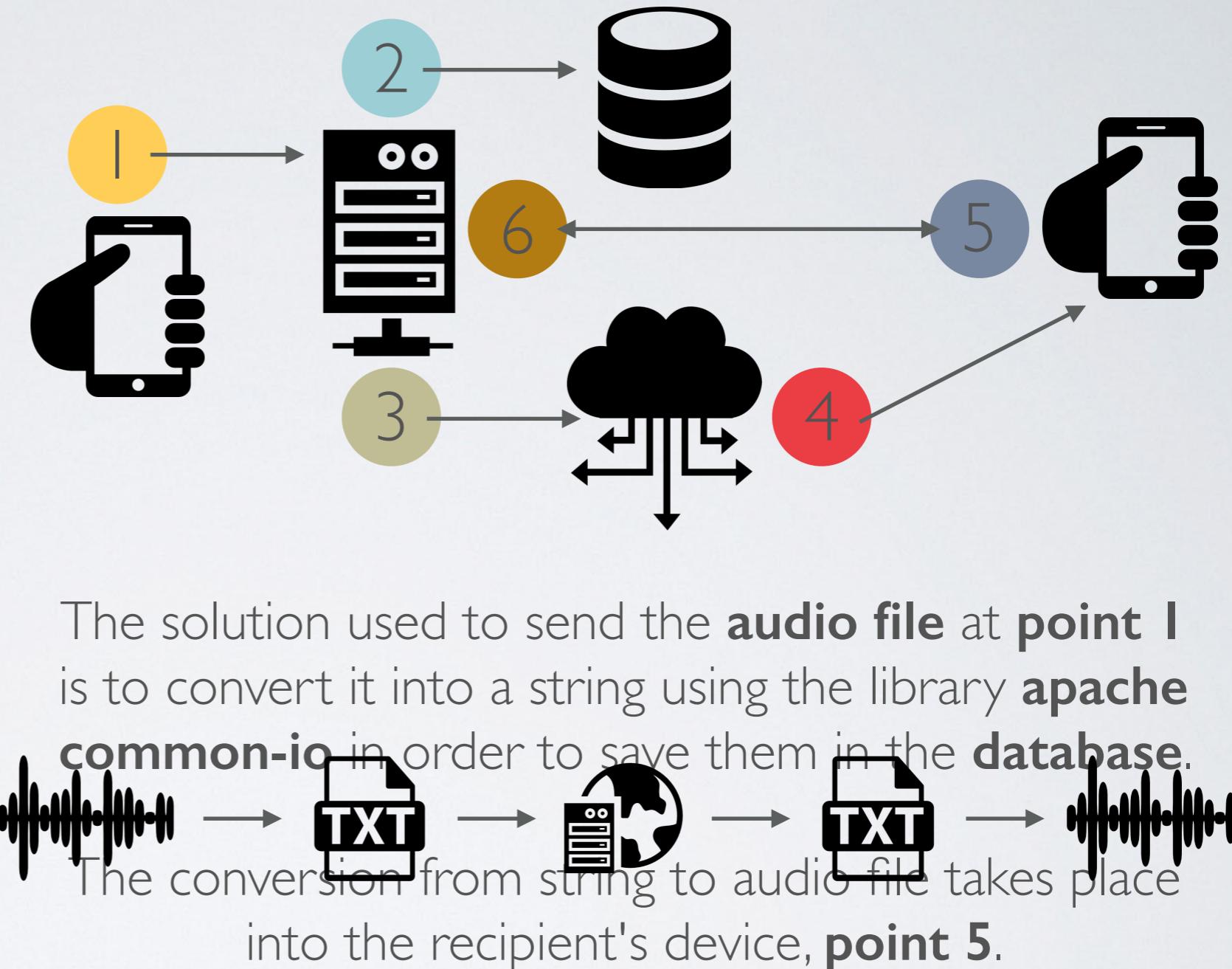
A **Runnable** takes care to update this information every 5 seconds

- Online
- Writing
- Interval from last access



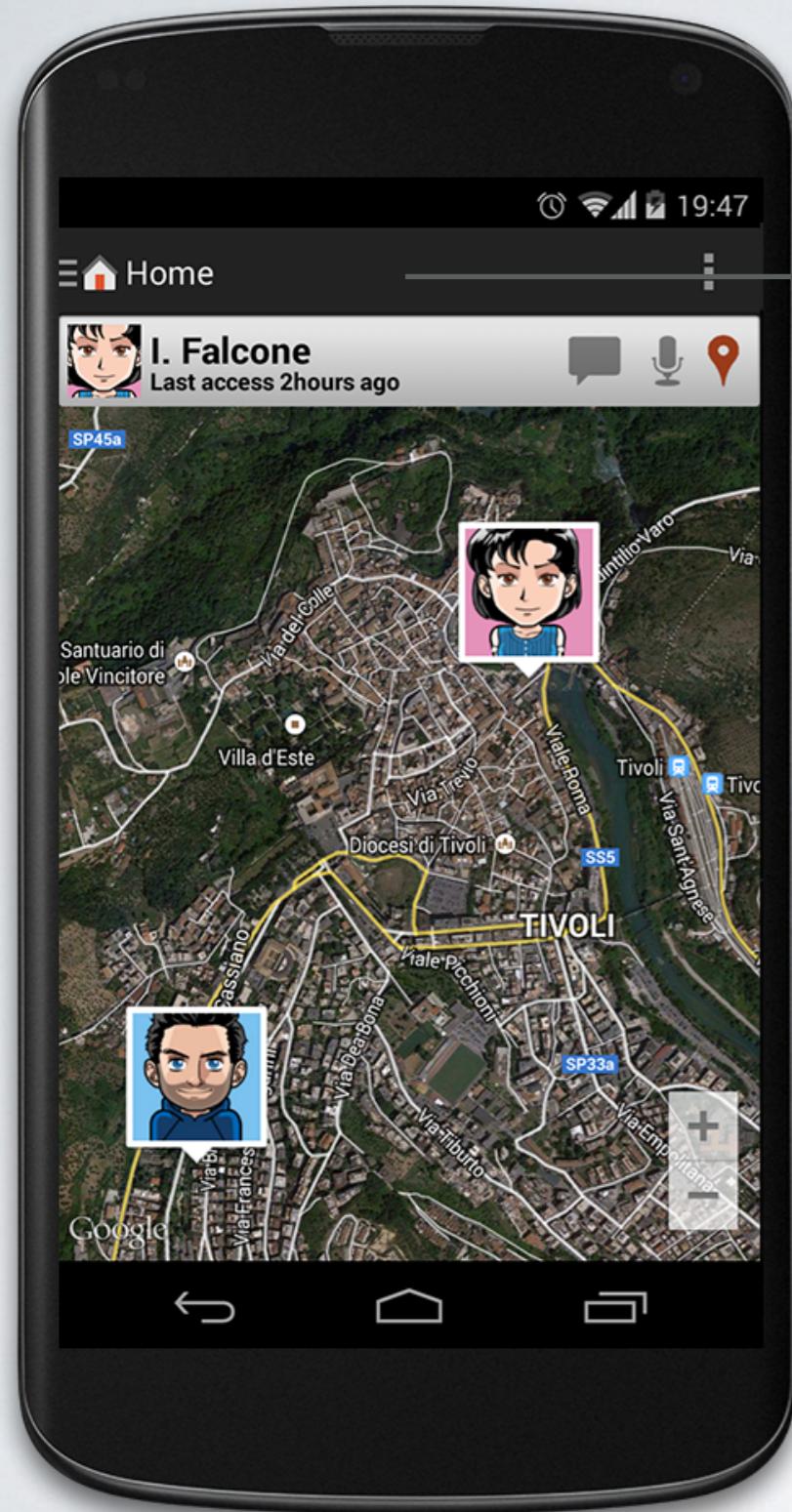
All interactions with the server take place in separate processes to obtain a fluid graphics

Here everything works as in the previous slide

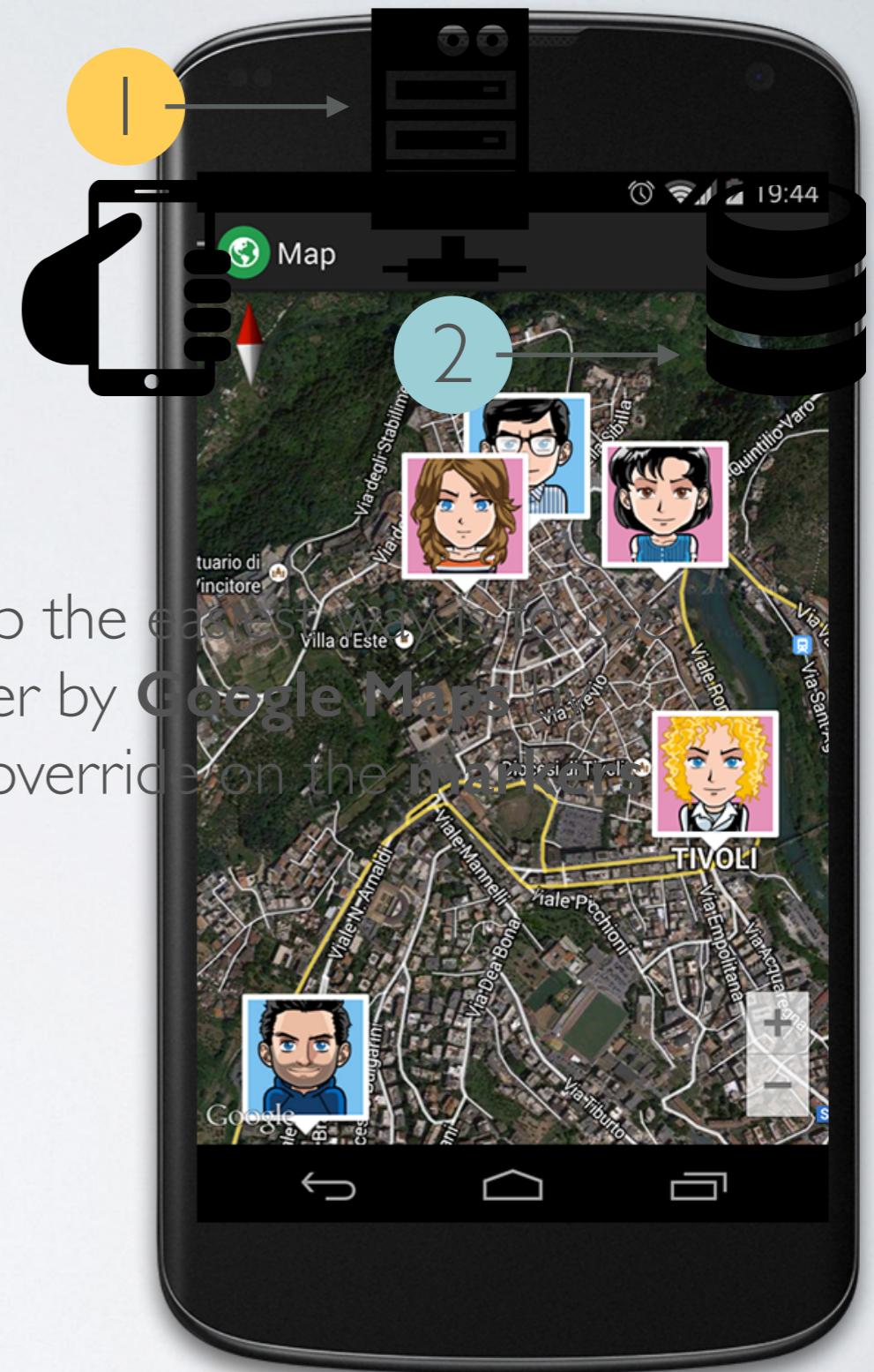
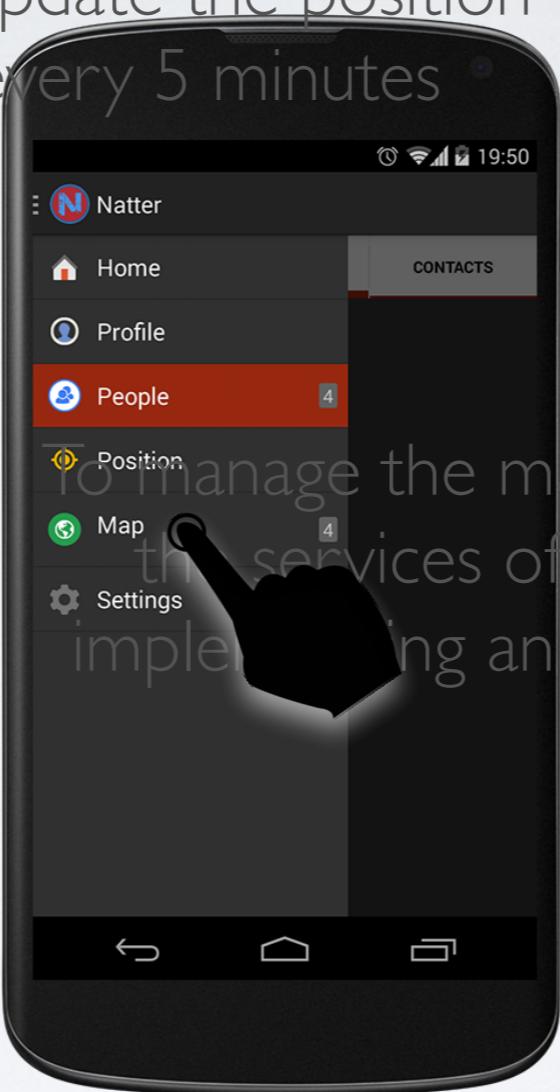


The solution used to send the **audio file** at **point 1** is to convert it into a string using the library **apache common-io** in order to save them in the **database**.
The conversion from string to audio file takes place into the recipient's device, **point 5**.

Use strings allows the implementation of a **encrypted** communication system in a simple way



A **Runnable** takes care
to update the position
every 5 minutes





Natter was presented as prototype at **contest**

*Sviluppo di idee progetto per una nuova generazione di App
“App On”*

proposed by **Regione Lazio** and **European Union**

