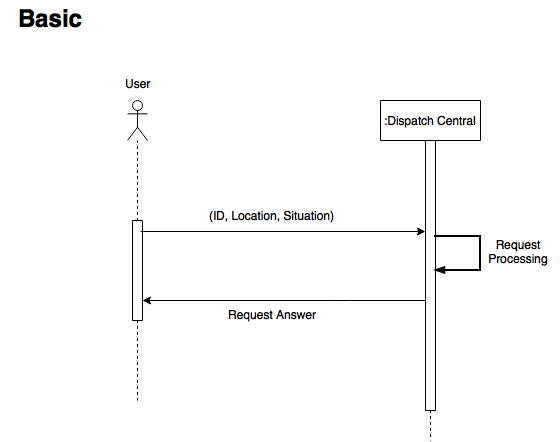
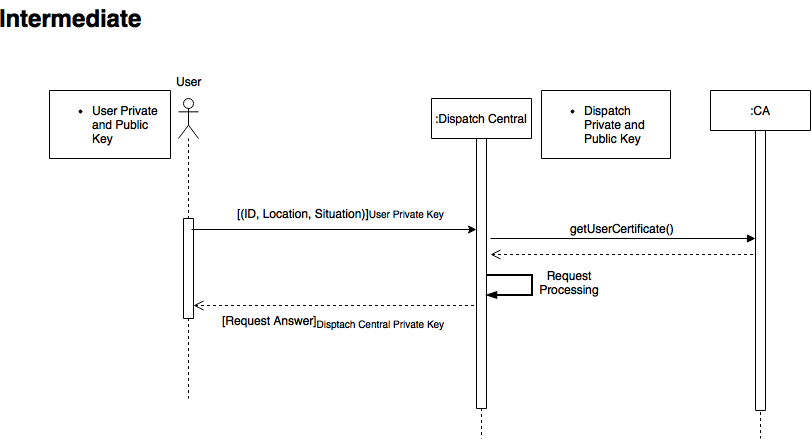
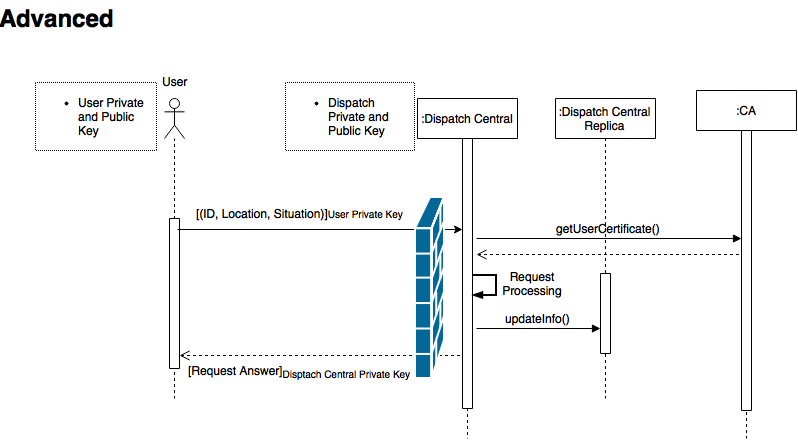
# Proposed Solution



In the Basic solution we assume that the communications between the User and the Dispatch Central are secure. The main goal of this sprint is to build the foundation and the architecture to facilitate the implementation of the security parts. There are no keys (and certificates) involved in this step so non-repudiation will not be the goal here. We will have a centralized Dispatch that will receive all the requests from the Users. Auditing mechanisms will be considered in this phase and for that we will use logs on the Central side. Everytime a request is received it is registered in a log for the auditing personal be able to review. Each entry will have a timestamp and the contents of the message (ID, Location, Situation). After processing the request, it sends the confirmation message that an ambulance is on the going (if the user have no faults associated).



In the second phase, we will create manually all the keys and certificates (self-signed). Each entity will have his own set of keys, but the certificates they will have to request to the CA (to get the public key of the issuer). This will give a structure to implement non-repudiation taking the assumption that the private keys are never stolen by some sort of malware or someone accessing the physical machine and stealing them. Each message sent by the User will be signed with his own private key. With the previous assumption we can guarantee that each message sent was really him sending the message. On the Dispatch Central, it will retrieve the user public key from the certificate that was requested after receiving the message and then decipher the message. The procedure is equal on the message flow is in reverse (Central to User).



On the last phase of the project, we will add a replica for eventual failures on the primary server (DoS, crash, etc.) and a firewall that will be able to detect if there are any requests that are duplicate, redirecting connections to secure ports and analyzing packet contents.