

HACKATHON 17-18 NOVEMBRE 2012

La Gestione delle EMERGENZE



by MegaDevs

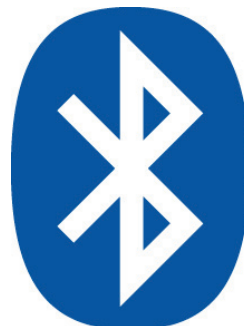
LA QUESTIONE_

Un'emergenza non è sempre prevedibile e, nella maggior parte dei casi, rende inefficienti connessioni e comunicazioni.

Come possiamo ovviare a questo rischio? E gestire una situazione di questo tipo?

L'INTUIZIONE_

Utilizzando un SISTEMA che agisca PRIMA, DURANTE
e DOPO l'evento.



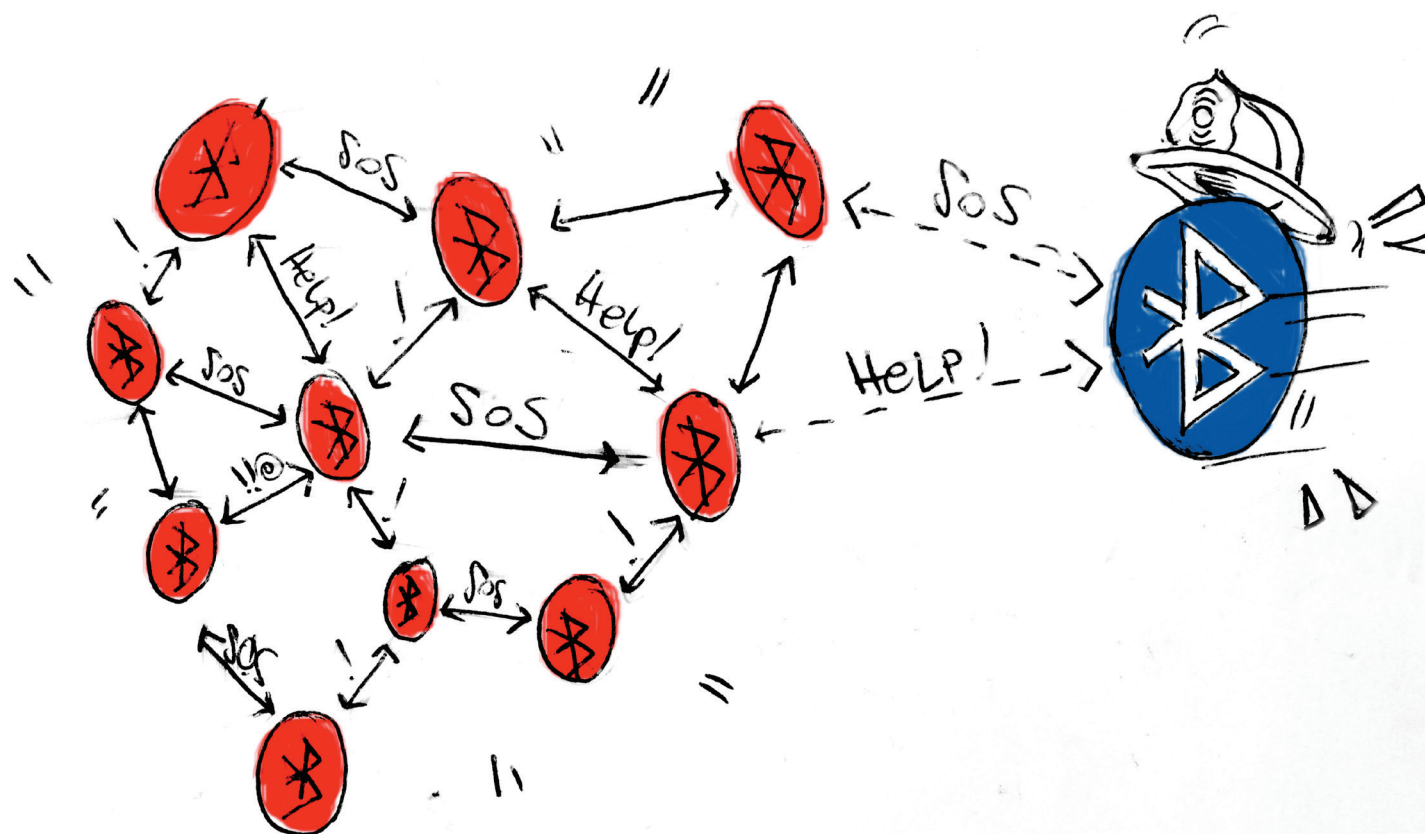
LA SOLUZIONE pt I _ Osservazione con telecamere a circuito chiuso.



Le telecamere a circuito chiuso, impostate per il face detection, monitorano la presenza di persone in un dato edificio.

Obiettivo: valutare a priori la necessità di intervento in un luogo.

LA SOLUZIONE pt II _ Segnalazioni via network Bluetooth.



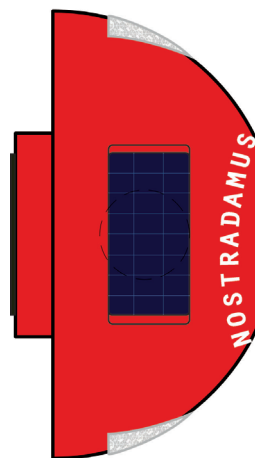
In caso di emergenze, si attiva un sistema di networking Bluetooth, collegando tra loro i dispositivi che possiedono l'applicazione. Cellulari piuttosto che altri tipi di dispositivi, ad esempio adatti all'uso per persone anziane o invalide.

Obiettivo: andare oltre l'ostacolo della caduta del sistema telematico e comunicare la propria presenza.

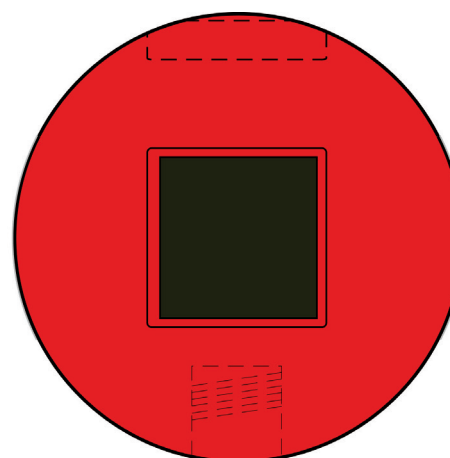
LA SOLUZIONE pt III _ Comunicazione stato emergenza via NFC.



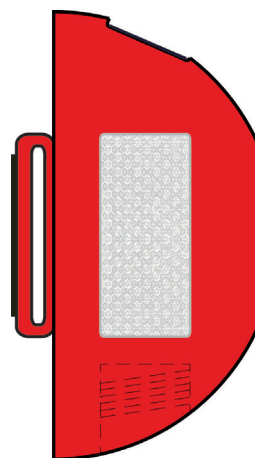
Fronte



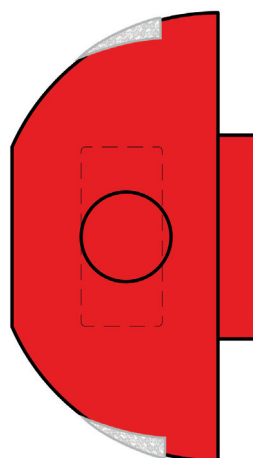
Sopra



Retro



Lato



Sotto

Fornire alle protezioni civili dei dispositivi mobili e versatili con la possibilità di essere depositati in luoghi in cui l'emergenza ha portato danni particolari.

Obiettivo: fornire un sistema di comunicazione, promemoria e avvertenza utile alla comunicazione tra le unità di soccorso, portando ad un intervento più mirato.

GRAZIE DELL'ATTENZIONE!

Hack's repo: <https://github.com/MegaDevs/Nostradamus>

Rekognition API: <https://rekognition.com/>

BayesBox (lib): <https://github.com/bonnyfone/BayesBox>

Team: <http://megadevs.com/>

Android | NFC | Bluetooth | Wi-Fi | Mesh Networks | Opportunistic communication | Face detection |
Google AppEngine | Bot crawling | Probabilistic decisional system | Rails+jQuery | Device hacking |
API hacking | NAT Avoidance