

Alla mia famiglia . . .

Introduzione

Questa è l'introduzione. Discorso sull'elaborato in generale, smart city, IoT, D2D...

Indice

Introduzione	i
1 Primo Capitolo	1
1.1 Scopo e problematiche	1
1.2 Architettura generale	2
1.3 Organizzazione delle informazioni	4
1.4 Altra Sezione	5
1.4.1 Altra SottoSezione	6
1.5 Altra Sezione	6
1.6 Altra Sezione	6
1.6.1 Listati dei programmi	6
Conclusioni	7
A Prima Appendice	9
B Seconda Appendice	11
Bibliografia	13

Elenco delle figure

1.1	Rappresentazione dell'Architettura generale	3
-----	---	---

Elenco delle tabelle

1.1	legenda elenco tabelle	6
-----	----------------------------------	---

Capitolo 1

Primo Capitolo

Il seguente capitolo ha lo scopo di illustrare in dettaglio il goal del progetto, le problematiche che vi stanno dietro ed il processo di progettazione su cui è stato necessario soffermarsi, per la successiva realizzazione ed implementazione. La mission, descrivendola in pochi concetti, è sostanzialmente quella di disseminare le informazioni, relative al concetto di Smart Parking, in maniera totalmente distribuita e senza alcun appoggio di una struttura centralizzata. Questo concetto diventa innovativo dal momento in cui non si passa dalla rete cellulare o da qualche gateway verso Internet per la comunicazione tra dispositivi, ma verranno utilizzate soltanto le tecnologie e apparecchiature presenti nei più comuni smartphone in commercio. Dettagli tecnologici ed implementativi saranno comunque discussi nel prossimo capitolo. Di seguito verranno illustrati soprattutto le principali componenti che compongono il sistema, descrivendone funzionalità, utilità ed integrazione con le restanti componenti.

1.1 Scopo e problematiche

Come già accennato e discusso in precedenza lo scopo principale è quello di propagare informazioni, tra dispositivi mobili, in maniera completamente distribuita, senza l'appoggio di agenti centralizzati. Il device utente, tramite

la componente di Activity Recognition presente nel software, compie un evento di parcheggio o rilascio, dopo di ch  tiene traccia dell'evento salvandolo in locale nella memoria. Le informazioni campionate sopra descritte, saranno l'oggetto della comunicazione tra i device che potranno fare assunzioni su eventuali zone dense di parcheggi liberi. Pi  gli utenti restano allineati e sincronizzati su questi dati, pi  restano aggiornati e consapevoli della situazione parcheggi nei dintorni. Per semplicit  concettuale la citt  viene divisa in celle(identificate con id univoci), permettendo cos  alla logica di disseminare informazioni relative alla cella in questione e a quelle adiacenti. Un altro aspetto non meno importante di quelli gi  illustrati,   il fatto che il servizio deve funzionare in modo continuativo nel tempo e soprattutto in modo totalmente autonomo. Questo significa che l'utente, senza nessuna interazione con il device(mantenendolo per esempio in tasca), sincronizza comunque i propri dati con quelli di utenti circostanti, nei paraggi. La funzionalit  appena descritta, per renderla robusta, ha richiesto una fase implementativa consistente, la quale verr  illustrata tecnicamente in seguito.

1.2 Architettura generale

L'architettura generale, analizzandola da un punto di vista esterno, prevede come unici attori i device utenti. Questi muovendosi all'interno dell'ambiente citt  restano in attesa di possibili sincronizzazioni. Ogni dispositivo, che presenta il servizio di comunicazione e sincronizzazione attivo, pu  interpretare fondamentalmente due ruoli concettuali:

ROLE ACCESS POINT:

Il device funge da access point, offrendo la possibilit  a device presenti in prossimit  di agganciarsi all'HotSpot esposto. Una volta che un Client si connette stabilmente ad esso avr  l'opportunit  di aprire un canale di comunicazione e sincronizzare i propri dati con quelli del Access Point, denominato anche come Server. Inoltre l'Access Point notifica la propria presenza e in-

formazioni utili attraverso la propagazione di un beacon di servizio.

ROLE CLIENT:

Se un device si accorge della presenza di un HotSpot nelle vicinanze, grazie al beacon sopra citato, può tentare di connettersi come Client all'Access Point. Qualora riesca ad agganciarsi in modo stabile all'HotSpot verranno avviati i canali di comunicazione, su cui effettuare una sincronizzazione tra i device Client e Server.

A questo punto la topologia della rete generale può essere vista come hub di hub, ovvero tante piccole star, dove grazie al movimento inerziale dei device si riuscirà ad ottenere una propagazione dell'informazione, interconnettendo appunto le informazioni di diversi hub tra di loro, ad istanti di tempo differenti.

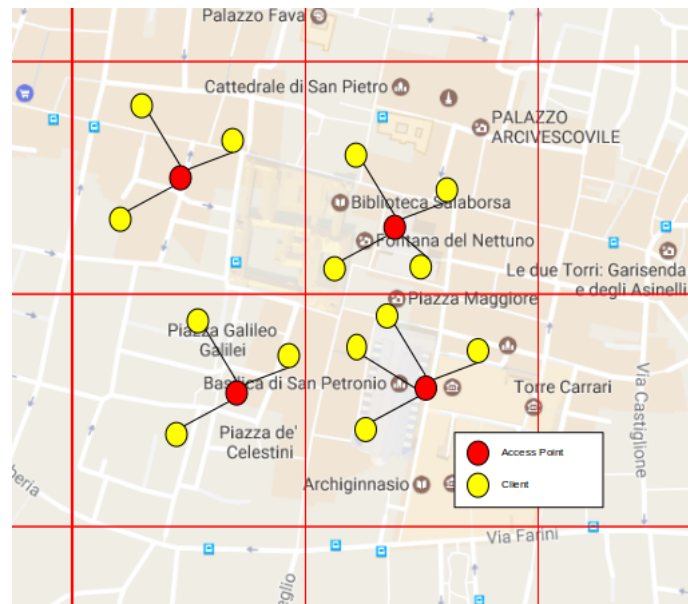


Figura 1.1: Rappresentazione dell'Architettura generale

Come si può comprendere dalla Fig. i device muovendosi continuamente all'interno dello scenario (identificato in questo caso con la griglia città), for-

meranno continuamente nuovi Hub, con attori sempre diversi. Questo fattore è determinante per il processo di propagazione delle informazioni, in quanto dati presenti in un Hub, verranno sincronizzati, in istanti successivi, con Hub diversi.

1.3 Organizzazione delle informazioni

L'organizzazione e persistenza dei city data è ovviamente una fase determinante del progetto. In fase di progettazione, si è giunti alla conclusione che non è sufficiente scambiarsi il numero di parcheggi liberi/occupati/unknown derivanti da dati statistici, in quanto ogni device/utente potrebbe avere una parziale, ma diversa, visione dello scenario. Da ciò deriva la necessità di sincronizzarsi su tutti gli eventi che sono successi recentemente in una determinata zona (cella) e quelle adiacenti. Si è optato per la realizzazione di un database di tipo relazionale composto principalmente da due tabelle, indipendenti l'una dall'altra. La prima tabella, denominata, `park_events` colleziona tutti gli eventi che gli utenti compiono o ricevono in seguito a sincronizzazioni con altri utenti. La seconda tabella `synchronizations` colleziona invece tutte le sincronizzazioni che hanno coinvolto il device in questione. Come verrà illustrato nel capitolo relativo all'implementazione, gli strumenti utilizzati si baseranno completamente sullo standard SQL, i quali permetteranno una semplice gestione di query di filtraggio, creazione, modifica, cancellazione. Entrando più in dettaglio nella composizione delle tabelle e degli attributi salvati, si mostra di seguito uno schema riassuntivo di ciò che viene messo a database.

Tabella `park_events`:

- `cell_id` : identificativo univoco della cella che la identifica all'interno della grid city. Informazione di tipo geografica.

- event : evento che può essere limitatamente PARKED o RELEASED. Indica se l'entry fa riferimento ad una sosta o alla liberazione di un parcheggio.
- timestamp : datetime che indica il momento in cui è stato performato l'evento in questione.
- mac : identificativo univoco del device che ha performato l'evento di parcheggio o rilascio.

La PRIMARY_KEY viene definita come l'aggregazione di <timestamp, mac >. L'idea che vi sta dietro è quella che un utente può performare, in un dato istante, al più un unico evento di rilascio o parcheggio.

Tabella synchronizations:

- timestamp : datetime in cui è avvenuta la sincronizzazione
- mac : id univoco dell'altro device con cui è stata possibile la sincronizzazione
- ap_role : booleano che indica se ero Access Point o meno al momento della sincronizzazione

Questa tabella oltre a fini statistici, è importante perché permette di evitare che due device si sincronizzino in continuazione o che siano sempre dietro a rincorrere eventuali sincronizzazioni. Possono essere visti come dati che permettano un minimo di salvaguardia da cicli continui o sovrapposizione di sincronizzazioni. Questo perché le logiche implementate a backend non permettono lo scambio di dati in istanti troppo vicini tra loro.

1.4 Altra Sezione

Vediamo un elenco descrittivo:

OGGETTO1 prima descrizione;

OGGETTO2 seconda descrizione;

OGGETTO3 terza descrizione.

1.4.1 Altra SottoSezione

SottoSottoSezione

Questa sottosottosezione non viene numerata, ma è solo scritta in grassetto.

1.5 Altra Sezione

Vediamo la creazione di una tabella; la tabella 1.1 (richiamo il nome della tabella utilizzando la label che ho messo sotto): la facciamo di tre righe e tre colonne, la prima colonna “incolonnata” a destra (r) e le altre centrate (c):

(1, 1)	(1, 2)	(1, 3)
(2, 1)	(2, 2)	(2, 3)
(3, 1)	(3, 2)	(3, 3)

Tabella 1.1: legenda tabella

1.6 Altra Sezione

1.6.1 Listati dei programmi

Primo Listato

```
In questo ambiente      posso scrivere      come voglio,
lasciare gli spazi che voglio e non % commentare quando voglio
e ci sarà scritto tutto.
```


Quando lo uso è meglio che disattivi il Wrap del WinEdt

Conclusioni

Queste sono le conclusioni.

In queste conclusioni voglio fare un riferimento alla bibliografia: questo è il mio riferimento [3, 4].

Appendice A

Prima Appendice

In questa Appendice non si è utilizzato il comando:
`\clearpage{\pagestyle{empty}\cleardoublepage}`, ed infatti l'ultima pagina 8 ha l'intestazione con il numero di pagina in alto.

Appendice B

Seconda Appendice

Bibliografia

- [1] Primo oggetto bibliografia.
- [2] Secondo oggetto bibliografia.
- [3] Terzo oggetto bibliografia.
- [4] Quarto oggetto bibliografia.

Ringraziamenti

Qui possiamo ringraziare il mondo intero!!!!!!!!!!
Ovviamente solo se uno vuole, non è obbligatorio.