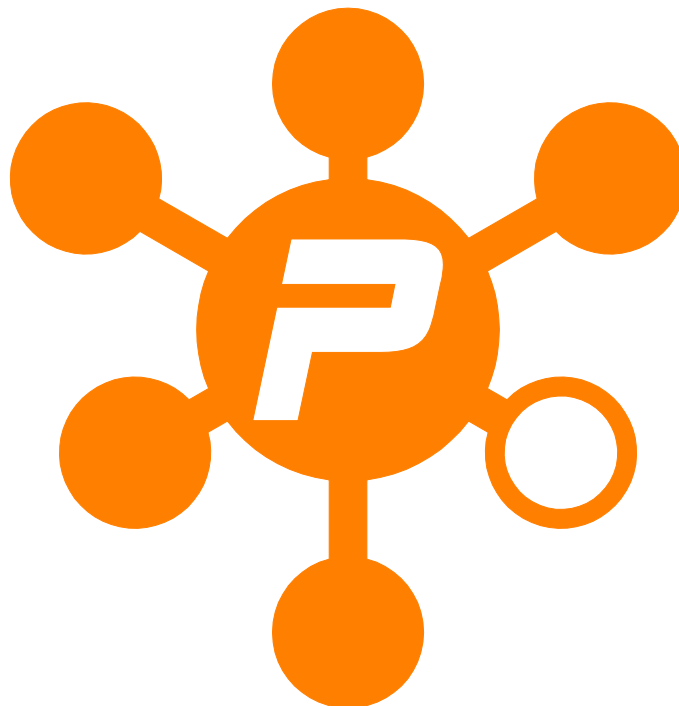


Progetto: Unisa Park	Versione: 1.4
Documento: SDD	Data: 13/12/2016

Università degli Studi di Salerno

Corso di Ingegneria del Software

Unisa Park
SDD
Versione 1.4



Data: 27/11/2016

Progetto: Unisa Park	Versione: 1.4
Documento: SDD	Data: 13/12/2016

Partecipanti:

Nome	Matricola
Maria Truvolo	0512103176
Federico Vastarini	0512103294

Scritto da:	Maria Truvolo e Federico Vastarini
-------------	------------------------------------

Revision History

Data	Versione	Descrizione	Autore
27/11/2016	1.0	Prima stesura del documento.	Maria Truvolo
03/12/2016	1.1	Aggiunta della sezione di suddivisione dei layers.	Maria Truvolo
08/12/2016	1.2	Aggiunta della sezione di permanenza dei dati.	Maria Truvolo
13/12/2016	1.4	Revisione della formattazione del testo ed indice.	Maria Truvolo

Progetto: Unisa Park	Versione: 1.4
Documento: SDD	Data: 13/12/2016

Indice

1. Introduzione	4
1.1 Scopo del sistema	4
1.2 Traguardi	4
1.3 Definizioni	5
2. Architettura software proposta	7
2.1 Panoramica del software	7
2.2 Scomposizione in sottosistemi	8
2.2.1 Rete di sensori	8
2.2.2 Server locale	9
2.2.3 Server remoto	10
2.2.4 Client	11
2.3 Software and hardware mapping	12
2.4 Persistenza dei dati	13
2.5 Protocolli e sicurezza	13
2.5.1 Protocolli	13
2.5.2 Sicurezza	14
2.7 Boundary conditions	14

Progetto: Unisa Park	Versione: 1.4
Documento: SDD	Data: 13/12/2016

1. Introduzione

Il design del presente sistema, offerto come soluzione al disagio dovuto alla presente configurazione delle aree di parcheggio dell'Università di Fisciano è stato pensato per offrire la massima flessibilità e scalabilità attraverso un'architettura distribuita dislocata tra l'ateneo stesso ed un server remoto.

1.1 Scopo del sistema

Il sistema si propone di fornire una soluzione all'avanguardia, suggerendo di fornire un quadro aggiornato in tempo reale riguardo le aree del parcheggio di Fisciano.

1.2 Traguardi

Il sistema si propone di raggiungere i seguenti traguardi:

Affidabilità: si garantisce che il sistema sia sempre online ed accessibile. *RAD 6.2*

Modificabilità: il sistema risulta completamente indipendente nella realizzazione dei layers e quindi facilmente modificabile in tutte le sue parti. *RAD 5, 6.4*

Mantenibilità: il sistema è naturalmente strutturato, suddiviso e documentato per essere facilmente mantenibile.

Comprensibilità: il codice prodotto e l'architettura sono completamente documentati e sviluppati secondo gli standard più recenti. *RAD 5, 6.1*

Adattabilità: il sistema può essere facilmente adattato ad altre località o configurazioni. *RAD 5, 6.4*

Riusabilità: il sistema si propone di essere il più possibile generico per offrire il riutilizzo della maggior parte delle sue componenti.

Progetto: Unisa Park	Versione: 1.4
Documento: SDD	Data: 13/12/2016

Efficienza: questo è il concetto chiave del progetto, ovvero la possibilità di avere informazioni aggiornate in tempo reale. *RAD 5, 6.3*

Portabilità: il sistema risulta in tutte le sue componenti facilmente ridistribuito su diverse piattaforme. *RAD 6.3*

Tracciabilità dei requisiti: i requisiti rilevati al momento della stesura del RAD saranno tutti esplicitati nella presente documentazione e finanche nelle successive.

Tolleranza ai malfunzionamenti: il sistema propone metodi semplici ed immediati per avvertire gli utenti di possibili guasti hardware od errori nel software. *RAD 6.2*

Costo: oltre ad utilizzare tutte le ottimizzazioni del caso e prevalentemente software open-source per ridurre i costi di implementazione il sistema si propone di essere un'occasione per l'università per giungere ad usufruire della copertura della rete LORA®. *RAD 6.3, 6.6*

Robustezza: un'attenta analisi del codice e l'utilizzo di controlli esterni alla piattaforma preverranno la possibilità dell'insorgere di errori e malfunzionamenti. *RAD 6.2*

Performance: codice e protocolli sono altamente ottimizzati per la velocità di esecuzione, mantenendo un occhio di riguardo verso l'utilizzo di risorse sia interne che esterne. *RAD 6.3*

Scalabilità: attraverso la piena e semplice adattabilità del sistema alle variazioni di configurazione delle aree. *RAD 6.3*

1.3 Definizioni

- **Slot**

La singola zona, definita solitamente dai vertici di un rettangolo, atta a consentire il

Progetto: Unisa Park	Versione: 1.4
Documento: SDD	Data: 13/12/2016

parcheggio di un unico veicolo.

- **Sensore**

Il sensore magnetico di tipo LORA®, montato all'interno di uno slot, capace di rilevare la presenza su di esso di un veicolo.

- **Area**

L'insieme di più slot di parcheggio solitamente adiacenti, raggruppati in una zona ben definita.

- **Array di Sensori (AdS)**

L'insieme di più sensori relativi ad un'unica zona.

- **Stazione**

La stazione ricevente del segnale dei sensori, costituita dall'antenna LORA® e da tutto l'hardware necessario al suo funzionamento.

- **Server Locale**

Il computer dell'Università di Fisciano adibito alla raccolta dei dati dei sensori e la trasmissione verso il server remoto.

- **Server Remoto**

Il computer locato in un qualsiasi punto adibito alla ricezione dei dati inviati dal Server locale, allo storage ed all'elaborazione degli stessi ed alla trasmissione verso gli utenti finali.

- **Client**

Il software scaricato dall'utente finale che permette la visualizzazione interattiva dei dati raccolti. Solitamente utilizzato tramite browser di un computer o mobile device.

- **Utente**

Qualsiasi persona che si trovi coinvolta nella ricerca di un parcheggio all'Università di Fisciano.

Progetto: Unisa Park	Versione: 1.4
Documento: SDD	Data: 13/12/2016

- **Veicolo**

Il mezzo di trasporto con il quale l'utente raggiunge l'università. Solitamente atto ad occupare un unico Slot.

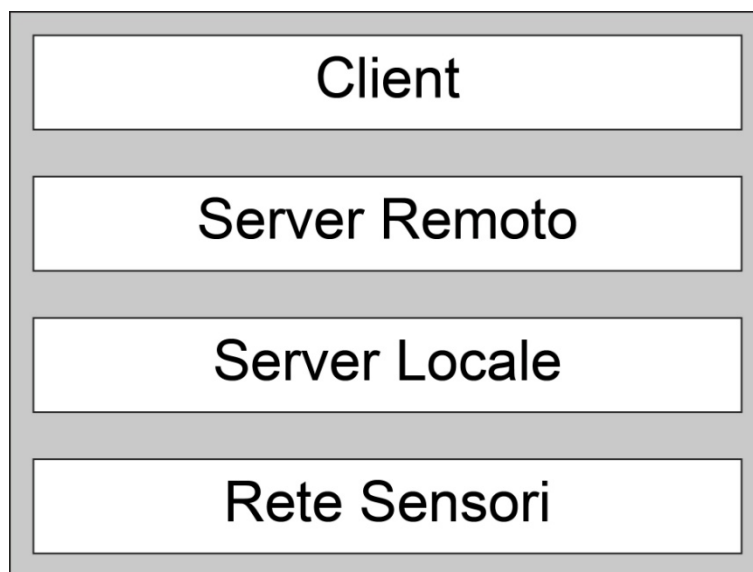
- **Amministrazione**

Qualsiasi persona atta alla gestione o manutenzione del sistema.

2. Architettura software proposta

2.1 Panoramica del software

Il software si compone di 4 layers, ognuno dei quali totalmente indipendente e sviluppato con diversa tecnologia.



Alla base si trova la rete di sensori, composta dai sensori e dall'antenna ricevente. Codice e protocolli interni sono definiti dallo standard LORA®. Questa invia i dati direttamente al secondo layer, il server locale, dislocato nell'ateneo. Il server, costruito su architettura Linux/JAVA si occupa della ricezione e della gestione del buffer di dati raccolti e da inviare al layer successivo.

Quest'ultimo, il server remoto, su architettura Linux/Apache/PHP e MySQL si occupa dello storage dei dati, della configurazione delle aree e della trasmissione dei

Progetto: Unisa Park	Versione: 1.4
Documento: SDD	Data: 13/12/2016

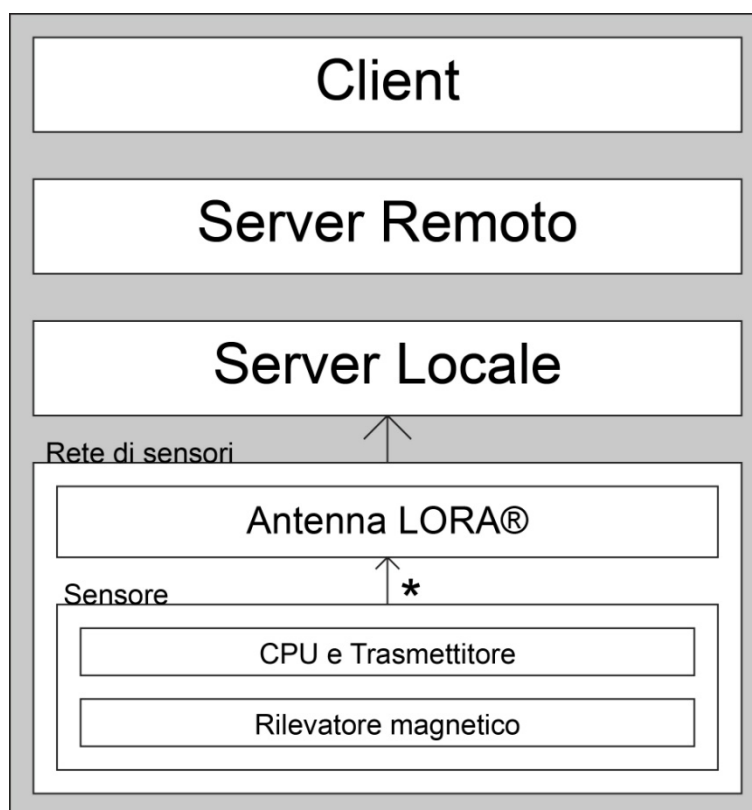
dati e del buffer al client.

Nel quarto layer, ovvero quello del client, avviene la presentazione che la traduzione dei dati raccolti, mediante scripting in HTML CSS e Javascript.

2.2 Scomposizione dei sottosistemi

2.2.1 Rete Sensori

La rete di sensori comprende l'antenna LORA® ed i sensori magnetici LORA® associati agli slot delle aree di parcheggio. Ogni sensore è dotato di una parte hardware di rilevazione del campo magnetico circostante e di una semplice CPU che si occupa semplicemente della rilevazione del segnale e della trasmissione dello stesso all'antenna.



La trasmissione del segnale all'antenna avviene tramite onde radio. Il protocollo di trasmissione è stabilito dal prodotto specifico utilizzato. Si faccia riferimento alle references di questo stesso documento.

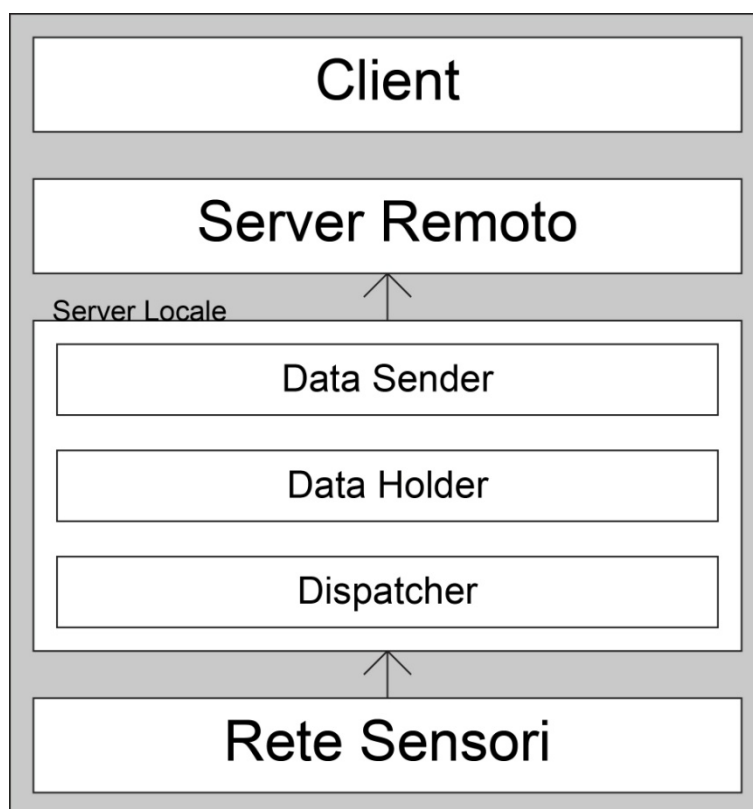
Progetto: Unisa Park	Versione: 1.4
Documento: SDD	Data: 13/12/2016

L'antenna a sua volta traduce il segnale radio al server locale come connessione standard TCP/IP.

Ad ogni sensore vengono associati al momento di deployment un indirizzo IP ed un ID di tipo intero incrementale.

2.2.2 Server Locale

Il server locale consta di una macchina Linux sulla quale sia stata installata un'istanza di JAVA Virtual Machine v.8. Il server ha semplicemente il compito di raccogliere i dati dal layer sottostante, organizzarli e tradurli al layer successivo.



Il software è composta da 3 parti fondamentali:

Un Dispatcher, che si occupa della ricezione dei dati dal layer dei sensori. Esso verifica l'attendibilità della connessione e trasmette il dato (on oppure off del sensore) ricevuto al Data Holder.

Il Data Holder si occupa di mappare il dato ricevuto sul buffer di dati. La mappatura avviene semplicemente secondo un'array associativa IP - ID definita nella

Progetto: Unisa Park	Versione: 1.4
Documento: SDD	Data: 13/12/2016

configurazione del sistema.

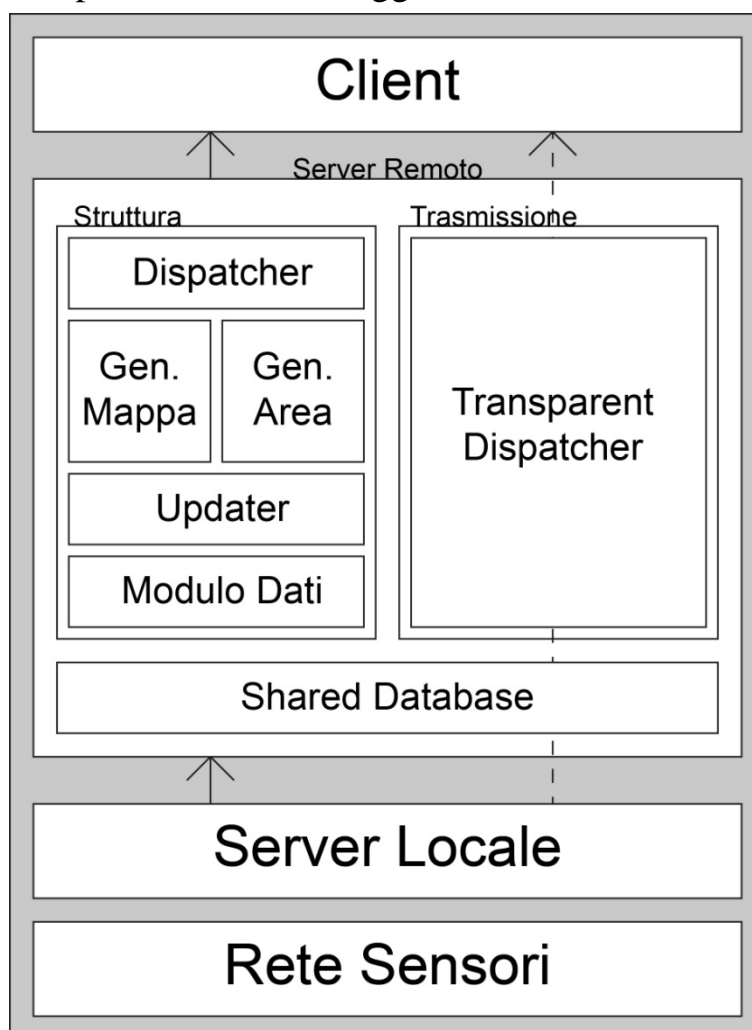
I dati utilizzati dal Data Holder non hanno nessun tipo di persistenza alla terminazione del programma).

Infine il Data Sender si occupa della trasmissione del buffer costruito dal Data Holder al layer successivo. Il protocollo utilizzato è definito successivamente in questo stesso documento.

La traduzione di queste 3 entità basta al semplice funzionamento del server locale.

2.2.3 Server Remoto

Il server remoto è suddiviso in 2 partizioni. La prima "Struttura" offre lo storage dei dati relativi alla configurazione attuale delle aree del parcheggio dell'Università. L'altra "Trasmissione" è invece dedicata alla trasmissione del buffer dal server locale al client, rendendo trasparente all'utente l'aggiornamento dei dati.



Progetto: Unisa Park	Versione: 1.4
Documento: SDD	Data: 13/12/2016

Entrambe le parti sono realizzate in PHP v5.6.x su piattaforma Apache v2.4.x e si appoggiano ad un database condiviso di tipo MySql v5.6.x.

Nessun Tool di sviluppo è utilizzato nella realizzazione del codice, non essendo nessuno dei tool comuni all'altezza di supportare i requisiti richiesti dal sistema.

Per quanto riguarda le componenti, la Struttura ha un proprio indirizzo IP e dominio DNS visibile all'utente. Nel nostro caso è stato utilizzato il dominio www.unisapark.com, già registrato e pronto ad essere fruibile con il deployment del sistema. Questo sarà il primo punto di contatto tra il client ed il server remoto.

Le componenti interne della struttura sono il Modulo Dati che gestisce la connessione e le strutturazione di query verso il servizio di database utilizzato.

Un Updater che gestisce la mappatura della configurazione delle aree di parcheggio e dei sensori sul database.

Un modulo di Generazione Mappa che gestisce l'estrazione dei dati relativi all'intera mappa dell'ateneo possibilmente richiesta dal cliente.

Un modulo di Generazione Area che gestisce l'estrazione dei dati relativi ad una singola area e la generazione statistica delle proiezioni dei posti liberi.

Un Dispatcher che permette la gestione delle chiamate dal client e la protezione da ddos ed altri tipi di attacchi.

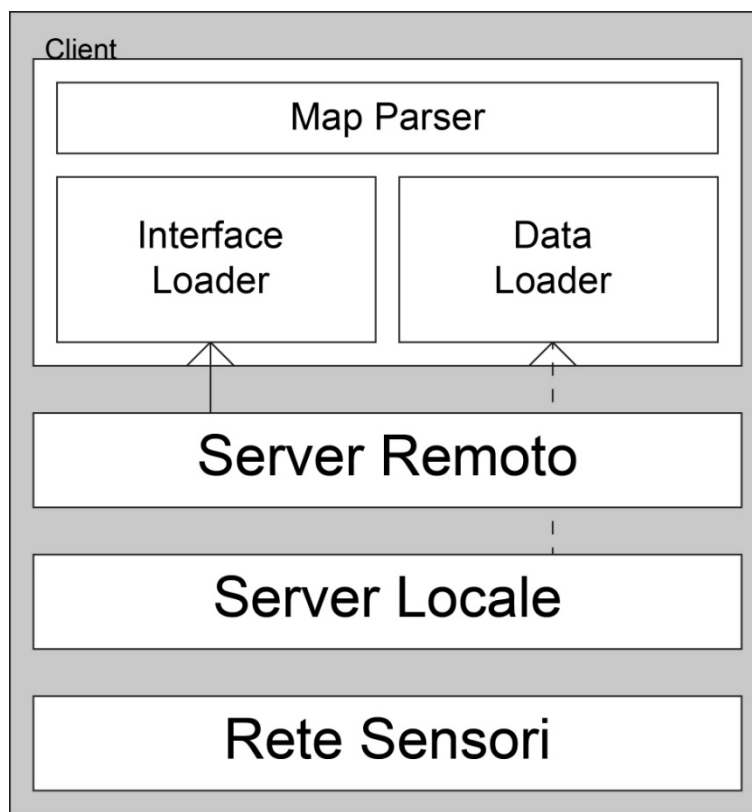
La parte di Trasmissione è invece composta da un unico elemento: un "Transparent Dispatcher". Data la natura estremamente performante del sistema ed il carico di lavoro che deve supportare si è optato per una totale cessione del lavoro al client. Inoltre in questo modo il buffer del server locale viene virtualmente reso direttamente al client rendendo il layer remoto completamente trasparente. E' infatti non esplicitato all'utente l'indirizzo IP da dove il client riceverà gli aggiornamenti. La composizione atomica di quest'elemento rispetto al resto del sistema garantisce la miglior performance possibile bilanciando il carico della sua controparte di Struttura. Quest'ultima infatti riceve pochi bursts di elevato carico rispetto alla Trasmissione che ne riceve un numero elevato per minuto ma di carico bassissimo.

2.2.4 Client

Il Client, ultimo layer del sistema si svolge interamente sul browser del computer o dispositivo mobile utilizzato dall'utente.

Progetto: Unisa Park	Versione: 1.4
Documento: SDD	Data: 13/12/2016

Esso è sviluppato direttamente in HTML5, CSS e Javascript senza l'utilizzo di bloatware che rallenterebbe l'immediatezza del caricamento e parsing dei dati.



Il primo modulo ad essere caricato, direttamente in contatto con il server remoto è l'Interface Loader, il quale sostituisce la comune home di una pagina web. Il suo compito è infatti quello di predisporre gli script di configurazione e caricare le risorse di base del layer.

Il Data Loader invece è il punto di arrivo dei dati aggiornati del sistema trasferiti in modalità trasparente dal server locale.

Infine il Map Parser utilizzando la configurazione degli script e l'aggiornamento dei dati dei precedenti due layers offre all'utente la visione globale e dettagliata della mappa in modalità seamless.

2.3 Software and Hardware Mapping

Come già descritto in precedenza il sistema sarà distribuito nel seguente modo:

Progetto: Unisa Park	Versione: 1.4
Documento: SDD	Data: 13/12/2016

Rete di sensori basata su un sistema LORA®.

Server locale costituito da una macchina Linux fornita dall'ateneo (possibilmente distribuzione Debian) con Java Virtual Machine v8 ed applicazione sviluppata interamente dal nostro team.

Server remoto in shared hosting basato su PHP v5.6.x su piattaforma Apache v2.4.x e database condiviso da due istanze di tipo MySql v5.6.x. Tutto il codice sarà sviluppato dal nostro team.

Client crossbrowser sviluppato in HTML5, CSS e Javascript che sia ottimizzato sia per computers che per mobile devices.

2.4 Persistenza dei dati

Per loro stessa natura i dati al livello di Rete dei sensori e Server Locale sono considerati volatili. La vera persistenza si ottiene al livello del Server Remoto che non solo tiene traccia della configurazione delle aree ma anche dello stato attuale e passato dei sensori. I dati vengono registrati su una serie di tabelle MySQL v5.6.x, di cui le principali sono:

- 1) Tabella della configurazione dei singoli posti: mantiene l'associazione IP, ID posizione e rotazione della shape dello slot, ID dell'area di riferimento e stato della disponibilità del sensore.
- 2) Tabella delle aree: tiene traccia della suddivisione degli slots in aree, dei nomi delle aree e del calcolo dei tempi di liberazione di uno slot.
- 3) Tabella di buffer: serve da perno e da storico al passaggio dei dati dal server locale al client.

2.5 Protocolli e sicurezza

2.5.1 Protocolli

Il sistema prevede una serie di protocolli per lo scambio di dati tra un layer e l'altro.

Progetto: Unisa Park	Versione: 1.4
Documento: SDD	Data: 13/12/2016

Tra la rete di sensori ed il server locale è utilizzato il protocollo standard della rete LORA® descritto nelle references.

Tra il server locale e quello remoto invece i dati vengono trasmessi attraverso un protocollo basato sulle seguenti caratteristiche:

I dati relativi allo stato dei sensori vengono ordinati ed scritti in una sequenza di bit.

All'inizio della sequenza viene aggiunto un byte indicante la lunghezza della sequenza stessa.

Il frame così composto viene inviato tramite richiesta di header al server remoto.

Tra il server remoto ed il client si utilizza lo stesso sistema. Il map parser del client fornirà la decifrazione della stringa al momento della visualizzazione per l'utente.

2.5.2 Sicurezza

La sicurezza del sistema è garantita dalla chiusura del sistema stesso a qualsiasi connessione non autenticata e possibilmente dalla rilevazione di tentativi di ddos.

2.6 Controllo globale del software

Una volta eseguito il deployment del sistema il software necessita di un controllo minimo se non nullo, essendo nella quasi totalità delle sue parti completamente automatizzato.

Se non interviene infatti un evento esterno quale la riconfigurazione di un'area od il malfunzionamento di uno slot, l'intero sistema sarà capace di continuare la sua esecuzione senza che necessiti di alcun controllo umano.

2.7 Boundary Conditions

Data la natura totalmente distribuita del sistema, le condizioni al limite quali lo

Progetto: Unisa Park	Versione: 1.4
Documento: SDD	Data: 13/12/2016

spegnimento di una macchina, di un sensore o di un client influiranno minimamente sulla ripresa del funzionamento del sistema stesso.

Esso sarà infatti capace di tornare pienamente in funzione non appena tutti i suoi stadi siano rimessi a regime.

Sarà inoltre in grado di notificare all'utente in tempo reale la mancanza di dati qualora essi non siano aggiornati.

Le procedure di sicurezza e la capacità di trasmissione dei dati assicurano già ampiamente l'affidabilità, la performance e la robustezza del sistema.