



Titolo: transCeiver infrAstrutturale e Multitecnologico per il Monitoraggio di pErsone e Oggetti in ambienti non coperti dal GPS

Acronimo: **CAMMEO**

Responsabile di Progetto T-Connect: Dr. Ing. Paolo Sperandio

Responsabile di Progetto Progesi: Ing. Massimo Baldasseroni

CAMMEO Architectural Design

Inizio Progetto: 5 Luglio 2010

Durata: 24 Mesi

Numero di Protocollo: 220

Versione: 2.2

Progetto finanziato da FILAS Finanziaria LAziale di Sviluppo
all'interno del programma POR FESR Lazio 2007/2013 Attività I.1



Informazioni sul documento:

Titolo:	CAMMEO Architectural Design
Società:	Progesi SpA, T-Connect S.r.l.
Data di emissione:	20/05/2011
Tipologia:	AD
Versione:	2.0
Livello di diffusione:	Pubblico
Autori:	Marco Torrisi, Fabiola Sillavi, Claudio Buttinelli

Tabella delle modifiche

Versione	Data	Modifica	Apportata da:
1.0	04/10/2010	Architectural Design	Marco Torrisi
1.1	20/10/2010	Aggiornamento contenuti documento	Marco Torrisi
1.2	19/11/2010	Inserimento DataFlow, Aggiornamento Struttura SW	Marco Torrisi
1.3	02/12/2010	Aggiornamento Modulo Applicativo Inserimento e aggiornamento capitolo 5.	Marco Torrisi
1.4	15/12/2010	Integrazione CAMMEO Server Integrazione CAMMEO Service	Claudio Buttinelli
1.5	20/12/2010	Revisione	Claudio Buttinelli
1.6	17/01/2011	Inserimento data flow aggiornati struttura Cammeo Transceiver. Aggiornamento introduzione.	Marco Torrisi
1.7	01/02/2011	Aggiornamento documento, Conclusioni.	Marco Torrisi Fabiola Sillavi
1.8	14/02/2011	Aggiornamento e modifica sistema Cammeo Transceiver e dell'architettura di sistema dello stesso	Marco Torrisi Fabiola Sillavi



1.9	23/02/2011	Revisione e modifica sull'intero documento	Fabiola Sillavi
2.0	07/03/2011	Revisione Generale: aggiornamento blocchi funzionali Cammeo Transceiver	Fabiola Sillavi Marco Torrisi
2.1	14/03/2011	Revisione uniformità documento	Claudio Buttinelli
2.2	28/03/2011	Revisione finale (Cammeo Transceiver) e controllo formattazione.	Marco Torrisi Fabiola Sillavi



Indice:

1	Executive summary.....	7
1.1	Campi di applicazione.....	7
2	Introduzione	9
2.1	Convenzioni utilizzate nel documento	9
3	Architettura di sistema.....	10
3.1	Architettura di primo livello	10
3.2	Relazioni tra i componenti di primo livello	13
3.3	CAMMEO Server, architettura di secondo livello.....	14
3.3.1	Localization Manager.....	14
3.3.2	Map Manager	15
3.3.3	Persistence Manager	17
3.3.4	Request Manager	17
3.3.5	Client Interface	18
3.3.6	Relazioni tra i componenti di secondo livello.....	18
3.4	CAMMEO Service.....	22
3.4.1	Localization Device Manager	22
3.4.2	Server interface	23
3.4.3	User Interface e Admin Interface	23
3.4.4	Lato Utente	24
3.4.5	Lato Amministratore.....	24
3.4.6	Relazioni tra i componenti di secondo livello.....	25
3.5	CAMMEO Transceiver	30
3.5.1	Wireless Sensor Network / GNSS (WSNG)	31
3.5.2	Localization Device Manager	32
3.5.3	Server interface	35
3.5.4	Zigbee/Bluetooth/etc Engine.....	36
3.5.5	Relazioni tra i componenti di secondo livello.....	37
4	Architettura di deployment	39



5	Conclusioni	41
6	Bibliografia.....	42



Lista Figure:

Figura 1: Architettura CAMMEO ad alto livello	11
Figura 2: Server CAMMEO - Architettura di secondo livello	14
Figura 3: CAMMEO Service - Architettura di secondo livello	22
Figura 4: diagramma di sequenza per rilevare i dati grezzi di posizione	26
Figura 5: diagramma di sequenza per aggiornare i dati di posizione del client	27
Figura 6: Richiesta posizione e mappa indoor.....	28
Figura 7: Richiesta posizione e mappa outdoor.....	29
Figura 8: Architettura di secondo livello del CAMMEO TRANSCEIVER	30
Figura 9: Wireless Sensor Network / GNSS	32
Figura 10: Diagramma di flusso del Priority Algorithm	35
Figura 11: Frame Incapsulation structure	36
Figura 12: Diagramma di flusso dell'Engine Aggregator.....	37
Figura 13: Schema del deploy della piattaforma CAMMEO.....	40



Lista Tabelle:

Tabella 1: Interfaccia client server.....	21
Tabella 2 : Tabella di priorità tecnologie CAMMEO Transceiver.....	35
Tabella 3: Contenuto del data frame	36



1 Executive summary

Questo Deliverable, utilizzando come input le informazioni prodotte nel D1.1, descrive i blocchi funzionali dei singoli moduli formanti il sistema CAMMEO. Come già indicato nella proposta di progetto CAMMEO è formato da tre macro blocchi principali:

- Server
- Service
- Transceiver

Questo Deliverable dettaglia sia i singoli macroblocchi che i loro sottosistemi, andando ad esplicitare le interfacce ed i dati necessari al corretto funzionamento del sistema.

1.1 Campi di applicazione

Da una prima indagine di mercato, seguendo una procedura top-down, cioè partendo dai servizi fruibili dalla futura piattaforma di localizzazione, sono stati evidenziati i seguenti mercati di interesse:

- Autonomous Guided Vehicles (AGV)

- Il corretto posizionamento di strumentazioni, come elevatori o gru, il cui erraneo utilizzo potrebbe portare ad incidenti sul lavoro

- Interactive guiding application (musei, orti botanici, fiere)

- Applicazioni in tempo reale potrebbero guidare i visitatori all'interno di musei, siti archeologici, centri commerciali e parchi di divertimento. Una volta entrato nel museo il dispositivo d'utente entra in comunicazione con la piattaforma che, una volta "equipaggiato opportunamente" l'utente, gli consentirà di godere in modo del tutto indipendente la sua visita

- Supermarket

- Carrelli intelligenti (equipaggiati con PDA) una volta scaricata la lista della spesa del cliente (per mezzo di un collegamento Bluetooth con il suo cellulare), questo viene guidato all'interno del supermercato alla ricerca dei suoi acquisti o di eventuali offerte del giorno

- Healthcare

- Supporto al monitoraggio dei reparti di maternità, potrebbe evitare eventi di rapimento, smarrimento o scambio dei neonati. L'uso congiunto di spille, (associate univocamente



alla coppia madre-figlio), e di sensori cardiaci, scatenerebbero un allarme nel caso di rimozione della spilla o di erronea associazione

- Interactive location-based games

- La visita ad un museo può essere trasformata in un avvincente gioco di ruolo dove il visitatore, invece di assistere passivamente alla descrizione di quadri o statue, può essere coinvolto nella risoluzione di avvincenti casi sfruttando la conoscenza acquisita durante la visita. Equipaggiato con un PDA all'ingresso del museo il visitatore entra in un mondo in cui, in base alla sua posizione vede apparire sul suo PDA domande, immagini, filmati, puzzle da comporre per poter passare al livello successivo. Dovrà interagire con altre persone nel museo, chiedere a loro informazioni, magari su opere che ancora non ha visto e così giungere alla fine della visita, che sicuramente avrà lasciato ricordi più vivi di quelli avuti per mezzo di semplici descrizioni

- Security

- In ambienti come cliniche psichiatriche o carceri, attacchi di violenza sono un evento non poco frequente. Il monitoraggio sia degli impiegati che degli ospiti della struttura, faciliterebbe la gestione di questi attacchi, informando immediatamente il personale più vicino con tutti i dati inerenti la posizione dell'evento e lo stato di chi lo ha causato

- Ambient intelligence

- All'interno di un ambiente intelligente (Smart Home) il motore di localizzazione potrebbe essere utilizzato per monitorare il comportamento di persone anziane. Il fermarsi troppo a lungo in ambienti isolati come il bagno (a causa di una caduta può aver perso conoscenza), o la cucina (una fuga di gas potrebbe aver causato uno svenimento del soggetto) potrebbe azionare un allarme e salvare la vita della persona

- Asset tracking

- Posizionamento attrezzature: la costante consapevolezza di conoscere lo stato delle cose non facilita solo il loro recupero, ma evita anche di perderle. Inoltre queste informazioni possono essere anche utilizzate a livello statistico per monitorare il loro livello di uso o il tempo di inutilizzo, che all'interno di un ospedale ricopre un ruolo indispensabile



2 Introduzione

Il deliverable presenta l'architettura di sistema partendo da una sua schematizzazione generale, rappresentazione dei tre blocchi generali, fino a presentare anche le interazioni tra ogni singolo elemento costituente ogni macroblocco.

Il deliverable si conclude con un sezione dedicata al deploy della piattaforma.

L'architettura indicata in questo deliverable è differente da quella indicata nella proposta a seguito del successo delle fasi di raccolta dei requisiti e di analisi degli stessi, che hanno permesso di ridefinire e rivalutare la detection e l'elaborazione delle informazioni raccolte da CAMMEO.

2.1 Convenzioni utilizzate nel documento

Per rappresentare l'architettura è stato utilizzato il linguaggio UML [1]. Questo linguaggio ci ha permesso di specificare l'architettura sotto diversi punti di vista che ci hanno permesso di definire il sistema indipendentemente dal tipo di tecnologia utilizzata.

Di seguito, riportiamo alcuni concetti usati comunemente sul documento:

- *Componente*: è una sezione modulare del sistema che incapsula un contenuto con uno scopo comune nell'ambiente in cui viene utilizzata. Definisce un insieme di comportamenti e funzionalità che sono forniti tramite le interfacce ad esso associate. Solitamente può essere espresso come una generalizzazione di un insieme di moduli e funzionalità che lo compongono.
- *Modulo*: è una singola unità o funzionalità all'interno di un componente. E' una specializzazione di un componente.
- *Data Frame*: una trama di dati costituiti da un intestazione, header e da un payload.
- *Toolchain*: l'insieme dei programmi (tools) usati nello sviluppo di un prodotto software, nel nostro caso, i tools presenti comprendono l'ambiente di test basato su Linux embedded.

Nel presente documento vengono utilizzati i seguenti acronimi:

AD	Architectural Design
ADD	Architectural Design Document
SR	Software Requirement
SRD	Software Requirement Document
SV	Satellite Vehicle
PRN	Pseudorange Number



3 Architettura di sistema

Questa sezione presenta l'architettura di CAMMEO ad alto livello, definendone le principali componenti e le relazioni tra di esse.

Per ogni elemento che compone quest'architettura sono stati dettagliati il loro contenuto, le relazioni interne tra gli elementi che lo compongono e le relazioni con gli elementi esterni, sempre ed esclusivamente quelli ad alto livello.

L'architettura di CAMMEO prende come riferimento gli use cases e i requisiti definiti nel [22] e nel [23].

3.1 Architettura di primo livello

E' stato necessario suddividere il sistema nei seguenti macro componenti di alto livello che ne raggruppano la maggior parte delle funzionalità:

- *Localization Device Manager*
- *Server interface*
- *Client interface*
- *Request Manager*
- *Localization Manager*
- *Map Manager*
- *Persistence Manager*
- *User Interface*
- *Admin Interface*

La Figura 1 fornisce la rappresentazione grafica dell'architettura ad alto livello della piattaforma CAMMEO. A sinistra della figura sono stati schematizzati il CAMMEO Service ed il CAMMEO Transceiver mentre sulla destra è rappresentata schematizzata la struttura del CAMMEO Server.

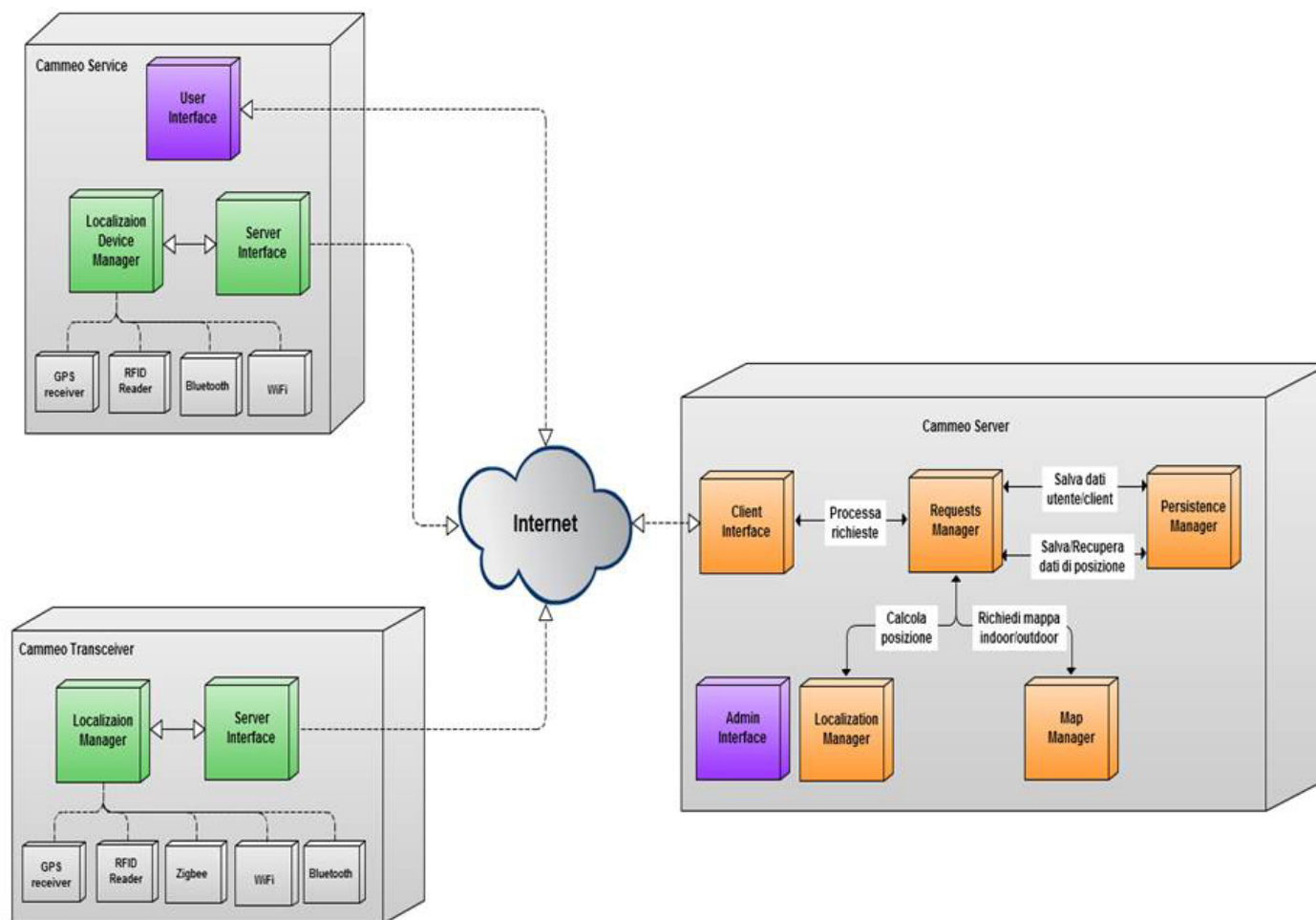


Figura 1: Architettura CAMMEO ad alto livello



All'interno del CAMMEO Service e del CAMMEO Transceiver troviamo i seguenti componenti condivisi:

- Server Interface: costituisce l'interfaccia di collegamento con il server, fornisce gli strumenti per la codifica/decodifica dei dati passati da e verso il server; crea e distrugge le connessioni; consente l'autenticazione dell'utente e si occupa del trasporto dei dati. Come specificato successivamente è costituito da un client con architettura REST.
- Localization Device Manager: è il componente principale di ciascun client, recupera le informazioni dai dispositivi di localizzazione (WiFi, RFID, GPS, Bluetooth, Zigbee, etc), le elabora e le invia al componente Server Interface per spedirle successivamente al CAMMEO Server.

All'interno del CAMMEO Server troviamo i seguenti componenti:

- Client Interface: rappresenta l'interfaccia alla quale ciascun client può connettersi per richiedere e inviare dati (posizione, mappe, autenticazione). È un server REST con modello RPC (Remote Procedure Call) che consente al client di eseguire chiamate di funzioni remote e, dove necessario, richiedere informazioni.
- Request Manager: si prende cura di reindirizzare ciascuna richiesta ai vari componenti interessati, verificando l'attendibilità del richiedente e le credenziali di utilizzo della risorsa/componente richiesto
- Persistence Manager: fornisce un sistema di persistenza dei dati, sia che essi riguardino l'utente che la posizione dei client (CAMMEO Service e CAMMEO Transceiver). Permette di memorizzare permanentemente le configurazioni del server e dei client.
- Localization Manager: rappresenta l'insieme motori delle tecnologie di localizzazione integrate in CAMMEO. Ciascuna tecnologia permette di calcolare la posizione di un client; un sistema di aggregazione permette di calcolare la posizione localizzata del client sulla base di priorità, accuratezza e disponibilità delle tecnologie.

Le interfacce utente (User Interface e Admin Interface) forniscono i seguenti meccanismi di visualizzazione:

- Visualizzazione della posizione dell'utente CAMMEO Service o CAMMEO Transceiver relativa ad una data mappa,
- Visualizzazione dei punti di interesse che l'utente ha ricercato,
- Sistema di back end per gestire i dati di amministrazione del server come la creazione di utenti, mappe, inserimento e modifica di nuovi dispositivi di localizzazione etc.



Come specificato nei successivi paragrafi l'interfaccia utente si dovrà basare su un'interfaccia web accessibile dalla maggior parte dei dispositivi portatili (smartphone) e non.

3.2 Relazioni tra i componenti di primo livello

Al fine di comprendere correttamente il funzionamento del sistema è necessario specificare la natura dei dati scambiati tra i vari componenti di alto livello e le interfacce che il sistema fornisce.

E' possibile specificare l'intero processo di comunicazione partendo da uno dei due client del sistema, il Service o il Transceiver, verso il server di localizzazione.

All'interno del client, nel componente Localization Device Manager, si effettuano i calcoli necessari a determinare i dati di posizione del dispositivo (sia che siano dati grezzi come numero di AP WiFi, i loro indirizzi MAC e il livello RSSI o la posizione stimata con algoritmi di multilaterazione). Questi dati vengono inviati all'interfaccia Server Interface, che li converte in formato JSON, per poi inviarli al server per la successiva elaborazione. L'interfaccia di comunicazione tra client e server, è basata sulla tecnologia dei web services con architettura REST. Il formato di codifica dei dati utilizzato per questi web service è JSON, Javascript Object Notation [2].

Il CAMMEO Server espone le interfacce di comunicazione tramite il componente Client Interface. Questo componente è costituito da un server REST che riceve le richieste, di posizione e di invio dati di localizzazione e le inoltra a sua volta al componente Request Manager. Questo componente smisterà infine le richieste ai componenti interessati come spiegato successivamente:

- Per le richieste di elaborazione dei dati grezzi di posizione, il Request Manager invia i dati di localizzazione raccolti dal client al componente Localization Manager; in questa comunicazione l'integrità dei dati deve essere verificata a priori dal Request Manager.
- Per richiedere la posizione al CAMMEO Server, operazione esclusiva del CAMMEO Service, il Request Manager comunicherà invece con il componente Persistence Manager che mantiene, in uno stato consistente, i dati di posizione di ciascun Service e Transceiver CAMMEO.
- Le richieste relative a configurazioni, utenti e client sono inoltrate anch'esse al Persistence Manager.
- Ogni richiesta relativa alla visualizzazione di mappe indoor e/o outdoor viene inoltrata al Map Manager.



3.3 CAMMEO Server, architettura di secondo livello

Il server fornisce il supporto ai client CAMMEO nel calcolo della loro posizione indoor e/o outdoor e fornisce tutti i servizi location-based che essi richiedono.

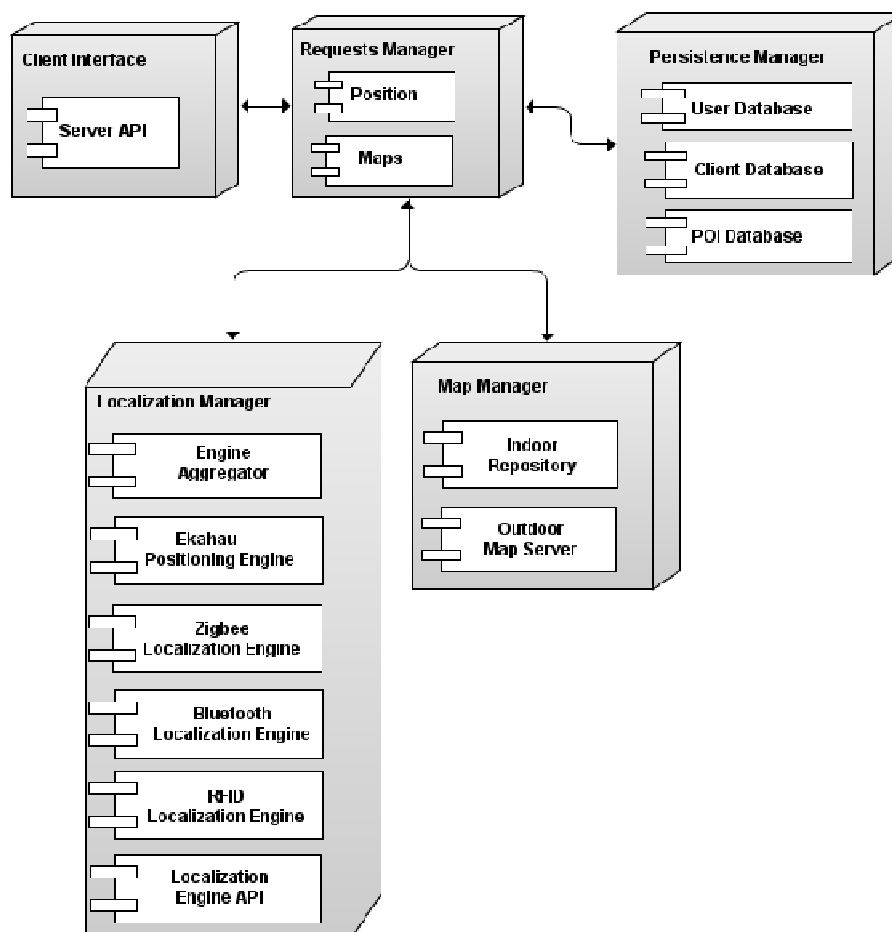


Figura 2: Server CAMMEO - Architettura di secondo livello

3.3.1 Localization Manager

Il principale componente, chiamato Localization Manager, costituisce il fulcro per il calcolo dei dati provenienti dal CAMMEO Transceiver e dal CAMMEO Service. Questo componente sfrutta una serie di tecnologie di localizzazione che utilizzano differenti tecniche per il calcolo della posizione: RSSI; reti WLAN limitrofe; RFID rilevati; ecc.. Per ottenere la cooperazione tra queste tecnologie è necessario integrare, all'interno di questo componente, uno specifico client per ognuna di esse:



- WiFi: è stato adottato Ekahau RTLS [3] come tecnologia di localizzazione indoor.
- RFID: è stato sviluppato un algoritmo di prossimità che sfrutta i dati inseriti in un DB relativi alle posizioni dei tag RFID all'interno delle aree coperte da CAMMEO. Nel DB ad ogni tag RFID è associata una mappa e una posizione relativa a quella mappa. A seconda dei tag RFID rilevati dal CAMMEO Service o Transceiver, viene calcolata la posizione dell'utente.
- GPS: I dati di posizione sono disponibili decodificando le stringhe NMEA inviate dal ricevitore GPS [19]. Come da requisiti funzionali è stato ritenuto necessario mantenere traccia della posizione outdoor rilevata tramite GPS, questo dato di posizione viene inviato dai client tramite la stessa interfaccia utilizzata per i dati WiFi e RFID; il dato non viene trattato ma viene direttamente memorizzato dal database relativamente alla posizione del client associato.
- ZigBee: I dati di posizione stimati tramite la rete ZigBee saranno calcolati tramite un algoritmo di multilaterazione[18], sfruttando un database contenente le posizioni dei singoli nodi Zigbee e le relative alle mappe.
- Bluetooth: attualmente in fase di studio e test, questa tecnologia sarà utilizzata come tecnologia di prossimità per il calcolo della posizione se dai test risulterà una tecnologia di scarsa accuratezza per un sistema di multilaterazione. Se utilizzata come tecnologia di prossimità, sfrutterà un database di dispositivi georeferenziati per determinare la posizione dell'utente (come avviene in parte per gli RFID e per gli Zigbee)
- API per integrazioni future: ogni tecnologia di localizzazione si integra all'interno del Localization Manager utilizzando le API messe a disposizione dal Localization Manager. Queste API consentono di integrare ulteriori tecnologie di localizzazione all'interno del CAMMEO Server.

Il componente Engine Aggregator permette la collaborazione motori delle varie tecnologie di localizzazione. Una volta calcolate le posizioni per mezzo di ciascuna tecnologia le correla ottenendo una posizione ottimizzata. In questo calcolo è possibile tener conto anche dei livelli di accuratezza delle singole tecnologie, dai tempi impiegati per l'elaborazione e dall'ambiente in cui si trova il client.

3.3.2 Map Manager

Il secondo componente dell'architettura del CAMMEO Server è il Map Manager. Questo componente fornisce al CAMMEO Service le funzionalità per recuperare e visualizzare le mappe indoor e outdoor basandosi sulla sua posizione e sulle dimensioni della mappa da visualizzare. E' costituito principalmente da un server cartografico per le mappe outdoor e da una repository di immagini per la parte indoor.



Il server cartografico utilizzato è GeoServer [4], un server di dati geospaziali basato su standard del consorzio Open Geospatial Consortium [5]. Qualora il CAMMEO Service richiedesse la visualizzazione di un ambiente outdoor, la relativa mappa renderizzata verrà richiesta direttamente a questo server. Il server in risposta fornirà una serie di immagini che costituiranno la porzione di ambiente outdoor richiesto a seconda delle dimensioni disponibili per la visualizzazione. Le operazioni di zoom e pan della mappa inviate dal CAMMEO Service verranno reindirizzate a questo server che produrrà le corrispondenti immagini. I servizi forniti all'interfaccia utente per la visualizzazione della mappa sono:

- WMS: protocollo standard per fornire le mappe georeferenziate generate da GeoServer utilizzando un database GIS. Ad ogni richiesta, il server genera una nuova immagine basandosi sui dati memorizzati all'interno del database GIS. Solitamente il database GIS contiene dati vettoriali che vengono renderizzati in vari formati grafici (png, jpeg, tiff)
- GeoCache: si basa sul protocollo WMS ma è comprensivo di una cache che consente una risposta rapida elaborando il processo di rendering durante il deploy del server.

Entrambi questi servizi sono disponibili al CAMMEO Service per la richiesta delle mappe relativamente alla sua posizione.

Per le mappe indoor si è scelta una strada alternativa diversa, una repository di immagini statiche. Due principali motivi hanno portato a questa scelta; la complessità del processo di georeferenziazione di piccole aree, la loro scarsa precisione e la necessità di dover ri-proiettare coordinate cartesiane su coordinate geografiche.

La georeferenziazione è la tecnica che permette di associare ad un dato, in formato digitale, una coppia di coordinate che ne fissino la posizione sulla superficie terrestre. Per una mappa indoor, la georeferenziazione consiste nel prendere almeno due punti della stessa mappa e associarli a due coppie di coordinate geografiche. Inizialmente è stato considerato necessario dover georeferenziare le mappe indoor per poterle visualizzare sulla stessa interfaccia utente senza dover modificare il sistema di coordinate di riferimento. Dopo un'analisi dei framework a disposizione per l'interfaccia utente, è stato valutato che l'utilizzo di mappe statiche, non georeferenziate e con sistema di coordinate di riferimento cartesiane (x,y in pixel dal punto in alto a sinistra della mappa) fornisce una gestione più semplice ed efficace per la loro rappresentazione.

Una serie di script lato CAMMEO Service permettono di richiedere l'una o l'altra mappa a seconda dei dati sulla posizione forniti dal CAMMEO Server. Il CAMMEO Server fornisce infatti sia le coordinate sulla posizione che l'ambiente indoor e/o outdoor in cui il CAMMEO Service si trova. A quel punto sono gli script lato CAMMEO Service che richiedono i dati spaziali al GeoServer o alla repository indoor. Per chiarire questa comunicazione si rimanda al capitolo Architettura di deployment.



3.3.3 Persistence Manager

Come definito nel D1.1 i dati relativi ai client e alle loro posizioni devono essere memorizzati e conservati in un database anche dopo il riavvio del CAMMEO Server.

Sfruttando la tecnica ORM (Object Relational Mapping) [6] è possibile integrare il paradigma orientato agli oggetti utilizzato dal linguaggio di programmazione scelto (Java) con un sistema RDBMS [7]. Questo consente di memorizzare le variabili e le istanze di oggetti utilizzati all'interno del CAMMEO Server in un database relazionale e effettuare le operazioni CRUD (Create, Read, Update and Delete) per questi dati in maniera trasparente riducendo la quantità di codice necessario. Per questo scopo è necessario adottare le ultime specifiche JPA 2.0 che permettono la gestione della persistenza e l'object-relational mapping in Java EE.

L'application server Glassfish [8], definito nelle specifiche software del server CAMMEO, fornisce l'implementazione delle JPA tramite il framework EclipseLink [9]. L'utilizzo di questo framework per mappare oggetti all'interno di un database permetterà un rapido sviluppo del server anche nel caso in cui, durante lo sviluppo, sia necessario modificare la base di dati precedentemente definita, l'aggiunta di variabili o oggetti non implica una nuova scrittura del database né tanto meno delle classi associate. La maggior parte degli aggiornamenti è automatica e trasparente allo sviluppatore riducendo di molto i tempi di implementazione.

All'interno del database relazionale saranno mappati i dati utente, i client e le mappe come specificato nel deliverable dei requisiti [23].

Il database utilizzato è Apache Derby[10], un database integrato nell'application server Glassfish.

3.3.4 Request Manager

Questo componente è un proxy tra il server REST (Client Interface), per la comunicazione con i client, e il resto dell'applicazione CAMMEO Server.

Il suo principale compito è quello di identificare le autorizzazioni per l'accesso ai dati richiesti e inoltrare le chiamate ai rispettivi componenti.

Come già anticipato, per l'elaborazione dei dati di posizione le richieste vengono indirizzate al Localization Manager, per le informazioni sulla posizione le richieste vengono indirizzate al Persistence Manager.

Le interazioni tra il Localization Manager e il Persistence Manager avvengono tramite il Request Manager. Il tutto permette una separazione netta dei vari componenti e la gestione delle relazioni in un unico componente software.

Nella Figura 1 e nella Figura 2 è indicato che il Map Manager è collegato direttamente al Request Manager per la gestione delle richieste di mappe indoor e outdoor. Questo collegamento è puramente logico, il Map Manager pubblica direttamente i servizi necessari al



client senza intermediari. Utilizza il Request Manager per verificare l'autorizzazione all'invio dei dati e per prelevare i dati di configurazione delle mappe dal Persistence Manager.

3.3.5 Client Interface

L'ultimo componente all'interno dell'architettura server è il Client Interface. Questo componente è rappresentato da un server con architettura REST che pubblica una serie di web services per la comunicazione client-server.

L'implementazione è basata su Jersey, implementazione di riferimento per la realizzazione di web services RESTful su standard JAX-RS [11].

I web services pubblicati permettono ai client di interagire con il server utilizzando il protocollo HTTP per lo scambio di dati. Come già specificato nei precedenti paragrafi, lo scambio di informazioni avviene con dati in formato JSON, contrariamente ai web services SOAP che utilizzano XML per la rappresentazione dei dati.

3.3.6 Relazioni tra i componenti di secondo livello

Come definito precedentemente, i componenti Server Interface e Client Interface forniscono i metodi per le interfacce di comunicazione tra i client e il server.

La maggior parte dell'architettura del sistema CAMMEO (CAMMEO Server e CAMMEO Service) è basata sul modello REST, inventata da Roy Fielding per descrivere la natura del Web.

I web services, che adottano una architettura REST, utilizzano il protocollo HTTP 1.1, restringendone le operazioni alle sole permesse dal questo protocollo. L'interazione non avviene per mezzo di chiamate di procedure o invio di messaggi, ma scambiando rappresentazioni dello stato di risorse.

I vincoli di REST variano in base ai concetti che regolano il Web:

- Gli agenti utente interagiscono con le risorse e le risorse sono tutto ciò che può essere rappresentato. Ogni risorsa può essere risolta mediante un URI (Uniform Resource Identifier) univoco.
- L'interazione con le risorse (localizzate tramite il loro URI) avviene utilizzando l'interfaccia fornita dello standard HTTP (GET, POST, PUT e DELETE).
- Le risorse sono autodescrittive. Tutte le informazioni necessarie per elaborare una richiesta per una risorsa sono contenute all'interno della richiesta stessa, consentendo servizi stateless.

Seguendo questi principi sono stati definiti gli URI e le risorse che il componente Client Interface fornisce per la comunicazione con il server.

Di seguito presentiamo la tabella con gli URI, le risorse, i metodi e i mime type che fornisce CAMMEO. Queste risorse possono essere utilizzate e richiamate anche da altre applicazioni differenti dall'interfaccia utente di CAMMEO. Questa opportunità permette di espandere le



possibilità di integrazione e creazione di applicazioni basate su CAMMEO e sulle sue tecnologie di localizzazione.

Le risorse messe a disposizione da questi web services consentono un completo controllo sul funzionamento del CAMMEO Server, dalla gestione degli utenti all'aggiornamento dei dati di posizione di un utente, il tutto sfruttando delle API accessibili da qualsiasi piattaforma software.



Risorsa	URI	Metodi	MIME type	Descrizione
Invio dati di posizione grezzi	<i>/position/{id}</i>	PUT	application/json	Invio dei dati di posizione grezzi dal client con identificativo id al server per elaborare la posizione
Posizione utente	<i>/position/{id}</i>	GET	application/json	Posizione del client con identificativo id
Informazioni mappa	<i>/map/{id}</i>	GET	application/json	Richiesta dei dati relativi alla mappa con identificativo id
Informazioni poi	<i>/poi/{id}</i>	GET	application/json	Recupero dei dati relativi ad un punto di interesse identificato con id
Informazioni client	<i>/client/{id}</i>	GET	application/json	Recupero dei dati relativi ad un client identificato con id
Informazioni utente	<i>/user/{id}</i>	GET	application/json	Recupero dei dati relativi ad un utente identificato con id
Informazioni gruppo	<i>/group/{id}</i>	GET	application/json	Recupero dei dati relativi ad un gruppo di utenti identificato con id
CRUD punti di interesse	<i>/poi</i>	POST, PUT, DELETE	application/json	Creazione, modifica, eliminazione dei dati relativi ad un punto di interesse
CRUD mappe	<i>/map</i>	POST, PUT, DELETE	application/json	Creazione, modifica, eliminazione e prelevamento dei dati relativi ad una mappa
CRUD client	<i>/client</i>	POST,	application/json	Creazione, modifica, eliminazione e prelevamento dei



		PUT, DELETE		dati relativi ad un client
CRUD utente	<i>/user</i>	POST, PUT, DELETE	application/json	Creazione, modifica, eliminazione e prelevamento dei dati relativi ad un utente
CRUD gruppi	<i>/group</i>	POST, PUT, DELETE	application/json	Creazione, modifica, eliminazione dei dati relativi ad un gruppo di utenti
Informazioni su tag RFID	<i>/rfid/{id}/{mapid}</i>	GET	application/json	Ricezione informazioni relative ad un tag rfid (id) all'interno di una mappa (mapid)
CRUD tag RFID	<i>/rfid</i>	POST, PUT, DELETE	application/json	Creazione, modifica, eliminazione dei dati relativi ad un tag RFID
Informazioni su tag RFID	<i>/rfid/{id}</i>	GET	application/json	Ricezione informazioni relative ad un tag rfid (id)
Ricerca POI	<i>/poi/search</i>	POST	application/json	Ricerca punti di interesse a seconda dei parametri specificati nella richiesta

Tabella 1: Interfaccia client server



3.4 CAMMEO Service

Il CAMMEO Service presenta un'implementazione relativamente semplice. E' costituito da due componenti principali, il Localization Device Manager e il Server Interface come presentato in Figura 3.

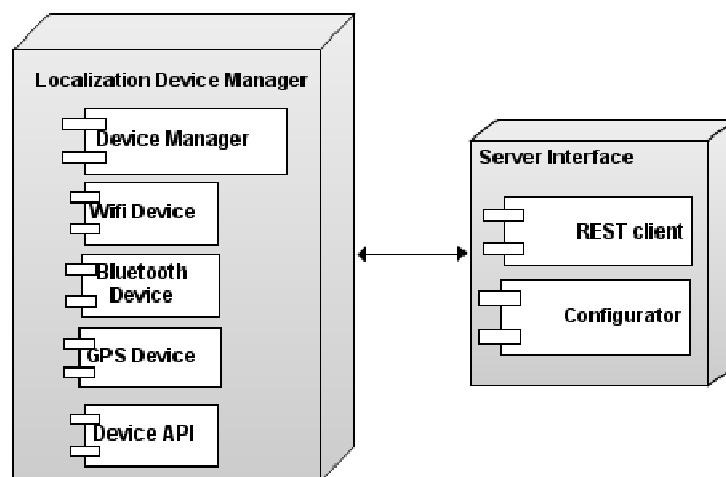


Figura 3: CAMMEO Service - Architettura di secondo livello

3.4.1 Localization Device Manager

E' il componente principale del CAMMEO Service. E' costituito da un insieme di moduli che si occupano di reperire autonomamente i dati necessari alla localizzazione tramite le differenti tecnologie a disposizione. Come indicato in figura dispone di 3 moduli principali WiFi Device, Bluetooth Device e GPS Device, che consentono il recupero dei dati grezzi di localizzazione derivanti dalle rispettive tecnologie. Questi 3 moduli devono rispettare le API imposte dal quarto modulo, il Device API. Queste API permettono l'utilizzo delle seguenti funzionalità:

- Avvio della ricezione delle informazioni grezze di posizione
- Interruzione della ricezione delle informazioni grezze di posizione
- Visualizzazione dello stato del modulo
- Visualizzazione delle ultime informazioni grezze di posizione

Il Device Manager a sua volta gestisce ognuno di questi moduli, ne verifica lo stato e ne richiede le informazioni ad intervalli regolari, valutandone la correttezza.

Ogni modulo Device è collegato ad una specifica tecnologia di localizzazione:

- WiFi Device: è collegato ai driver che gestiscono la scheda di rete wireless dello Smart phone in cui è installato il client. Questo modulo ha la possibilità di richiedere al driver della scheda di rete di effettuare delle scansioni per la ricerca di access point WiFi



nelle vicinanze. Ad ogni scansione identifica e codifica le seguenti informazioni (specificate anche nel capitolo 3.2):

- Indirizzo MAC dell'access point
 - RSSI
 - Potenza di trasmissione
 - Canale di comunicazione
 - Timestamp del momento della scoperta di questo AP
- Bluetooth Device: il modulo si dovrà collegare ai driver per la scheda Bluetooth collegata al client. Dovrà essere possibile effettuare una scansione dei nodi bluetooth vicini identificando il loro UUID e le caratteristiche del segnale.
 - GPS Device: questo modulo è collegato con i driver del ricevitore GPS, interpreta i dati NMEA e ne estrae la posizione dell'utente.

Ognuno di questi moduli è autonomo ed autogestito, dovranno essere implementate procedure ridondanti per il riavvio forzato del modulo in caso di malfunzionamenti o errori. Il collegamento con i rispettivi driver dovrà rispettare quanto più possibile il paradigma hot-plug per consentire il rilevamento automatico del rispettivo hardware senza dover interrompere l'applicazione.

3.4.2 Server interface

Costituisce l'interfaccia di collegamento con il CAMMEO Server. È costituito da un client con architettura REST e permette di contattare il server per inviare i dati grezzi rilevati dal Localization Device Manager. Deve implementare le funzionalità per la codifica dei dati rilevati dai vari moduli di localizzazione verso JSON per la comunicazione con il server.

Non è prevista una fase di autenticazione utente nel client, ma per ogni transazione con il server è necessario specificare le credenziali utente nell'header HTTP. Il server, tramite il Request Manager verificherà le credenziali e autorizzerà o meno la comunicazione.

3.4.3 User Interface e Admin Interface

Questi due componenti rappresentano l'interfaccia di sistema a cui un utente o un amministratore può accedere per poter visualizzare mappe, posizioni, percorsi e configurazioni.

Per mantenere un'ampia portabilità del sistema e permetterne l'uso su differenti sistemi operativi l'interfaccia utente dovrà essere sviluppato come web application. Con gli ultimi miglioramenti del linguaggio HTML5 [12] rispetto ai precedenti standard HTML4 e XHTML è possibile realizzare interfacce utente dall'alto contenuto interattivo e con una presentazione consistente nella maggior parte dei browser per desktop e per dispositivi mobili.



Visto che tra attori di sistema molte operazioni sono comuni (come la visualizzazione di una mappa e delle posizioni), è stata valutata la necessità di riutilizzare nell'interfaccia utente lo stesso codice e gli stessi framework per ottimizzare il lavoro, ridurre le possibilità di errori e fornire una piattaforma stabile.

L'interfaccia utente presenta differenti funzionalità a seconda dei differenti utenti che la utilizzano (User o Admin). E' possibile comunque definire due schermate che possono essere utilizzate per interagire con il sistema: l'interfaccia lato client utilizzata dagli utenti finali, e l'interfaccia lato server usata dagli amministratori di sistema e dall'amministratore del server di mappe.

Le scelte tecnologiche ricadono sui framework Java Server Faces e Facelets, oltre che sull'utilizzo di JavaScript per determinate funzioni di auto completamento, validazione form lato browser e interazione con il server se necessario.

Per la visualizzazione delle mappe, dei punti di interesse e della posizione utente le scelte ricadono sul framework OpenLayers [13]. Un framework in JavaScript che fornisce le principali funzionalità di interazione (pan, zoom, drag) e visualizzazione di mappe, vettori e immagini.

Il contenuto dell'applicazione web sarà quindi presentato in HTML con il supporto di CSS per la presentazione grafica.

3.4.4 Lato Utente

L'interfaccia lato utente, deve presentare tutte le funzionalità che permettono l'utilizzo della piattaforma CAMMEO da parte di un utente finale. L'applicazione web che sarà realizzata dovrà permettere, come da requisiti, l'autenticazione utente, la visualizzazione della mappa, la rappresentazione della posizione utente, i punti di interesse e i percorsi richiesti.

In particolari condizioni deve essere possibile visualizzare ogni utente, con un breve dettaglio sulle sue informazioni, sulla mappa.

3.4.5 Lato Amministratore

Per l'utenza lato Amministratore devono essere disponibili le interfacce di configurazione del sistema, dalla gestione degli utenti e client alla visualizzazione delle mappe e delle posizioni di ciascun utente. Deve essere possibile inserire modificare o eliminare mappe indoor e specificarne le configurazioni.

Queste operazioni di back-end dovranno essere accessibili tramite un sistema di autenticazione e permettere di modificare in tempo reale le configurazioni della piattaforma, senza incidere sull'operato degli altri utenti utilizzatori. Ogni operazione eseguita dovrà quindi essere "atomica" e riflettersi nel sistema senza la necessità di un riavvio o di un downtime momentaneo della piattaforma.



3.4.6 Relazioni tra i componenti di secondo livello

Di seguito è riportata l'interfaccia che dovrà utilizzare il CAMMEO Service per integrare i vari dispositivi di localizzazione (WiFi, Bluetooth, RFID, ecc). Questa interfaccia deve rendere omogenee le varie tecnologie in modo che il Device Manager possa trattarle tutte allo stesso modo:

Metodo	Valori ritornati	Descrizione
Enable	<i>true</i> se il dispositivo è stato abilitato con successo, <i>false</i> altrimenti	Abilita l'uso del dispositivo di localizzazione implementato e avvia i processi per rilevare i dati grezzi di posizione
Disable	<i>true</i> se il dispositivo è stato disabilitato con successo, <i>false</i> altrimenti	Disabilita il dispositivo di localizzazione, fermando i processi di rilevamento dei dati di posizione
Get Status	<i>Enabled</i> : Se il dispositivo è abilitato <i>Disable</i> : se il dispositivo è disabilitato <i>Enabling</i> : se il dispositivo è in fase di abilitazione <i>Disabling</i> : se il dispositivo è in fase di disabilitazione <i>Unknow</i> : se lo stato del dispositivo non è definito <i>Scanning</i> : se il dispositivo è attivo ed è impegnato nella "scansione" dei dati di posizione	Permette di identificare lo stato in cui si trova il dispositivo di localizzazione.
Get Relative Position	La posizione relativa alla tecnologia del dispositivo di localizzazione	Permette di ricevere gli ultimi dati grezzi di posizione.

Questo semplice set di metodi dovranno essere implementati per ogni tecnologia di localizzazione a disposizione; inizialmente lo sviluppo dovrà comprendere le tecnologie WiFi, e GPS, in un secondo momento sarà possibile aggiungere e aggiornare le tecnologie a disposizione semplicemente implementando questi metodi, senza dover modificare il Device Manager o ogni altro tipo di componente software.

Tra i metodi indicati particolare attenzione ha il metodo *Get Relative Position* utilizzato per ricavare, dal dispositivo, l'ultimo dato di posizione utile. Questo metodo viene chiamato



direttamente dal componente Server Interface del CAMMEO Service, ad intervalli regolari, per poter inviare i dati grezzi di ogni dispositivo al server centrale utilizzando l'URI "/position/{id}" e il metodo PUT per aggiornare i dati relativi alla posizione utente. Questa interazione è illustrata nei seguenti diagrammi di sequenza.

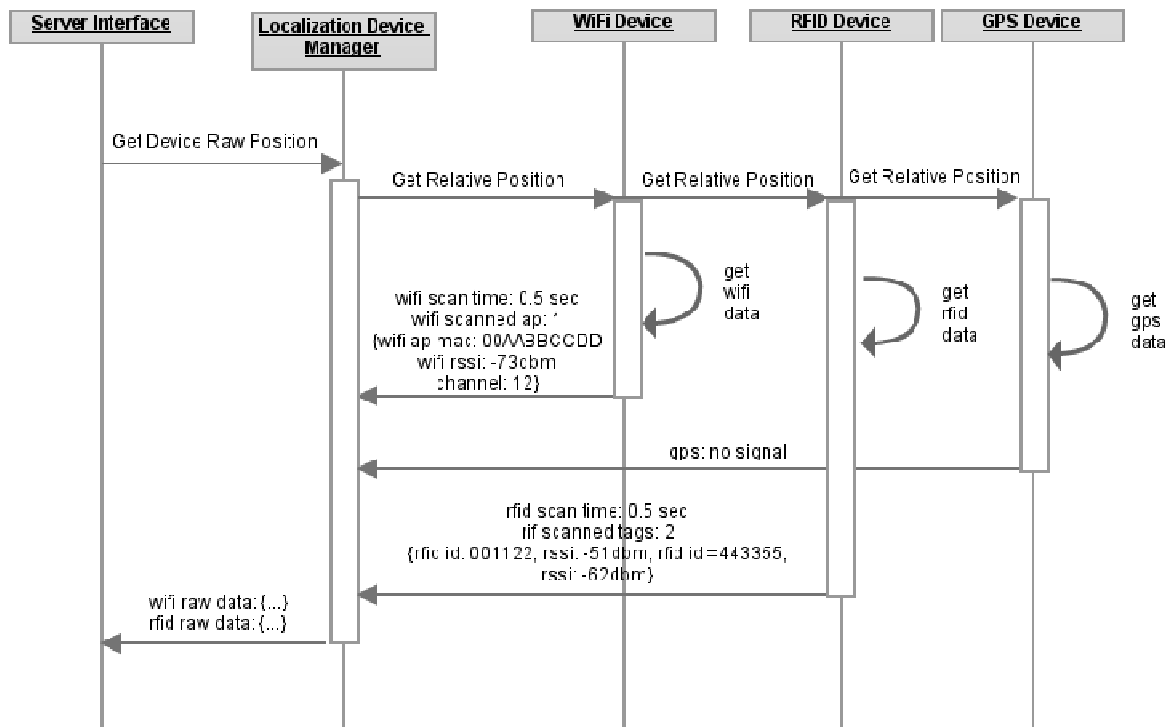


Figura 4: diagramma di sequenza per rilevare i dati grezzi di posizione

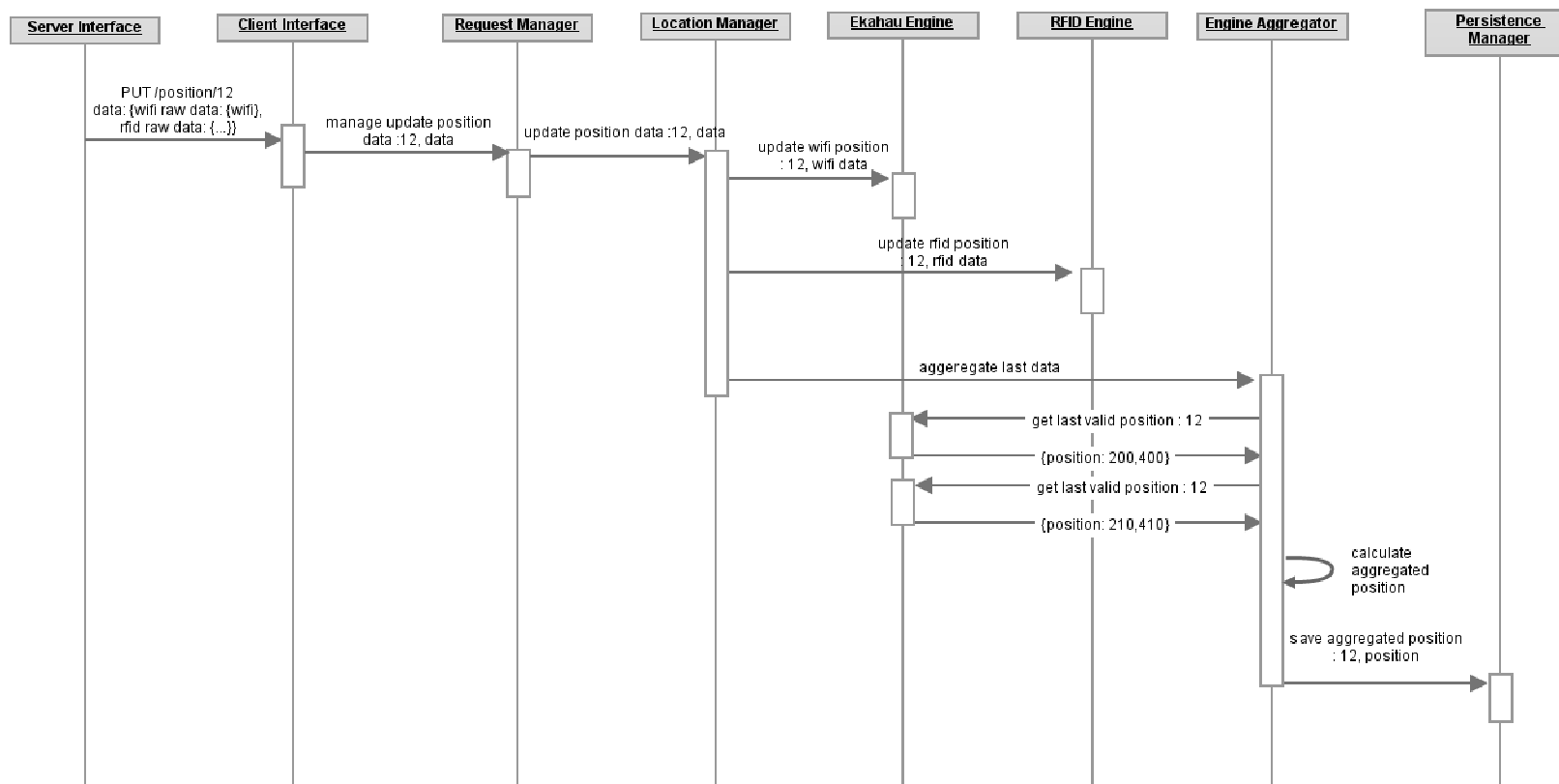


Figura 5: diagramma di sequenza per aggiornare i dati di posizione del client



Mappe e posizioni verranno richieste al CAMMEO Server passando per un browser web. Il CAMMEO Service ha l'unico compito di dover inviare al CAMMEO Server i dati grezzi della sua posizione

Lo schema seguente rappresenta la comunicazione interfaccia utente e server per la richiesta di posizione e di visualizzazione delle mappe.

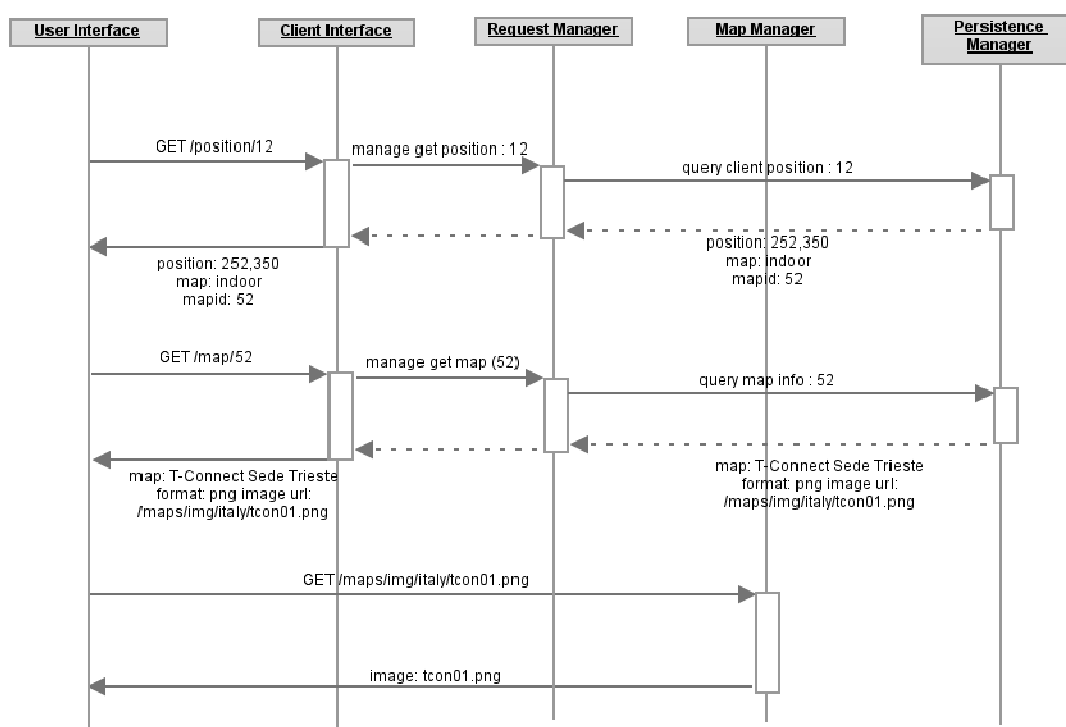


Figura 6: Richiesta posizione e mappa indoor

Si può notare come in Figura 6 le mappe outdoor vengano richieste tramite i servizi WMS/GeoCache forniti dal GeoServer, mentre in Figura 7 la richiesta di mappe viene fatta alla repository di immagini statiche.

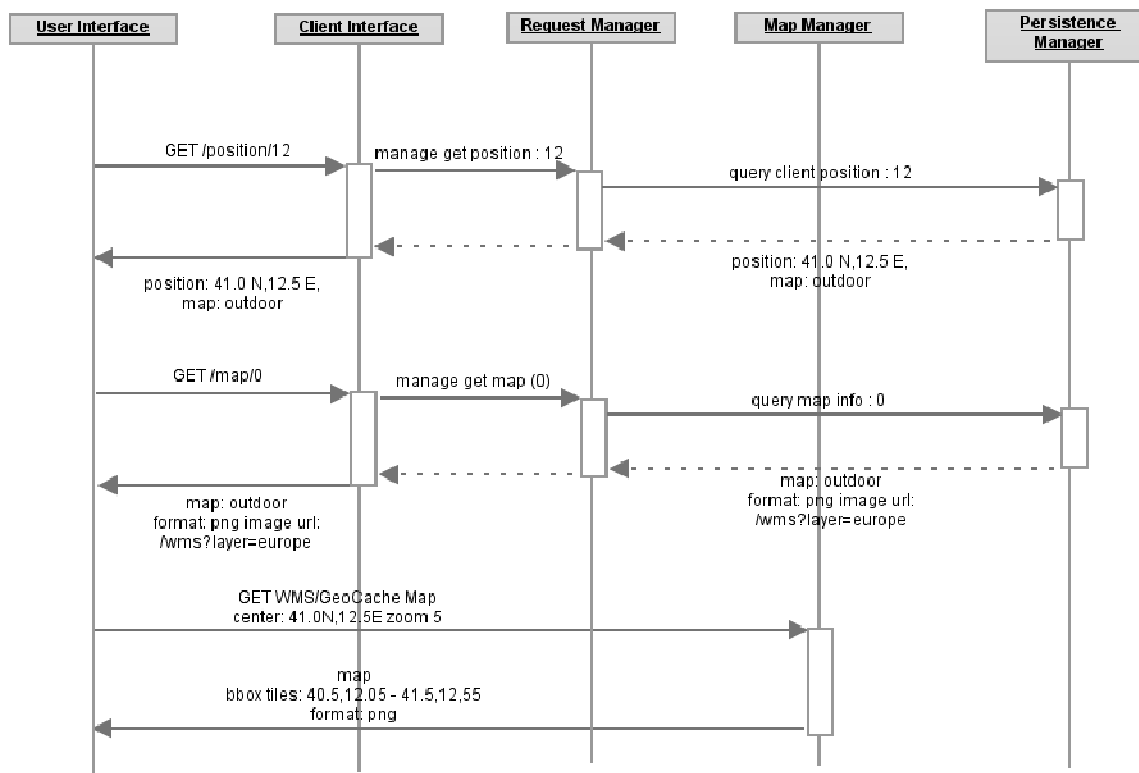


Figura 7: Richiesta posizione e mappa outdoor

Nella successiva tabella sono rappresentate le interfacce relative al modulo Localization Engine API che forniscono il supporto all'integrazioni di ulteriori tecnologie di localizzazione.

Metodo	Parametri	Valori ritornati	Descrizione
Update Client Raw Data	Dati grezzi di posizione provenienti sia dal Service che dal Transceiver CAMMEO	<i>true</i> se l'aggiornamento è andato a buon fine, <i>false</i> altrimenti	Permette al Service o al Transceiver CAMMEO di inviare i dati grezzi di posizione e aggiornare i dati di posizione.
Get Client Position	L'id del client CAMMEO	La posizione del client calcolata con questo motore di localizzazione, comprendente id della mappa di riferimento	Consente di richiedere al motore di localizzazione la posizione di un client CAMMEO, relativo all'id passato come parametro.



3.5 CAMMEO Transceiver

Il CAMMEO Transceiver è un sistema pensato per tutti, anche per le persone che non hanno dimestichezza con le ultime novità tecnologiche e che non vogliono avere a che fare con un telefonino, ma allo stesso tempo vogliono usufruire di servizi di posizionamento per applicazioni di diverso tipo, quali richiesta di soccorso in caso di malori.

L'architettura di secondo livello del CAMMEO Transceiver presenta i componenti riportati in Figura 8:

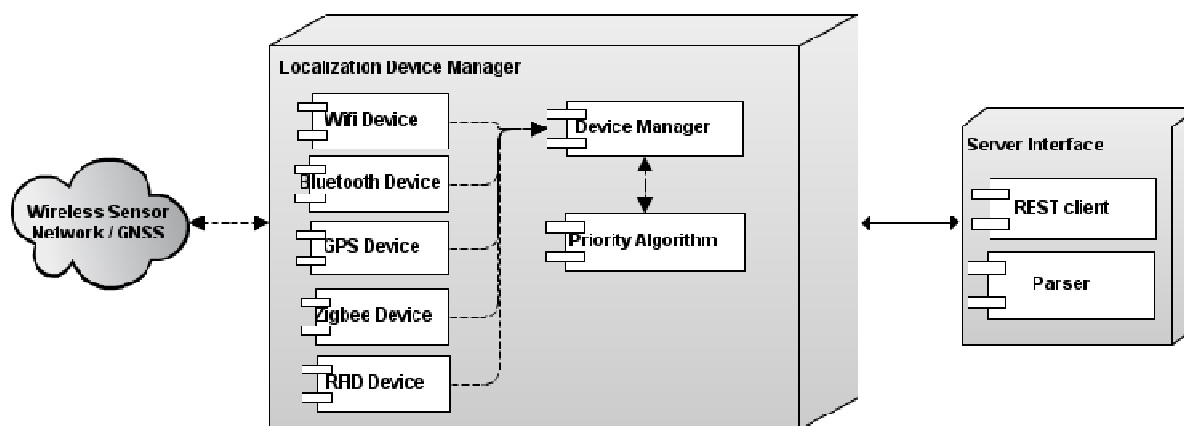


Figura 8: Architettura di secondo livello del CAMMEO TRANSCEIVER

Il modello di sistema è costituito da tre componenti principali:

- Wireless Sensors Network / GNSS (per comodità la indicheremo come WSNG). Questo componente fornirà i dati indispensabili per la localizzazione grazie alla rete di sensori o al sistema di localizzazione satellitare.
- Localization Device Manager (Piattaforma HW), che costituisce il cuore del CAMMEO Transceiver. Esso contiene l'algoritmo di priorità, gestore per l'acquisizione dei dati, dai dispositivi connessi, indispensabili per gli algoritmi di localizzazione.
- Server Interface, che costituisce l'interfaccia con il CAMMEO Server per l'invio dei dati di posizione.

Questi tre componenti dell'architettura, sono a loro volta composti da ulteriori moduli funzionali descritti in maggiore dettaglio nei successivi paragrafi.

Tale architettura è supportata dal CAMMEO Server per le seguenti funzioni:

- utilizzare il corretto algoritmo di multilaterazione (all'interno del Localization Manager del CAMMEO Server), a seconda dei dati grezzi di posizione inviati dal CAMMEO



Transceiver. Il risultato di questa funzione permetterà di ottenere la posizione assoluta relativa all'ambiente in cui si trova l'utente.

- visualizzare la posizione assoluta dell'utente CAMMEO Transceiver tramite le interfacce grafiche fornite dal CAMMEO Server.

3.5.1 Wireless Sensor Network / GNSS (WSNG)

La WSNG rappresenta la rete di sensori dislocati nelle aree dove avviene la localizzazione, e che permette la georeferenziazione del CAMMEO Transceiver. Tale WSNG si compone di:

- fari WiFi, ZigBee, Tag RFID e sensori Bluetooth per quel che concerne le reti indoor,
- costellazioni GPS/GNSS per quanto riguarda la rete outdoor.

Un faro è dispositivo in grado di scambiare messaggi con il CAMMEO Transceiver o inviare informazioni di posizionamento. Tali dispositivi sono dislocati in una posizione nota al sistema CAMMEO e sono identificati da un ID univoco. Sono esempi di fari i satelliti, gli access point WiFi, dei nodi ZigBee, degli peer Bluetooth e i tag RFID.

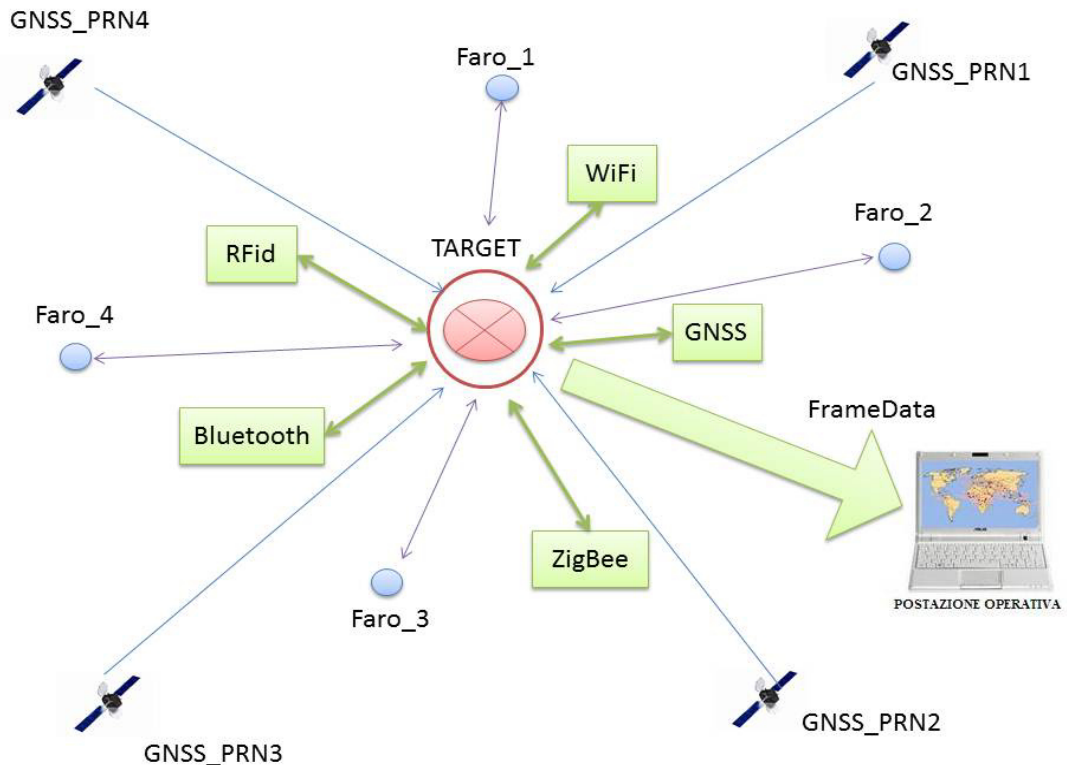


Figura 9: Wireless Sensor Network / GNSS

3.5.2 Localization Device Manager

Questo componente, integrato all'interno di un sistema operativo Linux like, costituisce il cuore del CAMMEO Transceiver. A questo componente sono collegati i device hardware di comunicazione (WiFi, Zigbee, Bluetooth, RFID e GPS)

Questo componente gestisce le seguenti tipologie di dati:

- Gestione delle Air Interfaces a bordo.
- Implementazione algoritmo di priorità.

3.5.2.1 Priority Algorithm

L'algoritmo di priorità è necessario per poter attivare fra le tecnologie presenti a bordo del transceiver, quella a più alto guadagno e disponibilità secondo una lista decisa a priori e inserita nell'algoritmo stesso. Il transceiver non presenta la caratteristica di usufruire di due tecnologie in parallelo, ciò è stato deciso per poter ottimizzare le risorse energetiche a disposizione, e per poter interagire di volta in volta con la singola tecnologia che presenta il miglior guadagno in termini di visibilità e disponibilità di network.



La gestione della priorità è affidata all'algoritmo di priorità, il quale valuta quale tecnologia è a disposizione nel momento della richiesta di localizzazione e successivamente avvia la localizzazione. Tale blocco è rappresentato nella Figura 10:

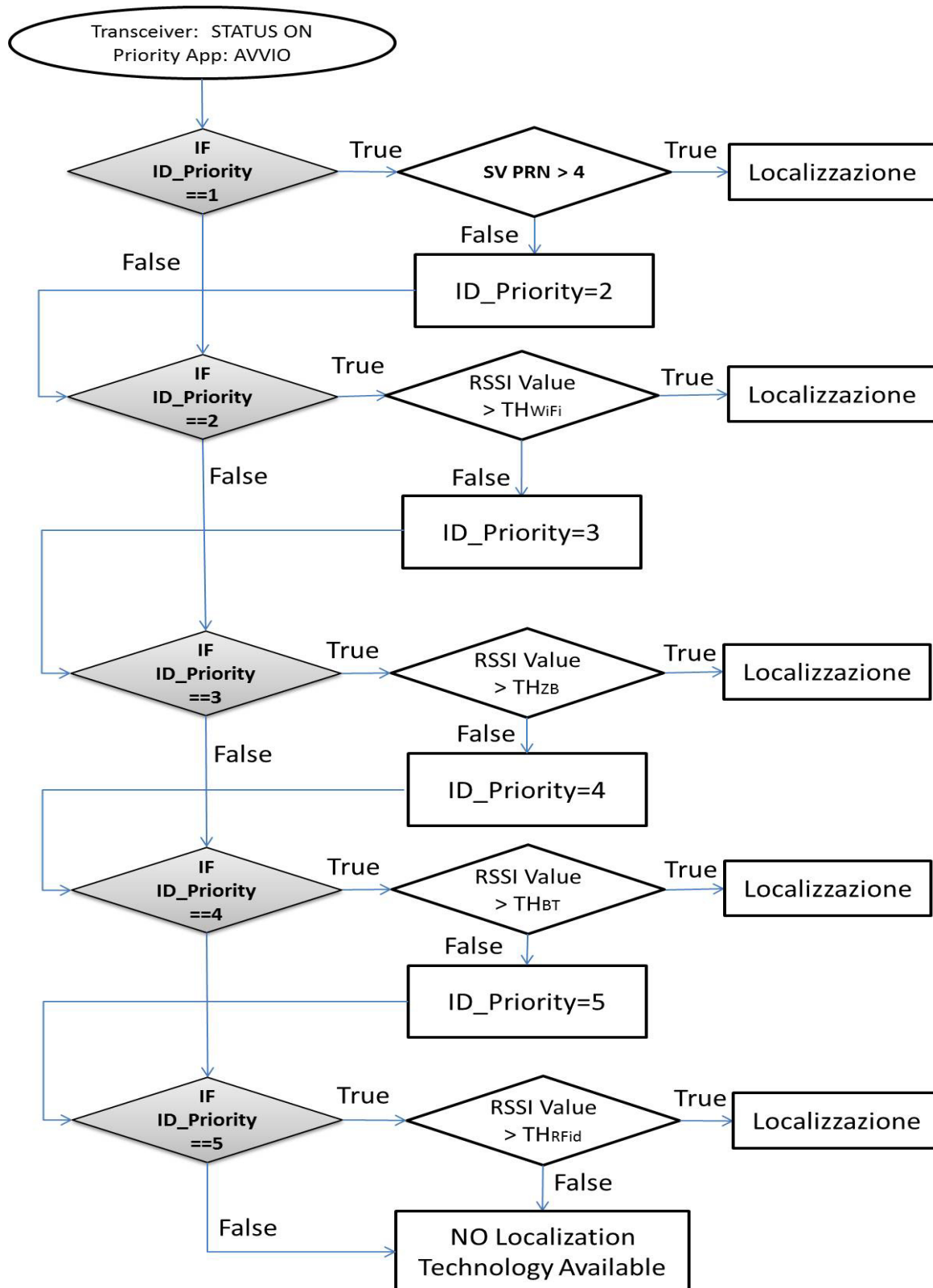




Figura 10: Diagramma di flusso del Priority Algorithm

L'algoritmo di priorità si basa sulla Tabella 2 : , che specifica l'ordine con cui vengono interrogati i dispositivi di localizzazione. Andando in ordine crescente, la prima tecnologia i cui valori di priorità (RSSI o numero di costellazioni) superano un determinato valore di soglia, viene scelta e utilizzata come tecnologia di localizzazione.

Tabella di priorità	
<i>ID_Priority</i>	<i>HW Device</i>
1	GNSS
2	WiFi
3	ZigBee
4	Bluetooth
5	RFId

Tabella 2 : Tabella di priorità tecnologie CAMMEO Transceiver

Nel caso in cui l'HW Device selezionato è la tecnologia GNSS, la valutazione sulla disponibilità è basata dal numero minimo di satelliti in vista ($SV > 4$). Nel caso di tecnologie indoor, la valutazione del superamento della soglia, indicata nel diagramma con la verifica del valore $RSSI_{value} > TH$, prevede che la singola tecnologia effettui una misura della potenza ricevuta e poi, se il segnale misurato è maggiore del valore di Threshold (TH), attivi i sottosistemi per determinare i dati di localizzazione, altrimenti il processo passa alla tecnologia di ordine seguente, riverificandone i parametri richiesti.

3.5.3 Server interface

In questa Sezione vengono descritti i flussi dati tra i sottosistemi componenti il CAMMEO Transceiver.

3.5.3.1 Parser

I dati elaborati dall'algoritmo di priorità, che rappresentano i dati grezzi di localizzazioni rilevati a seguito dell'algoritmo di priorità, vengono integrati, mediante l'operazione di parsing, in un data-frame che sostituirà la stringa JSON da inviare al CAMMEO Server per le successive elaborazioni. La struttura di questo data-frame è relativa alla provenienza dei dati di localizzazione.

Ad esempio per i dati di posizionamento GPS, un esempio di data frame in format di string JSON sarà il seguente:

```
{"gps":{"longitude":"41.142435","latitude":"13.2344325"}}
```

Questa stringa sarà passata al REST Client per il successivo invio al CAMMEO Server.



Per quanto riguarda le tecnologie Zigbee, il data frame sarà composto dalla seguente trama:

Identificativo	Contenuto
MSG_1	✓ ID_Trama: identificativo della frame data, univoco e distinto. ✓ ALARM, identificativo di allarme. ✓ ID_Net: identificativo della WSN
MSG_2	Stringa contenente i dati di localizzazione.
MSG_3	Stringa di caratteri opzionale, per futuri sviluppi.

Tabella 3: Contenuto del data frame

In notazione JSON:

```
{"zigbee":["MSG_1":"ID_TRAMA;ALARM;ID_NET","MSG_2":"LOC","MSG_3":"EXTRA"]}
```

In dettaglio, il campo MSG_1 contiene i dati, sotto forma di stringa caratteri, utili per la gestione e determinazione del target in riferimento alla network indoor/outdoor in uso.

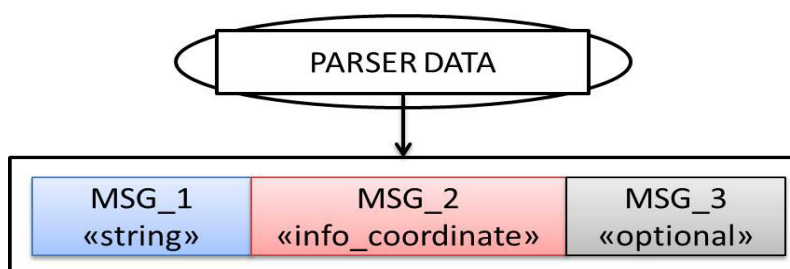


Figura 11: Frame Incapsulation structure

3.5.3.2 REST client

Questa interfaccia si occupa di creare il collegamento di rete tra il CAMMEO Transceiver e il CAMMEO Server. È un'interfaccia che utilizza il protocollo HTTP per contattare i web services REST presenti nel server. I dati precedentemente parsati dal modulo Parser vengono inviati al server tramite questa interfaccia.

3.5.4 Zigbee/Bluetooth/etc Engine

Il CAMMEO Transceiver necessita di particolari motori di localizzazione che differiscono in parte da quelli utilizzati per il CAMMEO Service e già presenti nel CAMMEO Server.

Il CAMMEO Server sarà quindi corredato dei seguenti motori di localizzazione:



- Zigbee Engine: conterrà gli algoritmi di multilaterazione per il calcolo della posizione del Transceiver per le tecnologie Zigbee.
- Bluetooth CTransceiver Engine: conterrà gli algoritmi di prossimità per il calcolo della posizione del Transceiver per le tecnologie Bluetooth.

Come avviene per i motori di localizzazione dedicati al CAMMEO Service, l'Engine Aggregator verifica la natura dei dati di posizione grezzi inviati dal CAMMEO Transceiver. Se provengono da tecnologie indoor, viene ricercato e attivato il rispettivo motore di localizzazione relativo. Se invece i dati provengono dalle tecnologie outdoor, viene valutata la veridicità dei dati (correttezza dei dati GNSS) e vengono memorizzati direttamente nel Persistence Manager.

Il grafico seguente specifica la procedura, appena descritta, per la scelta dei motori di localizzazione da parte dell'Engine Aggregator.

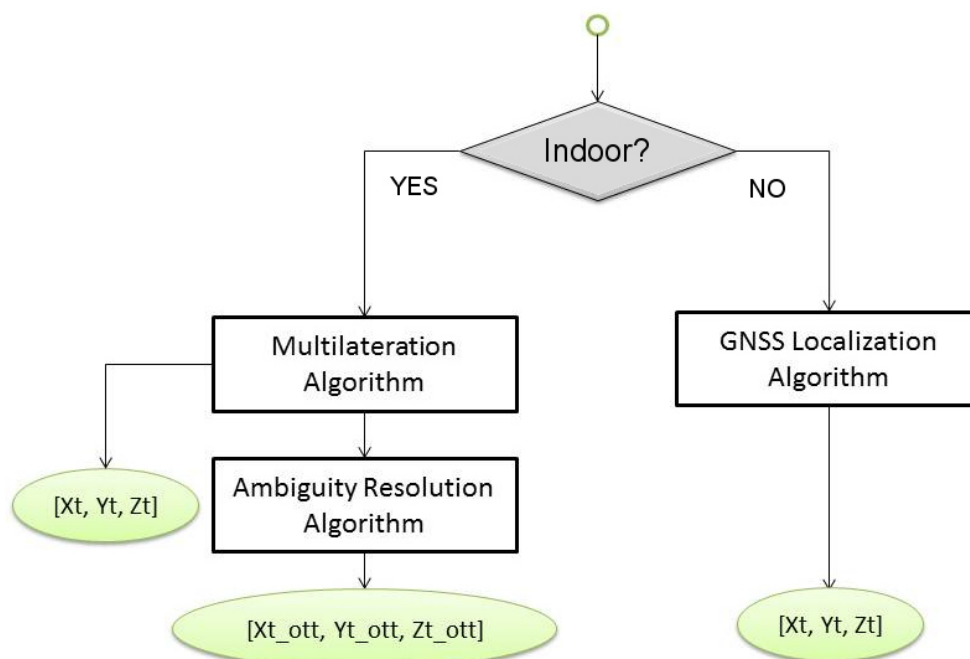


Figura 12: Diagramma di flusso dell'Engine Aggregator

3.5.5 Relazioni tra i componenti di secondo livello

All'interno del CAMMEO Transceiver, nel momento di richiesta di localizzazione, avvengono le seguenti procedure:



- a. Avvio della connessione, in base alla tabella di priorità, del prossimo dispositivo disponibile. Il primo HW device in ordine di avvio è il dispositivo GNSS (vedi Tabella 2).
- b. Nel caso di selezione di tecnologia outdoor, la verifica è basata sulla visibilità dei satelliti, che deve superare il valore minimo. Nel caso di selezione di tecnologia indoor, verificare se per ognuna delle tecnologie indoor è possibile stabilire una connessione con almeno tre dispositivi (fari) dello stesso tipo. Se non è possibile la scheda relativa torna in standby, bisogna quindi attivare la successiva e ripetere il passo b. Il modulo che si occupa di stabilire l'indice di priorità e quindi attivare la relativa tecnologia è il modulo di Priority Algorithm.
- c. Qualora non sia possibile ricevere dati da 3 fari dello stesso tipo, per nessuno dei dispositivi considerati, ripetere periodicamente la procedura dal passo b. Se, invece, è possibile connettersi solo ad una tipologia avente almeno tre fari, andare al passo successivo, ma periodicamente verificare se vi sono altri dispositivi disponibili.
- d. Calcolare la distanza da ognuno dei fari per la prima tipologia scelta (tipologia di Sensor Network selezionata e disponibile).
- e. In caso di richiesta utente, se la tecnologia selezionata è indoor, costruire una stringa contenente i dati relativi alla rete di sensori, comprensivi di ID e coordinate degli stessi. Nel caso di tecnologia outdoor (per il GPS), verrà creata la stringa GPS_longitudine_latitudine.
- f. Inviare tale stringa tramite WiFi.



4 Architettura di deployment

La Figura 13: mostra la visione d'insieme delle tecnologie che saranno utilizzate per lo sviluppo e per il deploy della piattaforma.

Ciascun utente di un CAMMEO Service interagirà direttamente con la piattaforma tramite un Browser web. Questo browser utilizzerà standard web e il framework OpenLayers per visualizzare le informazioni di posizione, le mappe, i punti di interesse, le configurazioni e quant'altro disponibile. Utilizzerà una connettività REST per comunicare con l'Application Server nel quale è contenuto il CAMMEO Server. Ogni richiesta ad una schermata dell'applicazione verrà renderizzata tramite la tecnologia Java Server Faces messa a disposizione Dall'application Server.

Il CAMMEO Service verrà invece dislocato in uno smartphone e avrà accesso almeno alle due principali fonti per il recupero di dati di posizione (una scheda di rete WiFi e un ricevitore GPS). La comunicazione con questi 2 dispositivi avverrà utilizzando i driver nativi del sistema operativo sul quale è installato il CAMMEO Service.

Il CAMMEO Transceiver verrà invece dislocato in un dispositivo hardware embedded dotato di processore ARM9. Questo sistema sarà corredato di interfacce di comunicazione: GPS (GNSS), WiFi, Zigbee, Bluetooth, e un RFID reader. La comunicazione tra i devices e il sistema operativo avverrà tramite driver nativi (o del produttore del device stesso), mentre per l'interfacciamento con i dispositivi di acquisizione, verranno sviluppati drivers/applicativi appositi.

Il CAMMEO Server verrà dislocato nell'Application Server GlassFish (versione 3) che implementa le ultime specifiche di Java Enterprise Edition 6.0.

Il server GeoServer per gestire le mappe outdoor sarà anch'esso dislocato nell'Application Server Glassfish.

L'Ekahau Positioning Engine, software proprietario, sarà adattato e anch'esso integrato nello stesso Application Server di CAMMEO. Attualmente questo motore di localizzazione WiFi utilizza Tomcat come servlet server e PostGres come database. La migrazione verso Glassfish e un database Derby dovrebbe risultare particolarmente semplice e priva di pericolose conseguenze.

Come per Ekahau, il database utilizzato per il CAMMEO Server sarà Derby, attualmente integrato all'interno dell'Application Server Glassfish.

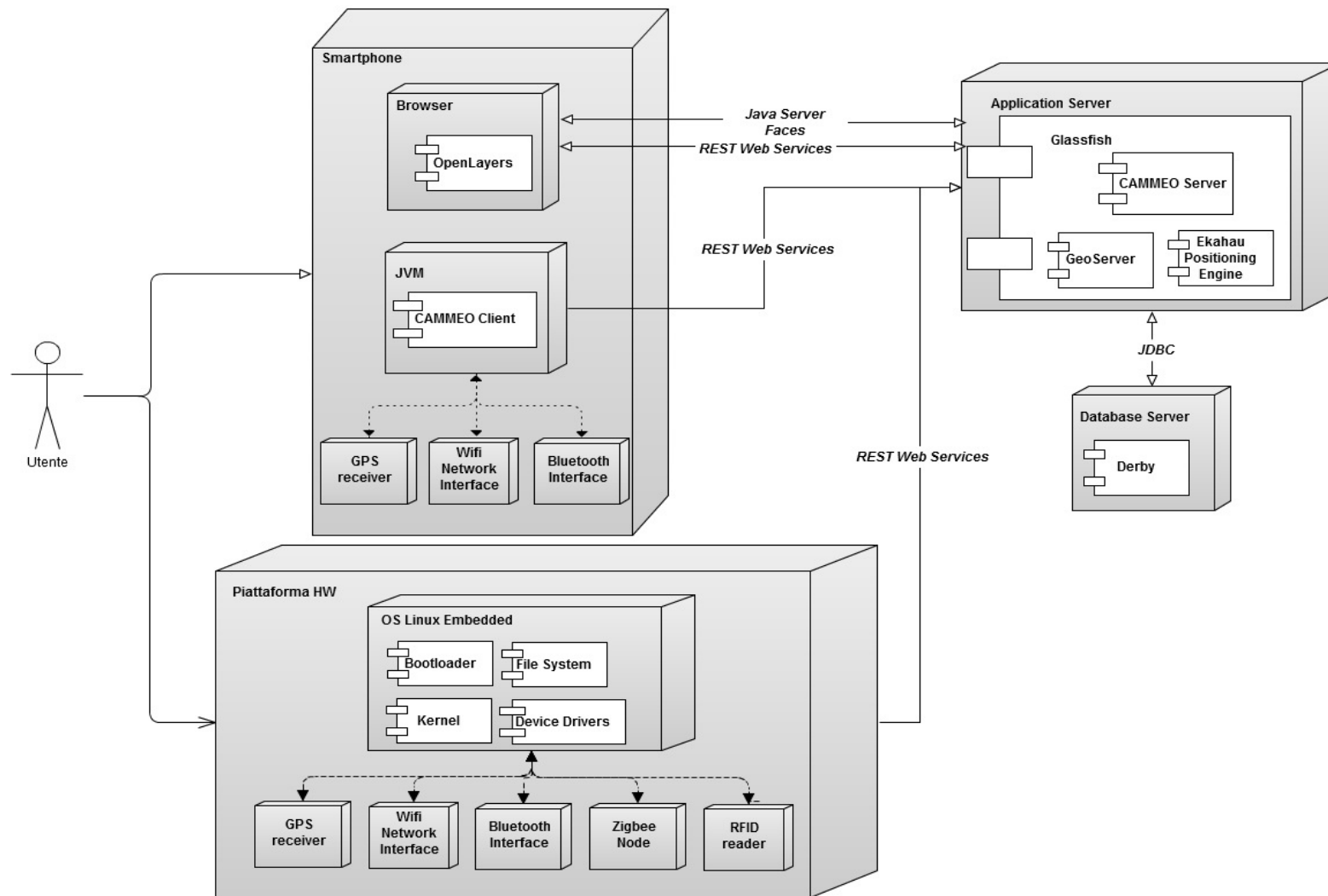


Figura 13: Schema del deploy della piattaforma CAMMEO



5 Conclusioni

In questo documento è stato descritto il design dell'architettura e le interfacce di comunicazione tra i vari moduli che compongono il sistema CAMMEO.

Come viene evidenziato più volte, l'architettura è costituita principalmente da tre macro componenti principali: il CAMMEO Transceiver, il CAMMEO Service e il CAMMEO Server.

I linguaggi di programmazione e i framework che sono stati ipotizzati in questo documento non costringono il team di sviluppo ad appoggiarsi su un particolare server, database o sistema operativo. Essi in effetti prevedono la possibilità di:

- Integrare o creare una user interface diversa da quella ipotizzata utilizzando l'interfacce di comunicazione messe a disposizione dal server.
- Riutilizzare il CAMMEO Service in ogni dispositivo che integri una Java Virtual Machine, ad esempio un PC o un dispositivo embedded come il Sun SPOT [24].
- Portare il CAMMEO Server verso un application server o un database differenti.
- Cambiare il server di mappe outdoor o appoggiarsi a server on the cloud o già server commerciali già esistenti online (Google Maps, Bing Maps, OpenStreetMaps etc)
- Distribuire il calcolo dei vari motori di localizzazione, specialmente per il calcolo delle posizioni tramite WiFi, verso altri server, anche remoti mantenendo la stessa architettura.
- Consentire di creare e integrare nuovi dispositivi embedded come il CAMMEO Transceiver mantenendo un'unica interfaccia di comunicazione al server.

Durante e a seguito dell'implementazione dei tre macro componenti, verranno effettuati dei test di validazione (di unità o di integrazione) che consentiranno di valutare le funzionalità ricercate della soluzione.



6 Bibliografia

- [1] Unified Modeling Language <http://www.uml.org>
- [2] JSON JavaScript Object Notation <http://json.org>
- [3] Ekahau RTLS <http://www.ekahau.com>
- [4] GeoServer <http://geoserver.org/>
- [5] Open Geospatial Consortium <http://www.opengeospatial.org/>
- [6] ORM Object Relational Mapping, <http://www.orm.net>
- [7] RDBMS Remote DataBase Managment System,
<http://www.dbms2.com/category/database-theory-practice/>
- [8] GlassFish Application Server <http://glassfish.java.net>
- [9] EclipseLink, <http://www.eclipse.org/eclipselink/>
- [10] Apache Derby, <http://db.apache.org/derby/>
- [11] JSR 311 JAX-RS: The Java API for RESTful Web Services
<http://jcp.org/en/jsr/detail?id=311>
- [12] HTML5 Specification, <http://dev.w3.org/html5/spec/Overview.html>
- [13] OpenLayers, <http://openlayers.org>
- [14] Piattaforma di sviluppo Cammeo Transceiver: <http://arm.mini-box.com/>
- [15] S.O. Linux: http://www.openembedded.org/index.php/Main_Page
- [16] GNSS: <http://galileo.cs.telespazio.it/liaison/>
- [17] WiFi: <http://www.ieee802.org/11/>
- [18] ZigBee: <http://www.ieee802.org/15/pub/TG4.html>
- [19] NMEA: <http://www.gpsinformation.org/dale/nmea.htm>
- [20] Struttura dati GPS: <http://www.gsmcontrol.biz/Applicazioni/SentenzeNMEA.htm>
- [21] RSSI: "RSSI Based Location-Aware PC Power Management", Zhong-Yi Jin and Rajesh K. Gupta, University of California, San Diego
- [22] Deliverable D1.1 RS
- [23] Deliverable D1.2 SRS
- [24] Sun SPOT <http://www.sunspotworld.com/index.html>