Università degli Studi di Pisa

SCUOLA DI INGEGNERIA
Corso di Laurea Magistrale in Computer Engineering

Tesi di Laurea Magistrale

Progetto e sviluppo di un servizio di lookup basato su hash table distribuite per la piattaforma FIWARE

Candidato:
Daniele Gori

Relatori:
Prof. Enzo Mingozzi
Prof. Giuseppe Anastasi

Anno Accademico 2015/2016

Indice

Elenco delle figure

Capitolo 1

Introduzione

L'*Internet of Things* (IoT) è un nuovo paradigma che recentemente ha guadagnato terreno nello scenario dell'Internet del Futuro. L'idea alla base di questo concetto è la presenza pervasiva intorno a noi di *oggetti* o *Things*, capaci di interagire e cooperare tra loro per raggiungere un comune obiettivo. L'idea dell'IoT avrà un elevato impatto sui diversi aspetti della vita di tutti i giorni e sul comportamento dei potenziali utenti sia in campo lavorativo e industriale sia in campo domestico, dove la domotica, assisted living, e-health e l'apprendimento potenziato sono solo alcuni esempi di possibili scenari applicativi in cui il nuovo paradigma giocherà un ruolo di primo piano nel prossimo futuro. Allo stesso modo, dal punto di vista degli utenti aziendali, le conseguenze più evidenti saranno ugualmente visibili in campi quali automazione e produzione industriale, logistica, commercio, gestione dei processi, di trasporto intelligente di persone e merci [1]. In IoT le *Things* sono abilitate a interagire e comunicare tra loro stesse e con il contesto in cui operano attraverso lo scambio di dati e di informazioni percepite e misurate dall'ambiente circostante. Possono reagire autonomamente agli eventi del mondo reale incidendo su di essi eseguendo azioni o creando servizi senza l'intervento umano. La visione del Future Internet prende in considerazione la fusione di reti di computer e IoT in una piattaforma IT globale basata sui protocolli di comunicazione standard, ciò comporta che i servizi capaci di interagire con le Smart Things utilizzino le interfacce standard necessarie a fornire il collegamento ad Internet, per interagire con essi [2].

L'Internet of Thing pone nuovi problemi riguardo agli aspetti di *networking*, infatti le things che compongono l'IoT sono caratterizzate da risorse limitate sia in termini computazionali che in termini di capacità energetica. Lo sviluppo dell'IoT porterà ad avere secondo stime riportate in [3] 16 miliardi di dispositivi connessi alla rete globale nell'anno 2020. Le soluzioni proposte per l'IoT dovranno quindi fare particolare attenzione all'efficienza energetica e di calcolo

e ai problemi di scalabilità, in modo da poter affrontare scenari complessi su larga scala in cui un grande numero di sensori e attuatori dovranno comunicare e interagire con applicazioni differenti. Un altro aspetto molto importante da affrontare è l'identificazione e la localizzazione di una risorsa/oggetto. Un'applicazione o un utente che vuole interagire con una Thing connessa attraverso Internet, deve sapere dove si trova e come connettersi ad essa. In un contesto come quello dell'IoT, dove i dispositivi sono interconnessi ad internet tramite connessioni wireless e soggetti a vincoli energetici, non è possibile sapere a priori se una thing è disponibile o meno. Per scoprire la disponibilità (*availability*) e recuperare le informazioni necessarie per connettersi con una risorsa è necessario fare affidamento ad un servizio di *discovery*. Il meccanismo di Discovery permette di ridurre la necessità di configurazione e amministrazione dei dispositivi connessi ad una rete [4]. I protocolli di discovery rendono possibile a chi ricerca una risorsa di identificare e di essere automaticamente consapevole delle funzionalità messe a disposizione dalle risorse connesse alla rete, senza la necessità di intervento umano. La capacità di self-configuring è un aspetto cruciale per poter realizzare applicazioni robuste in termini di resilienza ai cambiamenti, che possono accadere nel tempo in un contesto come quello dell'IoT dove la *availability* dei dispositivi non è sempre garantita. Un altro aspetto in cui la self-configuring fornisce un contributo fondamentale è la capacità di minimizzare o ridurre del tutto l'intervento umano per configurare e amministrare l'altissimo numero di dispositivi presenti in un tipico scenario IoT [5].

Le reti Peer-to-Peer (P2P) sono state progettate per fornire le caratteristiche desiderate per gestire sistemi su larga scala, come la scalabilità, la self-configuring e la resistenza ai guasti. Il principale punto di forza di questa architettura è il fatto che con l'aumentare del numero di nodi partecipanti alla rete, le prestazioni globali del sistema migliorano significativamente sia in termini di capacità di calcolo che di capacità di memorizzazione. Queste caratteristiche rendono questa architettura molto adatta a realizzare un servizio di discovery distribuito su larga scala attraverso la rete globale. Un aspetto fondamentale del meccanismo di Discovery è la capacità di localizzare le risorse secondo gli attributi richiesti. Sfruttando le architetture P2P basate su Distributed Hash Table (DHT) è possibile realizzare un servizio di discovery globale distribuito, in cui le informazioni sulle risorse sono distribuite tra i nodi che ne fanno parte. In questa tesi è stata analizzata una particolare soluzione di un servizio distribuito di lookup delle risorse che adotta un approccio peer-to-peer basato su DHT capace di gestire *query multi attributo* e *range-query* [6]. Tale soluzione fa riferimento all'approccio sviluppato in [7] chiamato LORM.

Un'altra piattaforma analizzata in questa tesi è FIWARE. Questa piattaforma è supportata dal progetto dell'Unione Europea Future Internet Public Pri-

vate Partnership (FI-PPP). Questo progetto ha lo scopo di sviluppare e mettere a disposizione secondo la policy dell'Open Source una piattaforma per la realizzazione di applicazioni innovative per l'internet del futuro. FIWARE si basa su elementi chiamati Generic Enablers (GE), i quali costituiscono i blocchi di base della piattaforma pronti per essere utilizzati. I vari Generic Enabler sono divisi in vari *Technical Chapters* in base al settore di utilizzo di cui fanno parte. L'attenzione principale è stata rivolta ai Generic Enablers forniti per l'Internet Of Things. Questi GE consentono alle *Thing* di diventare risorse di contesto NGSI, così da favorire l'interazione tra applicazioni basate su FIWARE e oggetti della vita reale. NGSI è lo standard adottato da FIWARE per la gestione e lo scambio delle informazioni riguardanti le risorse.

L'IoT Discovery GE è il modulo responsabile della gestione delle informazioni sulla availability delle Things. Le funzionalità che mette a disposizione permettono di registrare le informazioni riguardanti le thing e gli attributi relativi ad essa. Queste informazioni possono essere scoperte tramite operazioni di discovery oppure tramite il meccanismo di subscription/notify. L'architettura di questo GE è basata su un repository centralizzato in cui vengono memorizzate le informazioni delle risorse registrate.

L'obiettivo di questa tesi è quello di progettare e implementare una nuova versione dell'Iot Discovery Generic Enabler che segua un approccio distribuito e non più centralizzato. La soluzione proposta dovrà poter interagire con gli altri GE rispettando le Open Specification con cui il GE è fornito. Un altro aspetto su cui si è concentrato il lavoro è stata l'introduzione di nuove funzionalità in un'ottica di ottimizzazione del processo, in caso di operazioni di Query ricevute dal livello applicativo o dall'IoT Broker GE. All'IoT Discovery potrà essere richiesto di filtrare le risorse sulla base di range-query da applicare sui metadata associati ad esse prima di restituire l'identificativo e l'indirizzo a chi ne ha fatto richiesta.

Il lavoro di questa tesi si è sviluppato in due fasi. Nella prima parte si è svolto un lavoro di analisi dell'implementazione originale dell'IoT Discovery e dell'implementazione del servizio di discovery distribuito basato su LORM. In secondo luogo è stata progettata e realizzata la nuova implementazione, validandola e testandola verificandone il corretto funzionamento anche all'interno dell'ecosistema FIWARE.

Capitolo 2

Stato dell'Arte

L'*Internet of Things (IoT)* si propone di connettere, in una struttura Internet-like, un numero molto elevato di dispositivi, denominati *Smart Object* o *Smart Thing*, i quali forniscono e/o ricevono un servizio. La connessione di questi oggetti permette la comunicazione e lo scambio di informazioni, abilitando una nuova forma di interazione tra *Things* e persone. Gli Smart Objects sono tipicamente dotati di microcontrollore, interfaccia radio per la comunicazione, sensori e/o attuatori. Questi device sono vincolati da limitate capacità di calcolo, di memoria e di consumo energetico, in quanto sono stati concepiti per operare alimentati a batteria.

Un fattore cruciale per facilitare l'installazione e la configurazione degli Smart Object e per rendere possibile l'esecuzione di applicazioni robuste che lavorino con essi, è l'essenzialità di avere un meccanismo che minimizzi e idealmente elimini del tutto, la necessità di un intervento esterno da parte dell'uomo per configurare e gestire i device. Questo fattore permette di aumentare sensibilmente la scalabilità e realizzabilità del sistema (altissimo numero di oggetti da configurare), riducendo la possibilità di errore nella configurazione e favorendo la mobilità dei dispositivi.

Il protocollo IP è stato ampiamente previsto come il vero fattore abilitante dell'IoT, in quanto permette di avere piena interoperabilità tra oggetti eterogenei. Come parte del processo di standardizzazione che si sta svolgendo, nuovi protocolli *low-power* sono in fase di definizione da parte delle organizzazioni internazionali quali IETF e IEEE. A livello applicazione il protocollo *CoAP* [8] è stato progettato per portare il paradigma *REST* (REpresentational State Transfer) nell'Internet of Things. Il paradigma REST è finalizzato a prolungare la durata del software e l'evoluzione indipendente. Entrambi gli aspetti sono fondamentali per l'IoT e le applicazioni M2M sviluppate su smart object, i quali sono destinati a rimanere operativi per lunghi periodi (anni).

Oltre ai protocolli di livello applicazione, devono essere definiti anche dei meccanismi adeguati per il discovery di servizi e risorse. CoAP definisce con *Service Discovery* (SD) le procedure tramite le quali un client viene a conoscere gli endpoint esposti da un server. Il meccanismo di discovery dei servizi permette ai dispositivi e ai servizi di una rete di diventare automaticamente consapevoli dell'identità (URI) e delle funzionalità offerte (servizi) da ogni nodo della rete.

> RFC "A server is discovered by a client (knowing or) learning a URI that references a resource in the namespace of the server."

Il *Resource Discovery* è relativo alla scoperta delle risorse offerte da un endpoint. Le applicazioni M2M si affidano a questa funzione per mantenere le applicazioni resilienti ai cambiamenti, senza l'intervento esterno da parte dell'uomo. Una *Resource Directory* (RD) invece è un elemento della rete che ospita gli identificativi e le risorse offerte da server (es. sensori, attuatori...) all'interno della rete e che permette a dei client (es. altri sensori, applicativi di gestione...) di effettuare ricerche (lookup) su queste risorse [5]. Il Service Discovery può essere suddiviso in due classi di protocolli:

- Centralizzato - prevede una o più resource directory contenenti le liste di tutti i server e servizi offerti presenti nell'intera rete. I client utilizzano la ben nota RD come intermediario per scoprire i servizi.
- Distribuito - in questo approccio i dispositivi della rete scoprono i servizi interagendo tra loro senza utilizzare intermediari

Le funzionalità principali per un protocollo di Service Discovery possono essere individuate in questi processi:

- *Publication* - è il processo in cui i server annunciano i servizi che loro offrono (nome, tipo, identificativo, nome del dominio, attributi). Questo processo è utilizzato dai protocolli di tipo distribuito.
- *Registration* - è il processo in cui le descrizioni dei servizi offerti dai server vengono registrati nella RD, in modo da renderle disponibili per il discovery.
  - registrazione stateless - la directory ascolta gli advertisement (messaggi di pubblicazione) dei server e popola la propria lista dei servizi.
  - registrazione stateful - è necessaria una registrazione esplicita da parte del server verso la RD.
- *Discovery e Resolution* con discovery si intende il browsing di un dominio per localizzare le istanze di un determinato servizio. Con resolution si

intende la traduzione del'istanza del servizio scoperto in un indirizzo (o hostname) dell'endpoint associato.

Il meccanismo di SD applicato su sistemi di grandi dimensioni, dovrà soddisfare caratteristiche di scalabilità, fault tolerance e capacità di self-configuring. Nel contesto delle constrained network però per il SD sono necessari ulteriori requisiti

Basso overhead per i messaggi di controllo per ridurre il consumo di energia

Bassi requisiti di calcolo e di memoria , per poter eseguire il meccanismo di discovery anche sui nodi con risorse più limitate

Interoperatività con applicazioni web e reti basate su IP per permettere ai constrained device di comunicare globalmente

Robustezza della fornitura del servizio rispetto alla disponibilità imprevedibile dei dispositivi (connettività wireless e alimentazione a batteria)

Attualmente si hanno 2 protocolli di livello applicazione adottati per realizzare meccanismi di discovery dei servizi resource aware per reti con constrained device [9].

CoAP (Constrained Application Protocol) è un protocollo di trasferimento web RESTful. E' stato progettato per le comunicazioni M2M in reti vincolate, possiede quindi header ridotti per diminuire l'overhead e una complessità minore per il parsing delle query. CoAP definisce schemi URI per identificare e localizzare risorse CoAP e supporta la tipizzazione dei contenuti. Un *service* è definito dalla tupla protocol, host port che rappresenta l'entrypoint di un server CoAP. Una *resource* invece è una qualsiasi feature che un endpoint mette a disposizione per essere usata con metodi CoAP. Ogni risorsa è identificata da un URI ed è caratterizzata da un resource type. Gli endpoint e le risorse CoAP sono "scoperte" con i metodi di resource discovery di CoAP. Lo scopo di questi metodi è fornire gli URI e gli attributi che descrivono le risorse messe a disposizione da un endpoint. Ne sono stati proposti due:

- *CoAP resource discovery* E' un approccio distribuito, dove un device esegue una query diretta verso un altro dispositivo per scoprirne le risorse. Se il client conosce l'indirizzo del target esegue una richiesta (unicast) GET al "well-known" URI del server ("*GET /.well-known/core*"). Il server risponderà con gli URI di tutte le sue risorse "scopribili". Se il client non conosce l'indirizzo del target può

eseguire una richiesta di discovery multicast con uno scope limitato. In questo modo, con una richiesta al well-known URI, scopre gli endpoint (i server coap) e le risorse da loro offerte. Le richieste dei client possono anche riguardare specifici tipi di risorsa o appartenenza ad un dominio nella forma "parameter=value" ("*GET /.well-known/core?rt=tempC&d=mydomain.it*")

- *CoAP Resource Directory (RD)* E' un approccio centralizzato. I client eseguono tutte le query per il resource discovery su un'entità ben nota (Directory) la quale memorizza le descrizioni delle risorse possedute dai server all'interno della rete. La registrazione delle risorse nella directory avviene quando un server esegue una richiesta POST al path della RD (che deve essere conosciuto). Se ha successo, la RD restituisce al server il path dove sono state salvate le risorse. Devono essere disponibili anche funzionalità aggiuntive per gestire e mantenere coerente la RD (update, validation, remove) Un client per scoprire quali server implementano una certa risorsa eseguirà una richiesta GET alla RD utilizzando una query specifica. La RD restituirà direttamente una lista di risorse CoAP con le quali il client potrà interagire direttamente

DNS-SD è un protocollo internet per il Service Discovery già esistente, basato sulla tecnologia DNS. Comporta un overhead limitato in termini di risorse memoria. E' un meccanismo per il quale vengono sfruttate le interfacce di programmazione standard di DNS, il formato dei pacchetti e i server per il browsing di servizi in una rete. In particolare DNS-SD definisce come organizzare e assegnare i nomi ai record DNS con lo scopo di facilitare il service discovery in un subdomain, senza andare a modificare la struttura del protocollo DNS. Un server DNS-SD conterrà una lista di servizi definiti secondo una Service Instance Name con formato (Instance).(ServiceType).(Domain). I Service Type sono prestabiliti e seguono uno standard definito dalla SDO e identificano un application protocollo (service) e altre funzionalità aggiuntive (service subtype) es. *light._sub._coap._udp.*

In [5] viene proposta una soluzione per il *service* and *resource discovery* che presenta le caratteristiche di scalabilità e self-configuring. La soluzione proposta si basa sulla tecnologia P2P per fornire un'infrastruttura di discovery su larga scala, la quale prevede che gli *IoT Gateway* siano i peer di *due overlay network strutturate.*

- DLS fornisce un location service per salvare e recuperare informazioni per accedere a servizi e risorse

- DGT è basata su uno schema costruito utilizzando la locazione geografica dei nodi partecipanti. Attraverso questa overlay, uno nodo può recuperare informazioni sui nodi situati vicini geograficamente ad una posizione specifica.

Ogni *IoT Gateway* gestisce la propria rete (local scale) raccogliendo le informazioni sulle risorse fornite dai server presenti ed è responsabile dell'implementazione del comportamento richiesto dall'architettura SD. Ogni rete locale si autoconfigura attraverso meccanismi Zero-conf.

L'**IoT Gateway** è un elemento della rete che interagisce a livello applicazione con altri nodi IoT utilizzando CoAP. Permette di coordinare e rendere pienamente interagibili device eterogenei che possono essere anche geograficamente distanti. Per definizione può agire da

- *CoAP origin server* cioè un server CoAP sul quale risiedono delle risorse.
- *CoAP proxy* cioè un endpoint CoAP che implementa il server side che sia il client side inoltrando richieste ad un altro server e rimandando indietro la risposta ricevuta
- *HTTP-to-CoAP proxy* traducendo richieste HTTP in richieste CoAP (e viceversa) e comportandosi in seguito come un CoAP proxy.

L'architettura P2P fornisce caratteristiche desiderabili per sistemi di grandi dimensioni, infatti è scalabile (al crescere del numero di peer, cresce la potenza di calcolo e di memorizzazione), è fault-tolerant e si auto configura. La rete P2P organizza i nodi partecipati in overlay network costruite sopra la rete esistente. Per realizzare una overlay network strutturata si utilizza una DHT basata su un algoritmo di hashing consistente il quale garantisce che il routing delle richieste abbia un numero limitato e deterministico di hop per il completamento. I sistemi P2P strutturati con DHT, offrono una elevata robustezza ai guasti e distribuiscono responsabilità e carico in modo sistematico per mezzo di una struttura topologica a strati [10]. Dato che le DHT offrono funzionalità di ricerca di identificatori, possono costituire una base altamente scalabile e robusta per DS globali. In [11] è presentata un'implementazione di un DS DHT-based chiamato OIDA, basato su Bambù DHT, testato su circa 350 nodi distribuiti globalmente della piattaforma PlanetLab. In [12], i risultati della simulazione (utilizzando Pastry DHT) hanno mostrato la fattibilità di reti P2P costituite anche da 20000 nodi.

2.1 LORM

LORM-DHT (Low-Overhead Range-query Multi-attribute) [7] è un servizio di resource discovery che si basa su una rete DHT a grafo composto in grado di distribuire tra i nodi le informazioni sulle risorse. LORM è costruito su una singola DHT chiamata Cycloid [13]. LORM organizza ogni nodo per essere responsabile per le informazioni delle risorse che possiedono uno specifico attributo il cui valore è all'interno di un certo range. Questo è possibile sfruttando la struttura a grafo composto di Cycloid la quale connette in circolo dei cluster. In altre parole le resource information di un attributo sono contenute all'interno di un cluster e sono distribuite sui vari nodi appartenenti ad esso sulla base del valore che l'attributo assume. LORM raggiunge grande efficienza attraverso una distribuzione bilanciata dell'overhead di mantenimento e le operazioni di resource discovery. Inoltre, è in grado di gestire un alto numero di risorse e sistemi con caratteristiche dinamiche (alto numero di ingressi o cancellazione di nodi).

In Cycloid, ad ogni oggetto o nodo è assegnato un ID che è l'hash del nome dell'oggetto o dell'indirizzo IP del nodo. La rete fornisce due operazioni principali: *inserimento(chiave, oggetto)* e *ricerca(chiave)* che, rispettivamente, memorizzano un oggetto in un nodo o recuperano l'oggetto. Ogni messaggio relativo a una operazione è trasmesso da nodo a nodo in base all'algoritmo di instradamento, fino a raggiungere il nodo che possiede l'oggetto o su cui dovrà essere inserito. Ogni nodo mantiene una tabella di routing che registra i suoi vicini all'interno del cluster.

Cycloid è una overlay-network a dimensione costante con $n = d \cdot 2^d$ nodi, dove d è la dimensione della DHT [13]. Essa raggiunge una complessità in termini di tempo $O(d)$ per richieste di lookup utilizzando $O(1)$ vicini per nodo. L'**ID** di ogni nodo o oggetto in Cycloid è rappresentato da due indici $< k, a_{d-1}, a_{d-2}, \ldots, a_0 >$, dove k è chiamato *indice ciclico* e $a_{d-1}, a_{d-2}, \ldots, a_0$ è chiamato *indice cubico*. L'indice ciclico è un intero compreso tra $[0, d-1]$ mentre l'indice cubico è un numero binario compreso tra $[0, 2^d - 1]$ e identifica il cluster di appartenenza. I nodi con lo stesso indice cubico sono ordinati secondo k all'interno del loro cluster. I nodi con k maggiore all'interno del cluster assumono il ruolo di nodo primario. Tutti i cluster sono connessi secondo il loro indice cubico. Nella routing table di ogni nodo vengono memorizzati gli indirizzi dei nodi vicini: il predecessore, il successore e i nodi primari del cluster precendente e successivo rispetto a quello a cui appartiene il nodo.

L'idea alla base di LORM è di rendere ogni cluster responsabile per le informazioni di un solo attributo e distribuire i dati tra i nodi basandosi sul valore o sulla sua descrizione. LORM assegna ad ogni risorsa un Cycloid ID

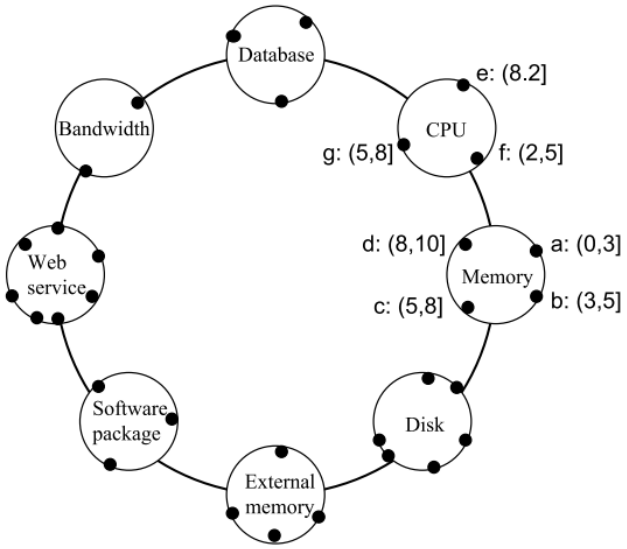


Figura 2.1: Struttura di LORM

$< k, a_{d-1}, a_{d-2}, \ldots, a_0 >$. Con l'indice cubico $a_{d-1}, a_{d-2}, \ldots, a_0$ viene identificato l'attributo a mentre l'indice ciclico rappresenta l valore dell'attributo a. Quindi l'indice cubico servirà a differenziare i cluster mentre l'indice ciclico stabilirà la posizione del nodo all'interno del cluster. Le informazioni verranno distribuite come in Figura 2.1.

Per generare gli indici cubici viene utilizzata una funzione hash consistente come SHA1 per garantire uniformità della rete ed evitare collisioni. Per poter implementare invece le range-query l'indice ciclico dovrà utilizzare una *funzione hash locality preserving*.

Definizione 1. La funzione hash H è una funzione hash locality preserving se essa possiede la seguente proprietà: $H_{v_i} < H_{v_j}$ se e solo se $v_i < v_j$ e se un intervallo $[v_i, v_j]$ è diviso in $[v_i, v_k]$ e $[v_k, v_j]$, il corrispondente intervallo $[H_{v_i}, H_{v_j}]$ dev'essere diviso in $[H_{v_i}, H_{v_k}]$ e $[H_{v_k}, H_{v_j}]$.

Si assume che ogni risorsa sia rappresentata da un valore π limitata da un valore minimo π_{min} e un valore massimo π_{max}. Dato che la lunghezza dell'ID di un cluster è $d-1$, viene definita la funzione hash locality preserving

$$H(\pi) = (\pi - \pi_{min}) \cdot \frac{(d-1)}{(\pi_{max} - \pi_{min})}$$

dove π è il valore dell'attributo a nel range di valori $[\pi_{min}, \pi_{max}]$. Come risultato ogni valore dell'attributo a di una risorsa avrà come hash H un valore compreso

in $[0, d-1]$. Nel caso in cui il valore di un attributo sia una stringa, l'indice ciclico sarà calcolato con una funzione hash consistente come SHA1.

Come in tutte le DHT, le due operazioni più importanti sono l'inserimento e la ricerca. Per quanto riguarda la prima operazione, una risorsa può inserire un dato attraverso l'interfaccia *Insert(rescID, rescInfo)* dove $rescInfo =< a, d_{\pi_a}, ip\_addr(i) >$ racchiude tutte le informazioni sulla risorsa (a = attributo, d_{π_a} = valore inserito, $ip\_addr(i)$ = indirizzo IP di chi fornisce il dato), mentre il rescID è praticamente il Cycloid ID. LORM per inserire il dato quindi cercherà prima di raggiungere un nodo del cluster che gestisce l'attributo del dato inserito, poi inserirà l'informazione nella posizione corretta data dall'hash function locality preserving. Per l'operazione di ricerca, il nodo richiedente invia Lookup(rescID, rescInfo), dove il campo *ip_addr* di rescInfo rimane vuoto. Questo verrà riempito con operazioni di join sui nodi che possiedono i dati di interesse. Per le range query, la richiesta verrà inviata da nodo a nodo fino a quando non cade sull'ultimo nodo che gestisce l'estremo superiore del range.

Capitolo 3

FIWARE

3.1 Introduzione a FIWARE

FIWARE (o FI-WARE) è una piattaforma di servizi software, pensata per lo sviluppo e la distribuzione di applicazioni per il *Future Internet*. Il programma che ha portato alla creazione di FIWARE, è un'iniziativa di innovazione finanziata dalla Commissione Europea e guidata dai principali attori industriali ICT e si prefigge la creazione e la messa in produzione di una piattaforma tecnologica che mira ad aumentare la competitività globale dell'economia europea nel campo ICT.

Questa piattaforma, attraverso un'infrastruttura Cloud, offre agli sviluppatori un insieme piuttosto semplice ma potente di *servizi* Open Source, che facilitano e rendono efficace nei costi lo sviluppo di applicazioni intelligenti. Le tecnologie sviluppate su FIWARE includono soluzioni per Big Data, Internet of Things, Sicurezza, Media e molto altro.

3.2 Struttura di FIWARE

FIWARE si basa su elementi chiamati *Generic Enablers (GE)*. Essi sono i blocchi di base della piattaforma e sono costituiti da un gruppo di componenti che insieme supportano un set concreto di funzionalità e che forniscono un insieme di API e interfacce tra loro interoperabili. FIWARE mette a disposizione un catalogo di Generic Enablers, dove all'implementazione di riferimento delle librerie e delle interfacce vengono associate le *Open Specification* del GE. Queste specifiche contengono informazioni rilevanti per gli sviluppatori che utilizzano le implementazioni relative oppure realizzano prodotti conformi che possono funzionare da implementazioni alternative al GE messo a disposizione da FIWARE. Le *Open Specification* di un Generic Enabler tipicamente includono:

Descrizione del comportamento, delle funzionalità offerte e dell'ambito di utilizzo e casi d'uso del GE.

Signature, cioè l'interfaccia di tipo *RESTful* e il comportamento delle operazioni collegate alle API che il Generic Enablers deve esportare.

Descrizione dei protocolli che il Generic Enablers deve supportare per garantire l'interoperabilità con altri GE o prodotti conformi di terze parti.

Un insieme di definizioni, terminologia e abbreviazioni utili a rendere chiara la comprensione delle specifiche.

I vari componenti come detto sono organizzati in una ricca libreria, il *FIWARE Catalogue* [14], dove possono essere trovare le implementazioni di riferimento di ogni componente, permettendo così agli sviluppatori di rendere concrete le funzionalità proposte come la connessione all'*Internet of Things* o all'analisi dei *Big Data*, e rendere più semplice la loro implementazione. I vari GE proposti possono essere combinati tra loro in modo da creare differenti tipi di prodotti o servizi:

Prodotti conformi alla Piattaforma FIWARE: un prodotto che implementa totalmente o in parte un *GE FIWARE* o una composizione di più GE FIWARE, cioè implementa dei servizi FIWARE. Quindi differenti prodotti conformi possono implementare lo stesso GE.

Instanza FIWARE: è il risultato dell'integrazione di un numero di prodotti conformi alla Piattaforma FIWARE. Esso comprende una serie di GE FIWARE e il supporto ad un numero di servizi FIWARE. La fornitura di un *Infrastructure as a Service* (IaaS) o di un *Context/Data Management Services* sono esempi di servizi che una particolare *FIWARE Instances* può supportare. Notare che un'istanza FIWARE può consistere non solo nell'integrazione di prodotti conformi alla piattaforma FIWARE, ma la loro integrazione può essere fatta anche con con altri prodotti permettendo una differenziazione sul mercato (es. integrazione in un proprio sistema operativo).

Future Internet Application: un'applicazione basata sulle API definite come parti di una Open Specification relativa ad un GE. Queste applicazioni possono essere portabili attraverso differenti *FIWARE Instances* che implementano i GE su cui l'applicazione si basa.

Nella catena dei valori complessiva prevista nell'ecosistema FIWARE, possono essere individuati ruoli differenti, classificabili in:

- *GE Provider*, cioè chiunque implementa un Generic Enabler.

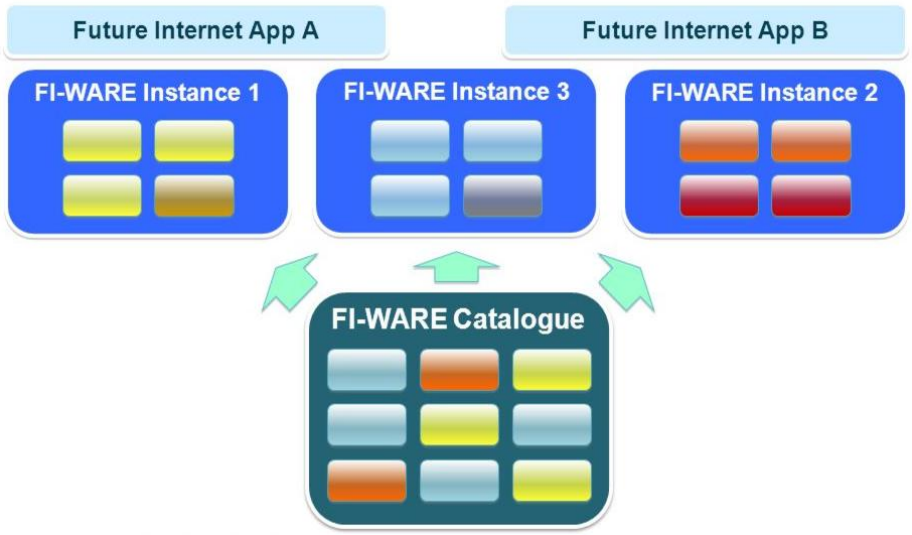


Figura 3.1: esempi di FIWARE Instances

- *Instance Provider*, cioè una compagnia o un'organizzazione che sviluppa e gestice un'istanza di una piattaforma, costruendo un modello di *business* intorno a questa particolare istanza.
- *Application/Service Provider*, cioè qualsiasi compagnia od organizzazione che sviluppa applicazioni per il Future Internet oppure servizi basati sulle API messe a disposizione da un GE, distribuendo queste applicazioni/servizi sopra una istanza FIWARE

3.3 Internet of Things in FIWARE

Tra i vari moduli che FIWARE mette a disposizione, è stato scelto come oggetto di studio e di lavoro il capitolo riguardante l'**Internet of Things** il quale fornisce una serie di Generic Enablers che consentono alle *Things* di diventare risorse di contesto

- *available* - disponibili
- *searchable* - ricercabili
- *accessible* - raggiungibili
- *usable* - usabili

favorendo l'interazione tra oggetti della vita reale e applicazioni basate sulla piattaforma FIWARE.

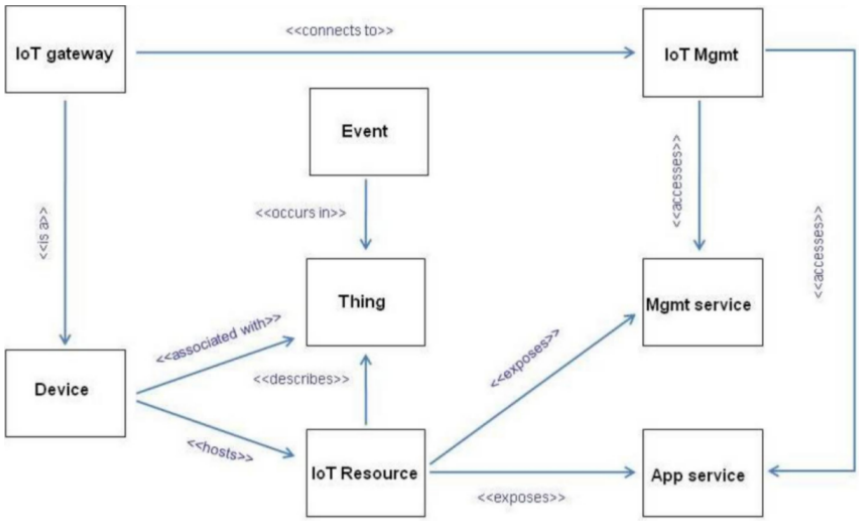


Figura 3.2: Concetti IoT

3.3.1 Concetti IoT

Nell'esposizione delle Open Specification riguardanti i GE dell'Internet of Things vengono definiti vari concetti, i quali rappresentano gli attori che prendono parte al contesto IoT (Fig. 3.2). Essi possono essere divisi in differenti livelli di astrazione:

Device Level, un'unità hardware con la capacità di eseguire misurazioni o attuare un comando. I dati prodotti a questo livello sono dati grezzi. In questo livello sono compresi i dispositivi che fungono da *IoT Gateway*, cioè quei dispositivi che forniscono oltre ad eventuali funzionalità di misurazione e attuazione, anche funzionalità di inter-networking e di conversione dei protocolli tra i dispositivi ad esso collegati e il backend IoT.

IoT Resource Level, una risorsa IoT è un elemento computazionale che fornisce l'accesso ai device di misurazione e attuazione. Le risorse IoT sono descritte attraverso l'utilizzo di un modello delle informazioni. I dati in questo livello consistono oltre che ai dati misurati, anche alle informazioni di contesto come il tipo dei dati forniti, un timestamp, l'accuratezza della misurazione e il sensore che fornisce la misurazione.

Thing Level, una Thing può essere un qualsiasi oggetto, persona o luogo nel mondo reale. Le Thing sono rappresentate come oggetti virtuali che possiedono un identificatore (entity Id), un tipo e diversi attributi. I dati in questo livello consistono in una descrizione delle Thing e dei loro attributi.

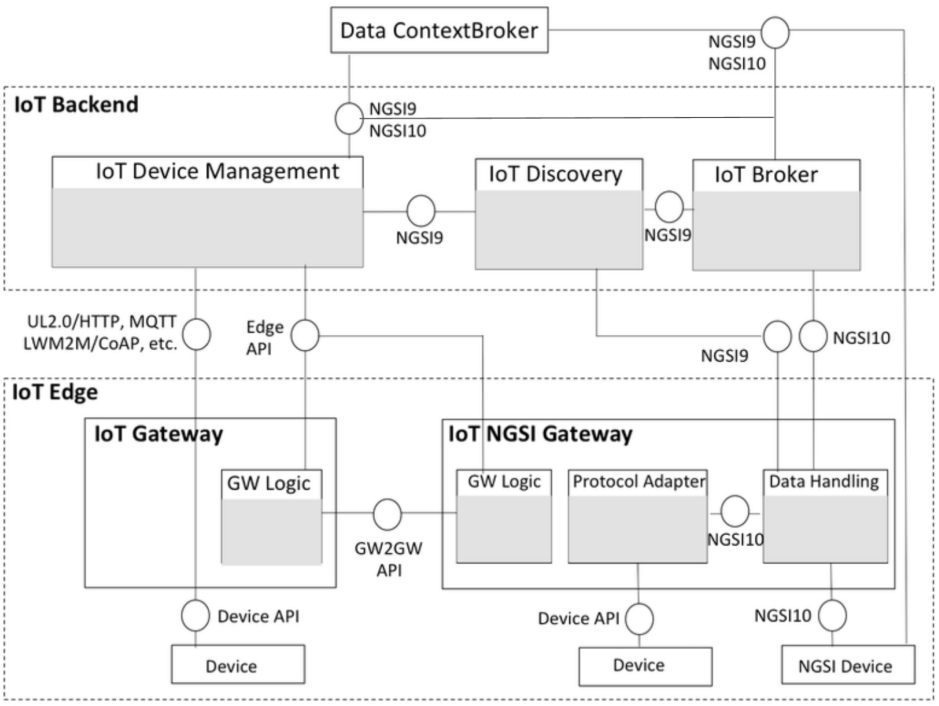


Figura 3.3: Architettura completa dell'IoT in FIWARE

Application Level, è composto dai seguenti elementi

- *Management Service* è la caratteristica della risorsa IoT di fornire un accesso software ai dati leggibili o scrivibili relativi al funzionamento del device.
- *Application Service* è la caratteristica della risorsa IoT di fornire un accesso software ai dati leggibili o scrivibili relativi alla Thing a cui è associato il device che possiede la risorsa. L'application service scambia dati applicazione con altri device incluso l'IoT Gataway e/o l'IoT Backend.
- *Event* il quale può essere definito come un'attività che si verifica in un dispositivo, gataway, IoT Backend.

3.4 Architettura dei Generic Enabler IoT

Lo sviluppo dell'architettura IoT di FIWARE varia da semplici scenari in cui sono presenti pochi device connessi, i quali utilizzano protocolli di comunicazione IoT standard, a scenari molto più complessi in cui vengono connessi un grande numero di IoT Nodes e IoT Gataway attraverso più IoT network fornendo una

composizione avanzata e funzionalità di discovery (Fig. 3.3) [15]. I Generic Enablers IoT sono divisi in due domini differenti:

IoT Backend: questo dominio comprende un insieme di funzionalità, risorse logiche e servizi ospitati in un datacenter Cloud. A livello superiore, è connesso verso applicazioni Future Internet o un'altra serie di GE relativi al settore del Data Context Broker di FIWARE. Deve quindi tradurre le risorse IoT in Context Entities NGSI. A livello inferiore invece è connesso agli elementi dell'IoT Edge, cioè elementi che costituiscono l'infrastruttura fisica dell'IoT.

IoT Edge: questo dominio è costituito da tutti gli elementi dell'infrastruttura IoT che sono necessari a connettere i dispositivi fisici alle applicazioni FIWARE. Tipicamente esso comprende gli IoT end-node, IotT Gateway e le IoT Network. Negli scenari più semplici i device che espongono protocolli di comunicazione conosciuti possono comunicare direttamente con il GE *IoT Device Management.* I GE dell'IoT Edge e le relative API aiutano nell'integrazione di nuovi tipi di device e di gateway.

Da un punto di vista delle funzionalità offerte, i GE dell'IoT Backend svolgono le seguenti funzioni:

Fornisce le NGSI Context Entities relative alle Things. Questo significa che viene creata una NGSI Context Entity per ogni risorsa IoT. Questa funzione viene tipicamente svolta dall'IoT Device Management, il quale svolge quindi la funzione di *Context Producer* per gli attributi di entità relativi a risorse IoT di misurazione, mentre svolge funzioni di *Context Consumer* per gli attributi di entità relativi a risorse IoT di attuazione.

Adattamento dei protocolli dell'IoT Southbound: gestisce la traduzione dei protocolli propri delle Things verso il protocollo interno a FIWARE, *OMA NGSI.* Questa funzione è svolta dall'IoT Device Management.

IoT Device Composition e Discovery: questa funzione è svolta dall'IoT Broken in combinazione con l'IoT Discovery.

Per quanto riguarda invece l'IoT Edge GE sono state previste e implementate da FIWARE le seguenti funzionalità:

Adattamento dei protocolli dell'IoT Southbound: è analogo alle funzioni svolte nell'IoT Backend. In questo dominio è svolta dal Protocol Adapter GE

Complex NGSI Event Processing: utile per ridurre il traffico di rete scambiato tra elementi dell'IoT Edge e l'infrastruttura Cloud. Queste funzioni sono realizzate dal Data Handling GE su un device di tipo Gateway.

Gateway Logic: Questa funzione è prevista per gestire le API Gateway-to-Gateway e le API di configurazione dell'IoT Edge (ancora in fase di definizione)

IoT end-node Configuration: alcuni nodi Ip-capable possono essere gestiti e controllati direttamente. Ciò significa che essi stessi implementano una IoT Edge configuration API al livello IoT node. In questo caso l'interfaccia principale prevista, data la natura constrained dei nodi IoT, è LWM2M.

IoT Networks Configuration: in scenari complessi i nodi IoT possono essere connessi attraverso varie reti eterogenee (cellular, 6LowPAN, IP, ecc..). Questo piano di controllo può prevedere l'interazione con delle network APIs.

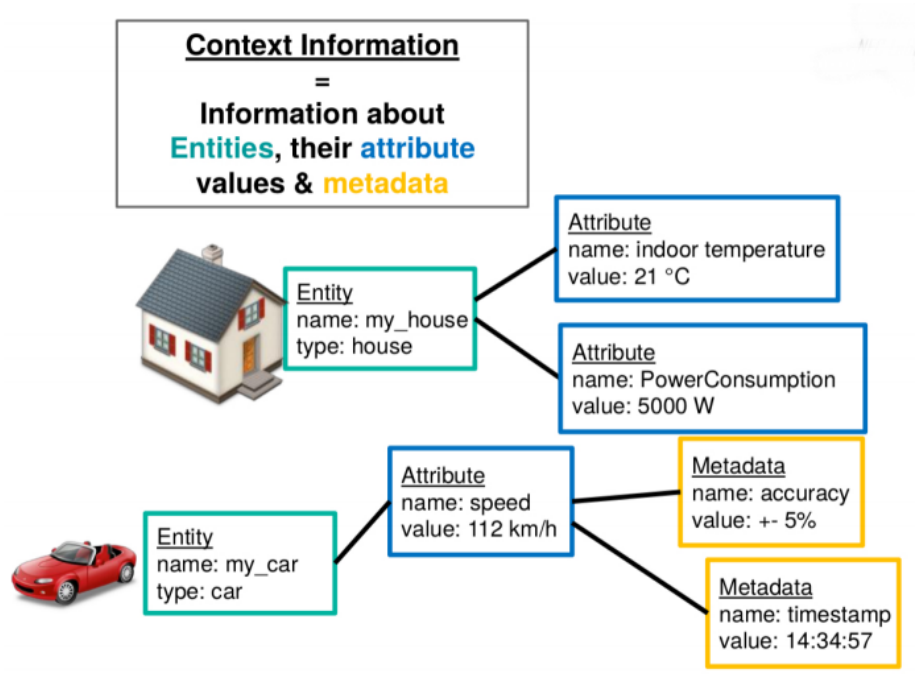


Figura 3.4: Context Information di NGSI

3.5 OMA NGSI - Context Management

Tutte le informazioni riguardo Things e/o IoT Resources sono gestite tramite lo standard OMA NGSI (*Next Generation Service Interface*), che descrive questi oggetti come **Context Entities** (Fig. 3.4). Lo standard OMA NGSI *Context Management* fornisce delle interfaccie per gestire e scambiare *Context Information* relative a Context Entities.

Una Context Entity è intesa come un'entità che ha un proprio stato.

Le Context Informations sono i valori assunti dagli attributi di una Context Entity. Le applicazioni definite context-awareness utilizzeranno le context informations come supporto alle loro attività. Le Context Information sono qualsiasi informazione volatile o persistente che descrivono lo stato di una Context Entity.

L'adozione dello standard OMA NGSI in FIWARE permette la gestione della configurazione e dei dati associati a Things presenti nel mondo reale. Consente questo ad un livello di astrazione che permette il superamento della complessità di gestione delle connessioni con gateway e dispositivi. L'aggiornamento dello stato e la configurazione di questi oggetti fisici si tradurrà in un aggiornamento delle Context Information.

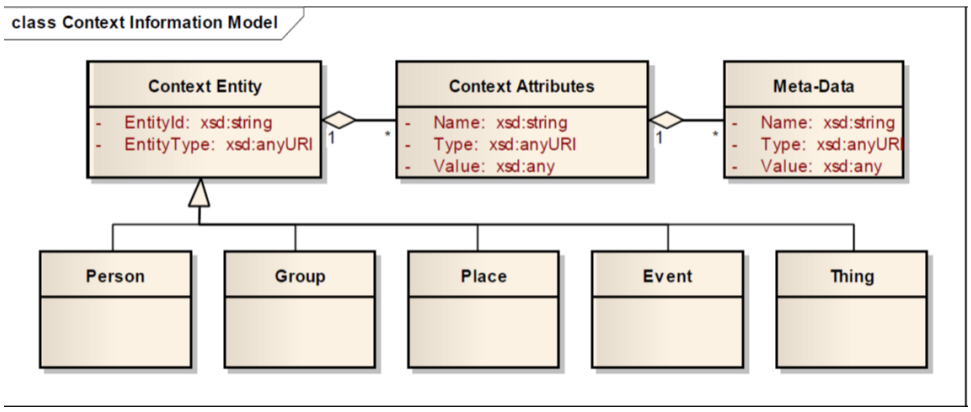


Figura 3.5: Modello delle informazioni di contexto in NGSI

Le librerie dell'NGSI Context Management forniscono delle interfacce che hanno lo scopo di rendere possibile la gestione delle Context Information relative alle Context Entities e la possibilità di accedere tramite query (o un sistema di subscribe/notify) alle Context Informations relative alle Context Entities **available.** Lo standard NGSI specifica due interfacce:

- **NGSI-9** con lo scopo di scambiare informazioni riguardanti la *availability* delle Context Entities e delle Context Informations.
- **NGSI-10** con lo scopo di scambiare le Context Informations stesse.

Context Information Model

Le Context Entities sono descritte dal *Context Information Model*, il quale dettaglia come le Context Information sono strutturate e associate alle Context Entities in modo da descrivere il loro stato. In questo modello le Context Information sono organizzate come *Context Elements*, che contengono un insieme di *Context Attributes* e dei *Metadata* associati (Fig. 3.5).

- **EntityId** identifica una Context Entity
- **EntityType** indica il tipo di ContextEntity. E' necessario se l'EntityId non contiene informazioni sul tipo, oppure se l'EntityId è univoco solo all'interno del proprio tipo di entità.
- **ContextAttributes** un attributo è definito con un insieme di informazioni (nome, tipo e valore) e un insieme di **metadata**.

3.5.1 Operazioni NGSI-9

registerContext Questa operazione permette la registrazione delle Context Entities, dei nomi dei loro attributi e della loro disponibilità. Permette anche l'aggiornamento delle informazioni delle entità già registrate.

discoverContextAvailability Questa operazione è sincrona e permette di scoprire (potenzialmente) un set di ContextEntities, il tipo a loro associato e le relative Context Informations (attributi e metadati). La verifica avviene solo sulle entità registrate nel registro e non sulle reali fonti delle Context Information. In altri termini questa operazione restituisce una lista aggregata di Context Registration

subscribeContextAvailability permette (potenzialmente) il discovery asincrono di un set di ContextEntities, il tipo a loro associato e le relative Context Informations (attributi e metadati). Questo però non garantisce che le Context Informations relative ad una Context Entity del set siano attualmente disponibili. In altri termini permette di sottroscriversi alle notifiche sulla availability di una lista aggregata di Context Registration

notifyContextAvailability questa operazione è la risposta asincrona ad una subscription. Permette di ricevere la notifica di un potenziale set di Context Registration sottoscritte dal richiedente che implementa l'interfaccia di notifica.

updateContextAvailabilitySubscription permette di aggiornare una precedente operazione di subscription

unsubscribeContextAvailabilitySubscription permette di annullare una precedente operazione di subscription

3.5.2 Operazioni NGSI-10

queryContext questa operazione permette il recupero in modo sincrono di Context Information. Chi richiede un'operazione di queryContext deve specificare una lista di entity identifiers. Tali identificatori possono rappresentare entità uniche oppure pattern di entity identifiers. Un *entity identifier pattern* deve essere rappresentato come espressione regolare. Il richiedente può specificare anche una lista di attributi che devono essere recuperati con la query. Gli entity identifiers possono includere l'attributo type per specificare il tipo di target entities. Si assume che il richiedente sia a conoscenza dei context identifiers, attributi e type attraverso *precedenti operazioni di context entity discovery.*

updateContext questa operazione permette di aggiornare un insieme di Context Information, relativi attributi e metadata. Il comportamento di questa operazione cambia in base al tipo di azione di update che l'operazione andrà ad eseguire.

- update - aggiorna un context element
- append - aggiunge un context element
- delete - elimina un context element

subscribeContext questa operazione permette il recupero in modo asincrono di Context Information. Viene utilizzata per sottoscrivere Context Information. La sottoscrizione attiva le notifiche sulle Context Entities che soddisfano le NotifyCondition passate al momento dell'operazione di subscription.

notifyContext questa operazione è la risposta asincrona ad una subscription. Permette di ricevere la notifica riguardo Context Information sottoscritte dal richiedente che implementa l'interfaccia di notifica.

updateContextSubscription permette di aggiornare una precedente operazione di subscription

unsubscribeContextSubscription permette di annullare una precedente operazione di subscription

3.6 IoT Edge

Data Handling Generic Enabler

Questo Generic Enabler risponde alla necessità di filtraggio, aggregazione e merging in tempo reale, di dati provenienti da fonti diverse. Tipiche applicazioni che necessitano queste feature sono applicazioni che hanno tra i requisiti processare eventi in real-time. Il Data Handling GE gestisce data stream dai device IoT, i quali non possono essere continuamente online essendo dispositivi con vincoli energetici. Ciò richiede tipicamente delle comunicazioni limitate e ottimizzate con le risorse IoT. Per questo motivo è desiderabile avere un local storage per mantere una cache degli ultimi dati processati. Le principali interazioni di questo GE avvengono tramite lo standard NGSI, con il IoT Broker GE, il Configuration Management GE e con il Protocol Adapter GE

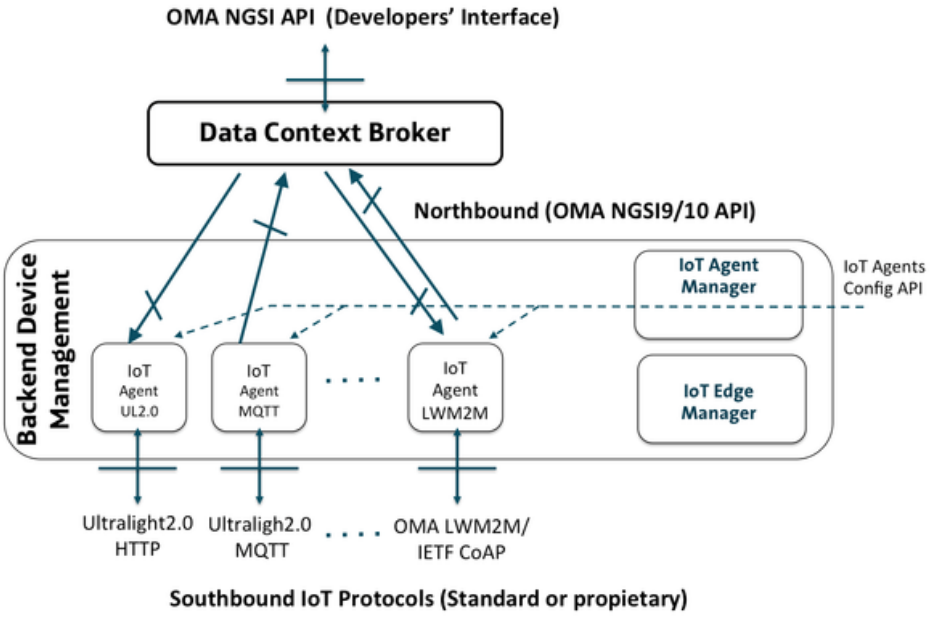


Figura 3.6: Device Management GE

Protocol Adapter Generic Enabler

Il Protocol Adapter GE ha a che fare con il traffico in ingresso e in uscita e con i messaggi tra l'IoT Gataway e i dispositivi registrati, da servire anche al Data Handling GE. Questo GE riceve dei protocolli specifici dei device e li traduce in API uniformate al protocollo interno (NGSI).

- Data Handling API
- Device Management API

Le API esposte gestiscono le capacità di leggere e scrivere dalle risorse, così come la gestione e la configurazione di servizi IoT specifici come il resource discovery (sia lookup che publication).

3.7 IoT Backend

3.7.1 IoT Device Management GE

Il Backend Device Management GE permette la connessione dei dispositivi fisici alla piattaforma FIWARE, adattando i vari standard e protocolli di comunicazione (anche proprietari) allo standard NGSI. Gestisce la connessione (northbound) verso un ContextBroker o l'IoT Discovery, per creare una NGSI

Context Entities per ogni dispositivo fisico ad esso connesso. Le Applicazioni FIWARE interagiscono esclusivamente con le NGSI Entities. Inoltre esso fornisce l'*IoT Edge Manager* che da la possibilità agli integratori IoT di configurare, manovrare e monitorare gli IoT node, gli IoT gateway e le reti IoT.

Questo Generic Enabler è composto da un insieme di moduli chiamati IoT Agent, dal gestore degli IoT Agents e dall'IoT Edge Manager (Fig. 3.6). L'IoT Agent Manager è un modulo opzionale che si interfaccia con tutti gli IoT Agent installati, attraverso le loro API di configurazione e amministrazione. Questo abilita un singolo punto di lancio, configurazione e monitoraggio per tutti gli IoT Agent presenti nell'ecosistema FIWARE.

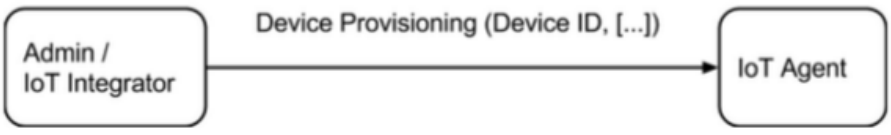
IoT Agent

Gli IoT Agents sono moduli software che gestiscono da lato southbound il protocollo specifico per il quale sono assegnati (es. LWM2M/CoAP, UL2.0/HTTP ....), mentre dal lato northbound gestiscono le interazioni OMA NGSI. La configurazione minima di un Device Management GE prevede almeno un IoT Agent.

Ogni Agent deve gestire la creazione di una NGSI Context Entities per ogni dispositivo IoT connesso. Esso agisce come *context producer* per quegli attributi relativi a capacità di sensing. Dovrà inoltre fornire API di configurazione/amministrazione dell'Agent. Infine ogni Agent può eseguire comandi di attivazione per quei device che possiedono attuatori, ciò è possibile nel caso in cui a livello superiore si esegue un update su specifici attributi (di tipo command) della Context Entities relativa al device.

Tra tutti gli IoT Agent in questa tesi è stato preso in considerazione per simulare il livello dei device **LWM2M/CoAP IoT Agent.** Questo IoT Agent (server) connette dei client (device) Lightweight M2M, i quali comunicano via CoAP agli altri Generic Enabler che supportano le interfaccie NGSI 9 e 10. Nell'attuale release, l'IoT Agent accetta scritture sugli attributi del device tramite scritture sulla corrispettiva Context Entities, ma non supporta ancora pienamente l'invio di comandi. Le principali interazioni che si possono avere con questo IoT Agent sono illustrate di seguito.

Device Provisioning

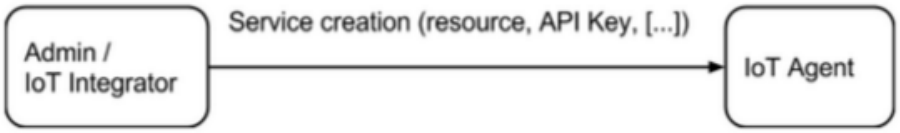


Il device provisioning è necessario affinchè il sistema riconosca il dispositivo e sia così possibile fare una mappatura appropriata in NGSI Context Elements. Può avvenire in due modi

- viene eseguito il provisioning del dispositivo nel sistema prima che venga connesso
- viene eseguito il provisioning di un servizio e il dispositivo viene assegnato ad esso

Nel primo caso il DeviceId, identificherà il dispositivo. Nel provisioning possono essere specificate: DeviceId, NGSI Entity Mapping, tipi di attributi previsti.

Service Creation



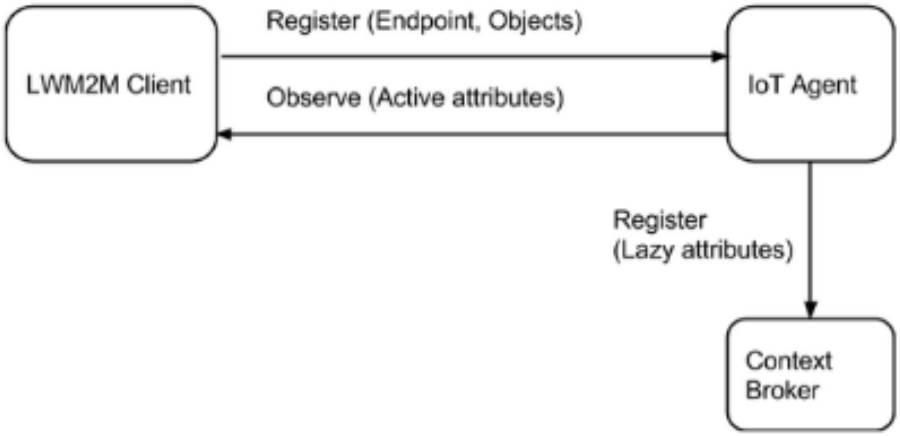
Nel caso in cui i device non siano specificati individualmente, dei servizi possono essere forniti nel suo complesso. Un servizio è identificato da una coppia di attributi (resource, APIkey) e può contenere all'incirca le stesse informazioni richieste per il provisioning di un dispositivo. Le risorse specificate corrispondono ad un endpoint (un server LWM2M) dove i client potranno inviare le loro richieste. Ogni volta che un dispositivo raggiunge l'endpoint specificato gli sarà assegnato il servizio appropriato in base all'enpoint.

Device Registration

Ogni device deve registrarsi al LWM2M server (l'IoT Agent in questo caso) prima di iniziare qualsiasi interazione. Durante la registrazione il LWM2M Client deve indicare al server le seguenti informazioni

- DeviceId (endpoint name)
- Supported Objects (una lista di collegamenti per ogni LWM2M object che può essere acceduto dal server)

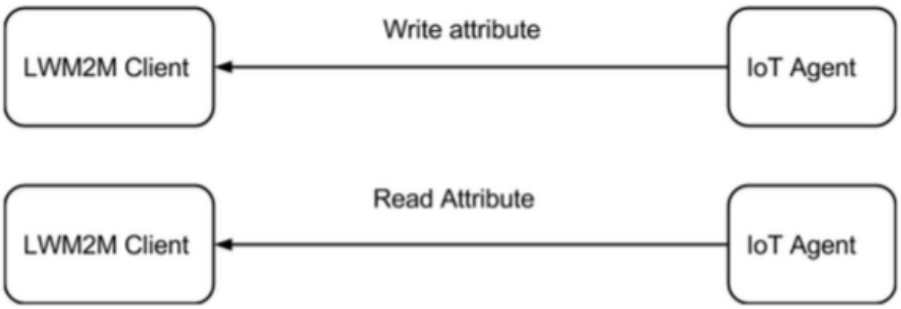
Il LWM2M server utilizza questi dati per recuperare le informazioni del device (se viene effettuato il provisioning diretto) oppure per assegnare il device ad un servizio. Basandosi sulle informazioni del device (o del servizio), il server deciderà quali attributi saranno



- attivi
- lazy
- command

Registrandosi esso stesso nell'IoT Discovery, come Context Provider per gli attributi di tipo lazy e command.

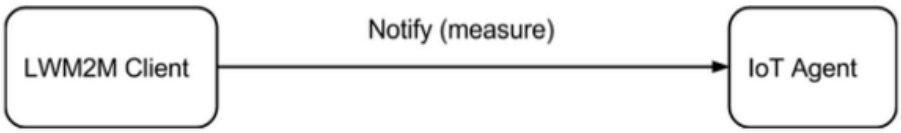
Device Lazy Observation



Queste operazioni sono per gli attributi del device che sono stati marcati come lazy. Le operazioni di read e update fatte dal Context Broker per la Context Entities associata, sono mappate in operazioni di read e write in LWM2M dall'interfaccia del Device Management.

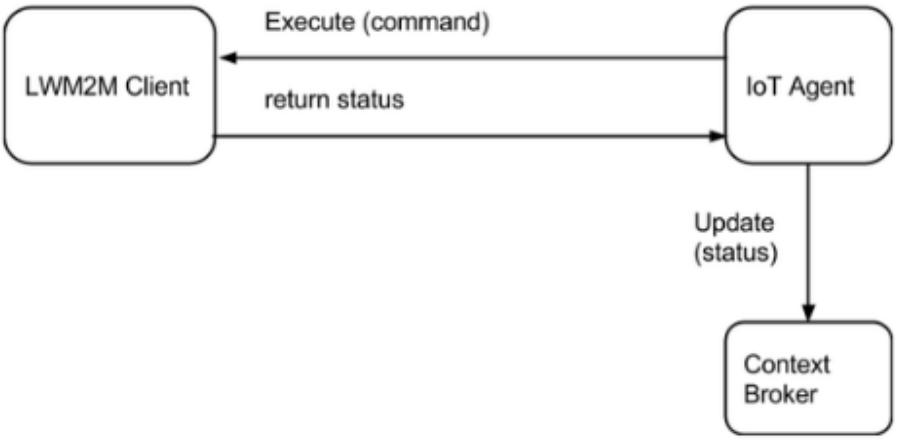
Device Active Observation

Per gli attributi del device che sono stati marcati come active al momento della registrazione viene eseguita dal server un'operazione di Observe, la quale prevede



successivamente che le misure dell'attributo vengano notificate al server da parte del LWM2M Client,

Device Command



I comandi vengono eseguiti come aggiornamenti con particolari valori degli attributi command nella Context Entity. Sono mappati in operazioni Execute dal server al LWM2M Client. Lo stato dell'esecuzione è mantenuto da un attributo speciale nella Context Entity, il quale viene aggiornato con la risposta ricevuta dal server per l'operazione di Execute.

3.7.2 IoT Broker GE

L'IoT Broker GE è utilizzato per la creazione e il mantenimento dei flussi di dati nei deployment IoT ed è previsto che venga eseguito su macchine di un datacenter dove serve come *middleware*. Questo modulo è progettato per interagire con un vasto numero di IoT Data Providers e IoT Data Consumers, nascondendo all'utente i dettagli tecnici delle varie installazioni IoT Fiware. Lo sviluppatore di Future Internet Application dovrà impostare la propria applicazione per comunicare tramite le interfacce NGSI con l'IoT Broker per poter ottenere i dati IoT di cui ha bisogno.

Come detto l'IoT Broker rappresenta il punto di contatto per accedere alle informazioni riguardo le Things. Per conto dei consumers l'IoT Broker recupera,

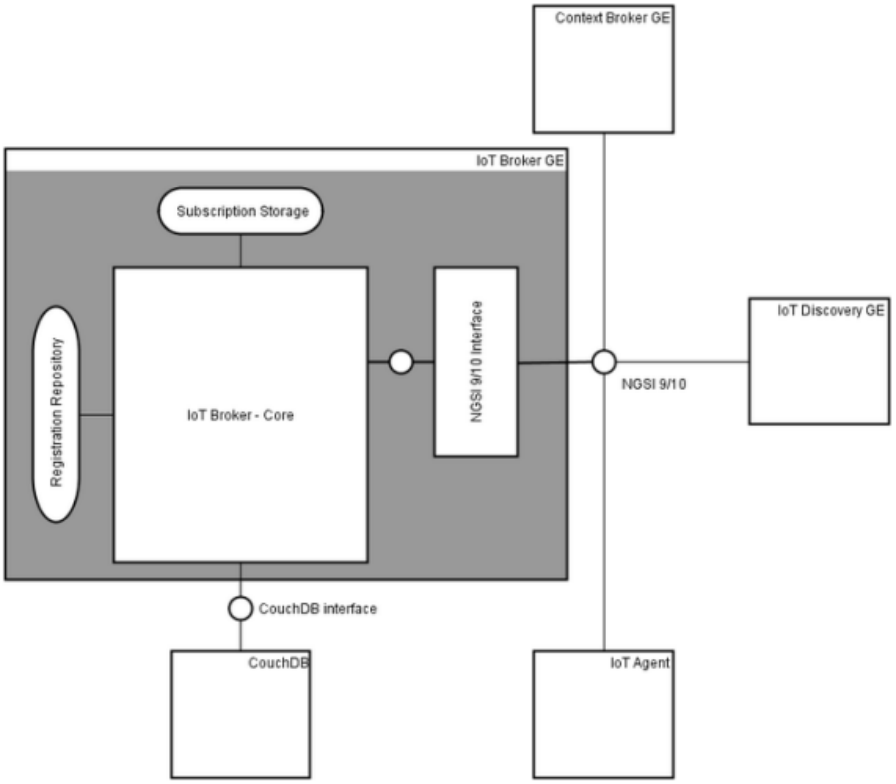


Figura 3.7: IoT Broker GE

assembla e processa i dati ottenuti dagli IoT providers offrendo ai consumers una semplice interfaccia che nasconde la complessità e l'etereogeneità dell'Internet of Things sottostante. L'IoT Broker raccoglie le informazioni sulle Context Entities NGSI e può

- aggregarle
- filtrarle
- tradurle
- arricchirle

prima di trasmetterle verso l'applicazione che ne ha fatto richiesta.

L'IoT Broker interagisce potenzialmente con un vasto numero di Gateway, altre istanze Backend, Devices, data consumers. Tipicamente interagisce con almeno un'istanza dell'IoT Discovery attraverso la quale recupera le informazioni su dove le context informations sono disponibili nell'installazione IoT. L'IoT

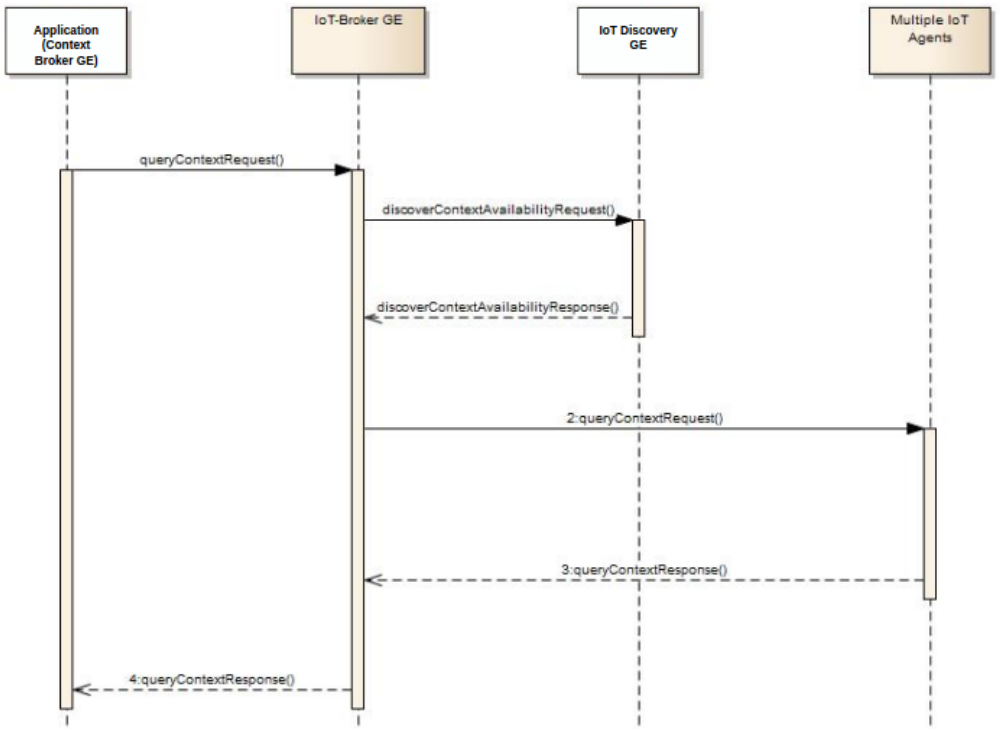


Figura 3.8: IoT Broker - Comportamento tipico

Broker GE comunica con il data consumer (un'applicazione o il Context Broker GE) tramite l'interfaccia NGSI da lato northbound. L'interfaccia NGSI da lato southbound è utilizzata invece per comunicare con gli IoT Agents, i quali forniscono i dati (Fig. 3.7).

Nell'utilizzo di base l'IoT Broker è un componente stateless nel senso che non memorizza context information ne context availability information. Quando l'IoT Broker riceve dal Context Broker (o direttamente da un'applicazione) una query su una Context Information (es. un attributo di una context entity), l'IoT Broker per prima cosa contatta l'IoT Discovery il quale esegue il discovery di quei dispositivi che potrebbero soddisfare la query. Utilizzando le informazioni sulla availability restiuite dal discovery attraverso NGSI-9, il Broker contatta gli information provider utilizzando NGSI-10 in modo da ricevere i valori attuali degli attributi coinvolti nella query. Le informazioni ottenute vengono poi aggregate e restituite all'applicazione che ne ha fatto inizialmente richiesta.

In un utilizzo più avanzato dell'IoT Broker vengono mantenute informazioni di stato all'interno di due repository.

Il Subscription Storage per memorizzare le richieste di subscription pervenute al Broker. Queste richieste possono arrivare sia da applicazioni che

da IoT Agents.

Context Registration per contenere informazioni riguardo alcune context information che possono essere ottenute in scenari più semplici dove non è presente l'IoT Discovery GE.

Operazioni dell'IoT Broker

L'interfaccia NGSI Context Information Management, utilizzata dall'IoT Broker, descrive tre tipi di operazioni per lo scambio di context information:

query, quando un'applicazione invoca una query si aspetta di ricevere context information come risposta.

subscription, quando un'applicazione fa una subscription (della quale ottiene il subscriptionId) ad una certa context information, riceverà le informazioni di contesto sotto la forma di notifications.

information update, quando una context information viene aggiornata, questa viene inviata all'IoT Broker senza la necessità di essere richiesta. Il broker la inoltrerà a qualche applicazione di default responsabile di processare/salvare tale informazione.

3.7.3 IoT Discovery GE

L'IoT Discovery GE è il modulo del livello Backend responsabile per la gestione delle informazioni sulla **availability** delle NGSI Context Entities relative alle Things. L'IoT Discovery GE è basato sul modello di informazioni definito da NGSI. Questo modello fornisce attraverso l'interfaccia NGSI-9 le seguenti funzionalità:

context availability registration - gli IoT Agents possono registrare le informazioni riguardanti le entity e i relativi attributi, rendendo in questo disponibile un punto di accesso per le risorse associate ad esse.

context availability discover - le informazioni riguardo la context availability sono richieste tramite operazioni di discovery da parte dei data consumer (tipicamente l'IoT Broker o il Context Broker)

context availability subscription e notification - viene realizzato un servizio di subscription-notify riguardo la context avaliability delle entità.

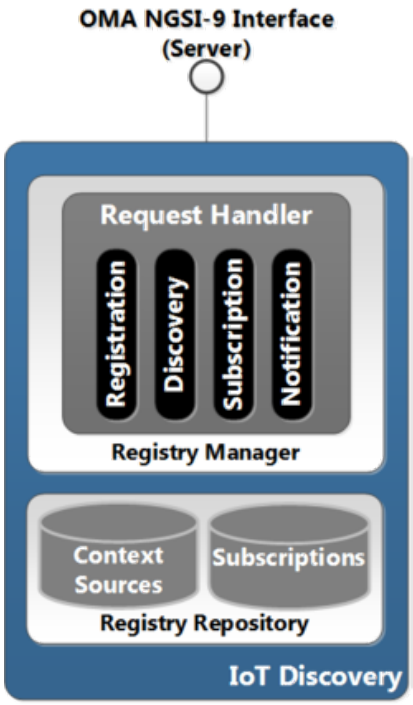


Figura 3.9: Struttura dell'IoT Discovery GE di FIWARE

L'architettura interna dell'IoT Discovery consiste in due componenti principali (Fig. 3.9).

Configuration Manager consiste in un registro nel quale i context provider possono registrare le loro context information. I context consumer che interagiscono con il Configuration Manager possono eseguire tramite l'interfaccia NGSI9 operazioni di discovery su tali registrazioni o sottoscriversi ai cambiamenti su di esse e riceverne le notifiche.

Configuration Repository memorizza le informazioni sulla availability delle context information e può essere acceduto attraverso il Config Manager.

Operazioni dell'IoT Discovery

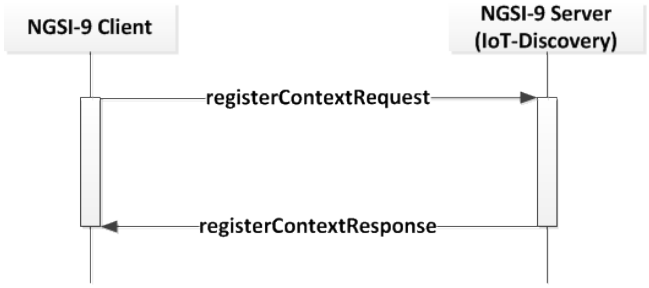


Figura 3.10: Registration

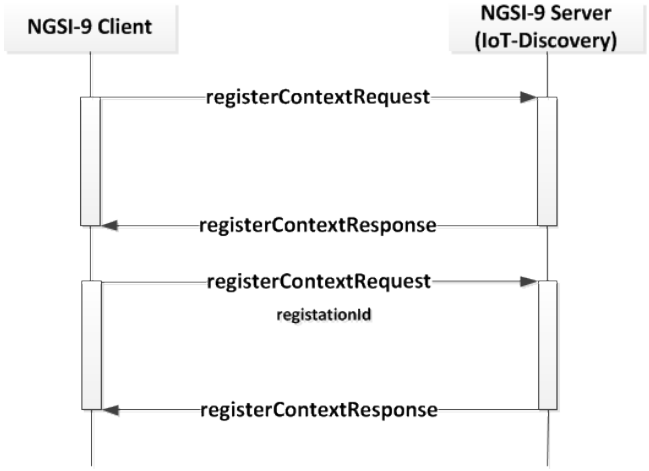


Figura 3.11: Update di una Registration

registration L'IoT Agent per rendere disponibili le sue informazioni sulle Thing, deve registrare queste informazioni all'IoT Discovery GE attraverso l'operazione di registerContext (Fig. 3.10). Questa operazione può essere effettuata a vari livelli di granularità

- registrando specifiche combinazioni di Entità/attributo
- registrando la disponibilità di certi tipi di entità.
- registrando più in generale che certe entity information sono disponibili

L'IoT Discovery risponde alla registrazione inviando al client il registrationId, il quale può essere utilizzato per effettuare un registration update (Fig. 3.11).

Figura 3.12: Discovery

discovery Il ruolo del Client tipicamente è svolto dall'IoT Broker GE, quando richiede delle *context availability information* per processare delle query/update NGSI-10, oppure da una qualsiasi applicazione NGSI che vuole conoscere la context availability associata ad una certa entità o attributo. La richiesta di discovery può ritornare zero, una o più istanze di *context registration* che soddisfano le richieste del client.

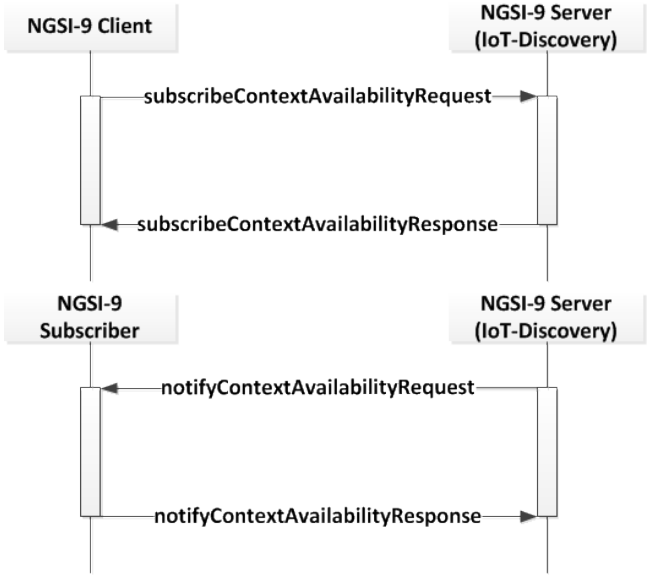


Figura 3.13: Meccanismo di Subscription e Notification

subscription e notify Il ruolo di NGSI-9 Client può essere svolto da tutte le applicazioni NGSI che devono essere consapevoli dei cambiamenti nelle context availability information. L'IoT Discovery notificherà il cambiamento di una context availability information a chi ha fatto la sottoscrizione ad essa legata (Fig. 3.13).

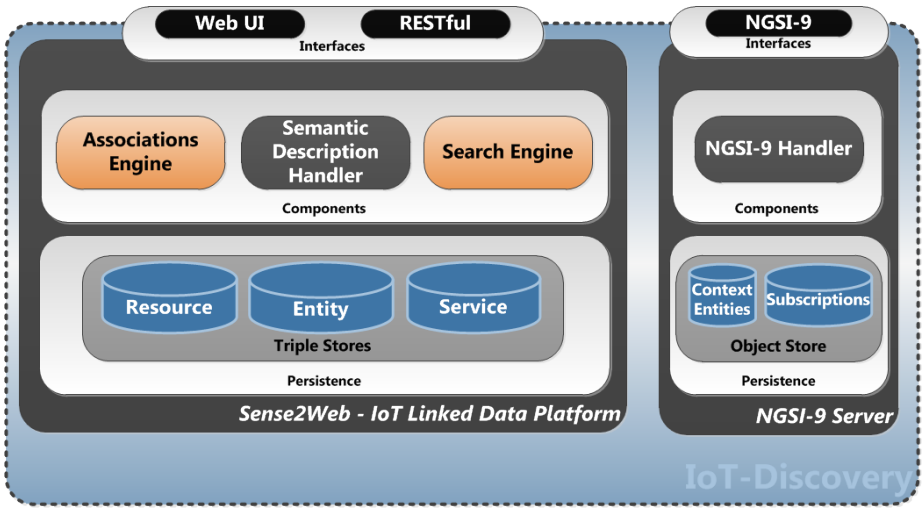


Figura 3.14: Moduli di IoT Discovery implementati da FIWARE

FIWARE attualmente mette a disposizione due implementazioni dell'IoT Discovery GE e le relative API. Entrambi i moduli fungono da service discovery mechanism (SDM) per IoT Descriptions, dove si ha una *directory centralizzata* per mezzo della quale si registrano e si possono scoprire informazioni circa le IoT entity (attributi, metadati, ecc) e come possono essere raggiunte (Fig. 3.14).

- **Sense2Web Platform** questo modulo è autonomo e permette agli utenti di registrare le IoT Description con la loro semantica.
- **NGSI-9 Server** gestisce le IoT Descriptions utilizzando il modello NGSI Context Entities, ciò significa che può essere utilizzato insieme agli altri GE che supportano l'interfaccia NGSI-9 (IoT Broker, Backend Device Management, Data Handling)

NGSI-9 Server

NGSI-9 Server è pensato per essere utilizzato come un servizio web e pertanto gli utenti e le applicazioni possono interagire con il GE tramite due serie di interfacce RESTful (in particolare HTTP-based). Questo modulo espone le interfacce per la registrazione, il discovery e il meccanismo di subscription e notification delle context information delle Entity. I componenti principali di NGSI-9 Server sono:

- **NGSI-9 Handler -** agisce da Configuration Manager. E' responsabile per la gestione delle richieste in base alla funzione corrispondente e anche per la gestione della rappresentazione della richiesta/risposta, in base a ciò il client invia e si aspetta (attualmente XML o JSON).

- **NGSI-9 Store** è responsabile per lo storage delle registrazioni e le subscription. Si occupa anche di interrogare lo storage sulla base delle richieste di discovery poste dal client.

Nell'attuale implementazione la persistenza utilizzata è un object store, in particolare è utilizzato il database per oggetti db4o.

Capitolo 4

Progettazione di *IoT-Discovery-Lorm*

4.1 Specifiche di Progetto

Le specifiche di progetto prevedono la realizzazione di un servizio di *Discovery* che operi in modo *distribuito* su più istanze sparse su una rete globale Internet e che possa essere inserito nell'ecosistema di *FIWARE*. Il servizio di discovery deve offrire le stesse funzionalità fornite dall'IoT Discovery GE implementato da FIWARE, a cui dovrà essere aggiunta la possibilità di fare lookup di tipo range e multi-attributo sulle entità registrate.

Il sistema progettato dovrà essere un'implementazione alternativa all'IoT Discovery Generic Enabler fornito da FIWARE, dovrà quindi essere conforme alle *Open Specification* che questo GE mette a disposizione. Il GE realizzato da FIWARE è stato pensato per operare in modo centralizzato, con un modulo che gestisce l'interfaccia NGSI-9 dalla quale riceverà le richieste e un modulo dedicato alla gestione dello *Storage* persistente delle registrazioni ricevute. In questa configurazione dovrà essere inserito un modulo che renda possibile la gestione distribuita delle informazioni riguardanti le Context Registration e permetta di eseguire range lookup multi attributo.

Il primo aspetto considerato in fase di progettazione è l'interfaccia con la quale *IoT-Discovery-Lorm* deve esporsi all'esterno. Questa dovrà rispettare le Open Specification fornite con il Generi Enabler. Valutate le strutture dati e le operazioni che il GE deve supportare, è stato deciso come realizzare la gestione distribuita delle informazioni su più istanze dell'applicazione, quali informazioni relative alle *Context Registration* memorizzare sui vari nodi e quali operazioni fosse possibile effettuare rispettando i vincoli imposti dal Context Information

Model e dalle operazioni NGSI-9

4.2 Interfaccia NGSI

Il Generic Enabler progettato deve essere conforme alle Open Specification relative all'IoT Discovery GE. Questo impone l'utilizzo dell'interfaccia e dei protocolli **NGSI-9** per comunicare con gli altri GE dell'architettura IoT di FIWARE [16]. Come illustrato nel paragrafo 3.5 questo modello entra nel dettaglio su come le Context Information sono strutturate e come vengono associate alle Context Entities in modo da descriverne lo stato. Le Context Information sono organizzate come **Context Elements**, che contengono un insieme di **Context Attributes** a cui possono essere associati anche dei **Metadata** (Fig. 3.5). Di seguito è illustrato nel dettaglio ogni elemento di questo modello.

4.2.1 Context Information Model

L'elemento di base del Context Information Model di NGSI è la *Context Entity*. Ci sono due tipi di Context Entity: il *Context Element*, utilizzato per rappresentare lo stato di una risorsa e il *Context Registration Element*, utilizzato per raggiungere le informazioni di una risorsa, in particolare dove può essere raggiunta.

Context Entity

Ogni Context Entity rappresenta una visione astratta di una Thing ed è composta da:

- EntityId E' una struttura che Identifica la Context Entity per la quale sono fornite le context information. E' composta da:
  - **ID**. Può essere una stringa nella forma anyURI, oppure un pattern rappresentato come espressione regolare
  - **Type**. (opzionale) è necessario nel caso in cui il tipo di entità non sia contenuto all'interno dell'entityId oppure quando l'Id è unico solo per tipo di entità. Il tipo di entità è definito come URI (una stringa che identifica univocamente una risorsa generica), quindi può essere un riferimento formale(URL) oppure un namespace (URN)
  - **IsPattern** (opzionale) indica se l'EntityID è un pattern (espresso come espressione regolare) oppure un Id.

- **ContextAttribute [0...unbounded]** è una lista di Context Attribute relativi all'entità. Se è specificato il dominio, tutti gli attributi appartengono ad esso.
- **AttributeDomainName** (opzionale) E' il nome del dominio degli attributi che raggruppa in modo logico l'insieme di ContextInformation Attributes.
- **DomainMetadata [0...unbounded]** (opzionale) sono metadati comuni a tutti gli attributi del dominio logico.

Context Attribute

Ad ogni Context Element possono essere associati dei Context Attributes. Gli attributi rappresentano le risorse offerte da una Thing (es. temperatura, umidità) e sono definiti come:

- **name**. Nome dell'attributo
- **type** (opzionale) Indica il tipo del valore nel campo value. Il tipo dell'attributo è definito come URI
- **ContextValue**. E' il valore attuale assunto dall'attributo.
- **metadata** (opzionale) Ad ogni attributo può essere associato in modo opzionale uno o più metadata.

Context Metadata

Ad ogni Context Attribute possono essere associati dei Context Metadata [0...unbounded], definiti come:

- **name**. Nome del metadata.
- **type** (opzionale) Indica il tipo del valore nel campo value. Il tipo dell'attributo è definito come un URI
- **value**. E' il valore attuale assunto dal metadata.

Nel Context Element possono essere definiti in modo opzionale dei DomainMetadata [0...unbounded], ovvero dei metadata validi per tutti gli attributi nel dominio logico (cioè all'attribute domain). E' presenta anche una lista di Context Metadata riservati (Timestamp, Expires, Source, ID)

4.2.2 Operazioni NGSI-9

L'interfaccia NGSI-9 mette a disposizione le operazioni di **registration**, **discovery**, **subscription** e **notification**. Vengono illustrate nel dettaglio qui di seguito insieme alle strutture dati di cui necessitano.

Register Context Entity

Questa operazione permette la registrazione di una *Context Entity*, i nomi dei suoi attributi e della sua *availability*.

- **Input message** - *RegisterContextRequest* Nel messaggio di richiesta di registrazione devono essere incluse
  - *ContextRegistrationList*, cioè una o più [1...unbounded] Context Registration.
  - *Duration* (opzionale) indica il periodo di validità della availability
  - *RegistrationID* (opzionale), quando si vuole aggiornare una Context Entity già registrata.
- **Output message** - *RegisterContextResponse*
  - **RegistrationID** è l'identificativo della registrazione appena avvenuta
  - **Duration** (opzionale) Conferma il periodo di validità
  - **ErrorCode** (opzionale) Riporta l'eventuale errore avvenuto durante l'operazione

Con questa operazione è possibile eseguire l'aggiornamento delle informazioni di entità già registrate, specificando il RegistrationId ricevuto in risposta in seguito alla registrazione.

Struttura di una ContextRegistration

Questa struttura specifica tutte le informazioni sulle entità che devono essere registrate e le informazioni su chi le sta fornendo

- **EntityIdList**. (opzionale) è una lista di identificatori (EntityId) delle Context Entity che devono essere registrate
- **ContextRegistrationAttribute** [0...unbounded] E' una lista di *Context Attributes* e/o Attribute Domains che vengono resi disponibili con la registrazione

- **RegistrationMetadata** [0...unbounded] Metadati relativi alla registrazione.
- **ProvidingApplication**. è un URI che indentifica l'applicazione che fornisce i valori dei Contex Attributes relativi alle Context Entities registrate.

Discover Context Availability

Questa operazione consente di eseguire in modo sincrono il discovery di un insieme potenziale di Context Entities. Il controllo della query di discovery avviene solo sulle Context Entities che sono state precedentemente registrate tramite operazioni di *RegisterContext* e non sulle reali sorgenti delle Context Information. Quest'operazione restituisce una lista aggregata di *ContextRegistrations* (vedere sopra la loro struttura)

- **Input Message** - *discoverContextAvailabilityRequest* Nel messaggio di richiesta di discovery devono essere incluse:
  - **EntityID [1...unbounded]** E' una lista di Modifier per identificare la/le Context Entity(ies) da scoprire. Vedi sopra la struttura dell'EntityID
  - **AttributeList [0...unbounded]** (string) E' una lista di attributi o gruppi di attributi da scoprire.
  - **Restriction** (opzionale) Sono delle restrizioni sugli attributi e sui metadati delle Context Information
- **Output Message** - *discoverContextAvailabilityResponse* La risposta alla richiesta di discovery dovrà esattamente contenere uno dei seguenti elementi
  - **ContextRegistrationResponceList [0...unbounded]** E' la lista di *ContextRegistrationsResponse* che soddisfano la richiesta
  - **ErrorCode** (opzionale) Indica il codice di errore per errori generici nell'operazione

Restriction Structure

La struttura dati per le Restriction contiene due differenti tipi di restrizioni:

- **AttributeExpression** (String) E' una stringa contenente una restrizione *XPath*. La attribute expression serve a filtrare l'insieme di risultati in base a espressioni sui valori dei context attributes. L'espressione XPath sarà valutata nei confronti della struttura dell'entità.

- **Scope [0...unbounded]** Si definisce una lista di *OperationScope* che restringe lo spazio di ricerca su cui una data operazione deve operare. Lo Scope limita a priori l'insieme di context sources che sono necessarie a servire la richiesta. La struttura OperationScope definisce:
  - **ScopeType** (string) la quale seleziona il tipo di scope (il suo nome) che viene utilizzato.
  - **ScopeValue** (anyValue) il quale definisce i parametri dello scope, contiene quindi il valore dello scope per il tipo definito.

ContextRegistrationResponce Structure

E' costituita da:

- **ContextRegistration** cioè la *Context Registration* che soddisfa la richiesta.
- **ErrorCode** (opzionale) Identifica lo stato dell'operazione richiesta relativa a questa ContextRegistration. Se non c'è errore questo elemento dev'essere omesso.

Subscribe Context Availability

Questa operazione consente di eseguire in modo asincrono il discovery di un insieme potenziale di Context Entities e le Context Information relative ma non garantisce che le Context Information riguardanti una Context Entity ottenute, siano attualmente disponibili. In altre parole, questa operazione permette la sottoscrizione a delle notifiche sulla *availability* di una lista aggregata di Context Registration.

- **Input Message** - *subscribeContextAvailabilityRequest*
  - **EntityId [1...unbounded]** è una lista di identificatori (EntityId) delle Context Entity da scoprire
  - **AttributeList [0...unbounded]** (string) E' una lista di attributi o gruppi di attributi da scoprire.
  - **Reference** indica l'interfaccia di riferimento a cui dovrà essere notificata la ContextAvailability
  - **Duration** (opzionale) ed indica il periodo di validità della subscription
  - **Restriction** (opzionale) Sono delle restrizioni sugli attributi e sui metadati delle Context Information

  - **SubscriptionId** (opzionale) identificatore della subscription.

- **Output Message** - *subscribeContextAvailabilityResponse.*
  - **SubscriptionId** identificatore della subscription, è utilizzato nei messaggi di notifica per identificare a quale richiesta essa è legata
  - **Duration** (opzionale) negoziazione della durata della subscription
  - **ErrorCode** (opzionale) Indica il codice di errore per errori generici nell'operazione

Notify Context Availability

Questa operazione consente la ricezione delle notifiche riguardanti un insieme potenziale di Context Registration sottoscritte.

- **Input Message** - *notifyContextAvailabilityRequest*
  - **SubscriptionId** identificatore della subscription a cui la notifica appartiene
  - **ContextRegistrationResponceList [0...unbounded]** E' la lista di *ContextRegistrationsResponse* che soddisfano la richiesta
  - **ErrorCode** (opzionale) Indica il codice di errore per errori generici nell'operazione

- **Output Message** - *notifyContextAvailabilityResponse.*
  - **StatusCode** indica il codice dello stato per errori generici nell'operazione

4.3 LORM

Per rendere il servizio di discovery distribuito su più istanze dell'applicazione, è stato deciso di utilizzare una architettura P2P basata su una Distributed Hash Table. In particolare è stata sfruttata l'implementazione di **LORM** [6, 7] realizzata da Marco Lupi nel suo lavoro di tesi.

LORM (**L**ow **O**verhead, **R**ange-query, **M**ulti-attribute) propone un servizio di *resource discovery* basato su una rete DHT a grafo composto, in cui lo scheletro della DHT deriva da Cycloid [13]. LORM organizza ogni nodo della DHT per essere responsabile delle informazioni relative ad una risorsa che possiede uno specifico attributo il cui valore è interno ad un range (Fig. 4.1). Ogni nodo viene identificato attraverso un ID composto da due indici: l'**indice cubico** che

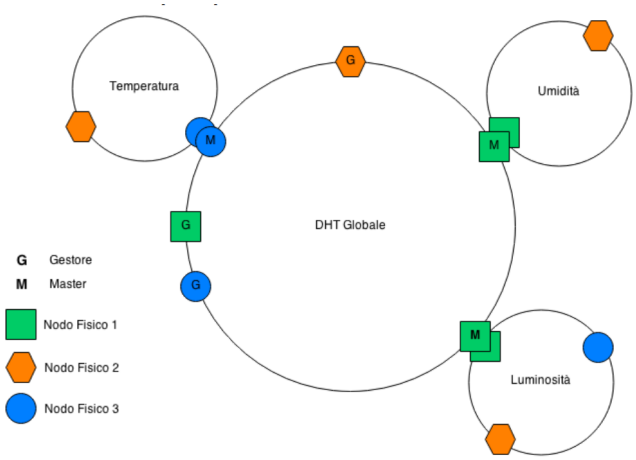


Figura 4.1: Distribuzione delle informazioni in LORM

rappresenta l'attributo a e l'**indice ciclico** che rappresenta il valore π dell'attributo. Nella struttura della DHT di LORM si distinguono quindi due livelli.

La DHT Globale, è il livello in cui sono presenti i nodi *master* degli attributi, i quali permettono di accedere al cluster che gestiscono. Per ogni attributo gestito da Lorm sarà presente nella DHT Globale un nodo master e il relativo cluster. Ogni cluster è identificato dall'indice cubico ottenuto dall'hash function SHA1 applicata alla stringa che descrive l'attributo.

Il Cluster è il livello che gestisce effettivamente le informazioni delle risorse relative all'attributo di cui si occupa. Per identificare all'interno di un cluster i nodi dove inserire e ricercare le informazioni relative ad una risorsa, viene utilizzato l'indice ciclico. Questo indice viene generato in due modi diversi a seconda del tipo di cluster:

- se il cluster gestisce un attributo con valori numerici sui cui sarà possibile eseguire una range-query, l'indice ciclico sarà ottenuto utilizzando una funzione *hash locality preserving*

$$H(\pi) = (\pi - \pi_{min})\frac{(d-1)}{(\pi_{max} - \pi_{min})}$$

  dove π è il valore dell'attributo caratterizzato dal range di valori $[\pi_{min}, \pi_{max}]$, mentre d è il valore massimo dell'ID.

- se il cluster gestisce un attributo con valori non numerici o sui cui non è necessario preservare la località, l'indice ciclico sarà ottenuto con la funzione hash SHA1 applicata al valore dell'attributo.

Sia la DHT globale che i Cluster sono gestiti con Pastry [17]. L'implementazione utilizzata è *FreePastry*.

Ogni istanza di LORM partecipante alla DHT può ospitare più nodi e quindi partecipare a più cluster. Per rendere più chiara la comprensione vengono definiti i vari tipi di nodo coinvolti.

- **Nodo Fisico** è un'instanza partecipante alla DHT Lorm. Ogni nodo fisico crea un *nodo gestore* da inserire nella DHT globale, un certo numero di *nodi master*, uno per ogni attributo gestito per il quale ancora non esiste un cluster e *n nodi virtuali*, ognuno dei quali sarà inserito in un cluster relativo all'attributo che il nodo fisico gestisce da configurazione iniziale

- **Nodo Gestore** è il primo nodo che viene creato da un nodo fisico. Questo nodo virtuale è connesso alla DHT Globale e gestisce tutte le operazioni sui nodi virtuali posseduti dal nodo fisico, dalla loro creazione (compresa quella dei nodi master) alle operazioni di inserimento e lookup di un dato.

- **Nodo Master** è il nodo referente per un attributo. Questo nodo viene creato nel momento in cui un nodo fisico che gestisce un attributo entra nella DHT e in essa non è ancora presente un cluster relativo a tale attributo. Il nodo master viene inserito all'interno della DHT globale, ma ha la particolarità di avere un nodo virtuale identico all'interno del cluster, questo per permettere di inoltrare le richieste che viaggiano nella DHT globale all'interno del cluster.

- **Nodo Virtuale** è il nodo che partecipa ad un cluster e che effettivamente memorizza i dati relativi ad una risorsa che possiede l'attributo relativo al cluster in cui il nodo virtuale è inserito. Il nodo virtuale gestisce quindi nella memoria persistente tutte le entry che hanno l'indice numericamente più vicino all'identificatore del nodo virtuale rispetto a quello dei nodi adiacenti.

4.4 Architettura interna di *IoT-Discovery-Lorm*

IoT-Discovery-Lorm è realizzato a partire dall'IoT Discovery Generic Enabler implementato da FIWARE, utilizzando le *data structure* e le operazioni definite nel modello delle informazioni NGSI-9. Si possono distinguere due moduli principali all'interno dell'IoT Discovery GE.

- Il **REST Request Handler** è il modulo che si occuperà di intercettare le chiamate restful inviate da un client. Per ogni richiesta su cui l'handler è messo in ascolto è mappata una delle operazioni NGSI9 previste dallo standard. In caso di registrazione di un'entità il payload della richiesta contente la Context Registration, se valido, viene inviato al Configuration repository per essere memorizzato. In caso invece di una richiesta di Discovery il payload della richiesta viene interpretato e viene interrogato il repository per ottenere le eventuali Context Registration che soddisfano la richiesta.
- Il **Configuration Repository** è il modulo che realizza la memoria persistente per salvare le informazioni sull'availability delle Context Informations. Nel local storage saranno salvate tutte le Context Registrations ricevute dal REST Request Handler

Per rendere questo sistema distribuito è stato progettato un terzo modulo da inserire all'interno del sistema già esistente. Tale modulo, il *LORM Core* (Fig. 4.2), dovrà distribuire le informazioni rilevanti sulle Context Entity registrate in ogni istanza di *IoT-Discovery-Lorm*. LORM Core mette a disposizione le funzionalità illustrate in 4.3, gestirà il protocollo della DHT, le interazioni tra le varie istanze di *IoT-Discovery-Lorm* in esecuzione e un repository in cui sono memorizzate le entry salvate sui nodi virtuali che gestisce.

4.5 Comportamento e diagrammi di sequenza

4.5.1 Comportamento della DHT

Qua verranno descritte le operazioni principali realizzate in LORM e il loro comportamento rispetto alla DHT.

Ingresso di un nuovo nodo nella DHT

Per accedere alla DHT, un nodo fisico deve conoscere l'indirizzo e la porta di boot di un nodo già presente all'interno di LORM (Fig. 4.3). Nel caso in cui sia esso il primo nodo a entrare nella DHT dovrà occuparsi esso stesso della creazione della DHT. Il primo passo che viene effettuato è la creazione del Nodo

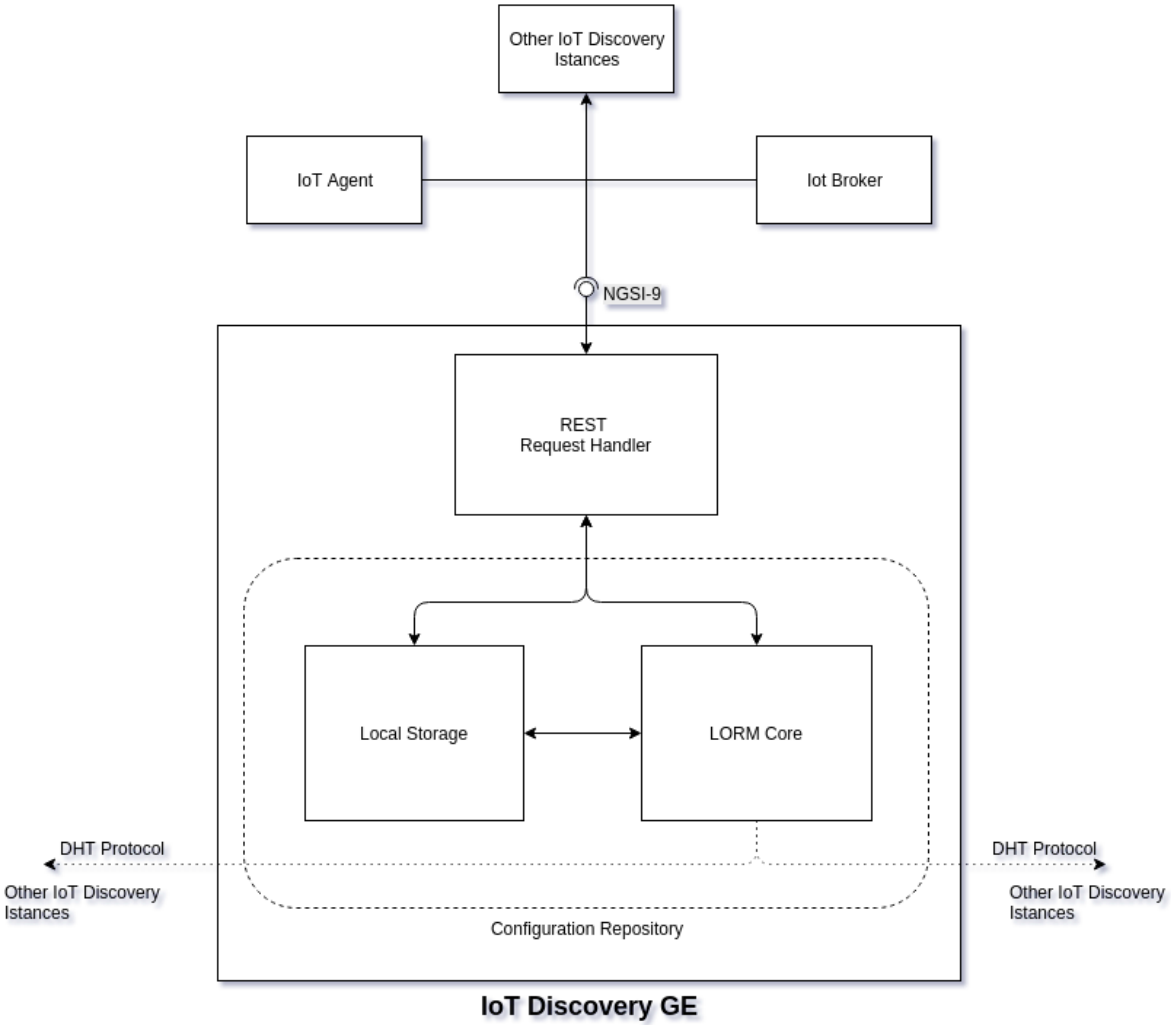


Figura 4.2: Componenti di *IoT-Discovery-Lorm*

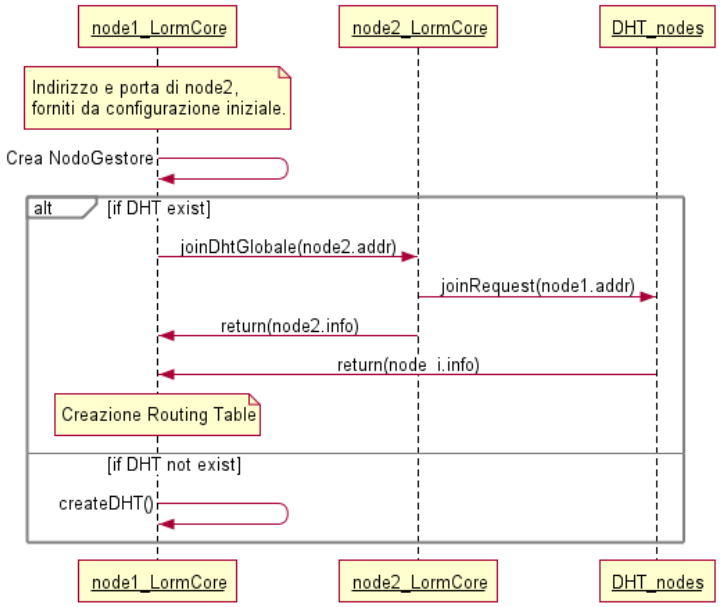


Figura 4.3: Ingresso di un nodo nella DHT

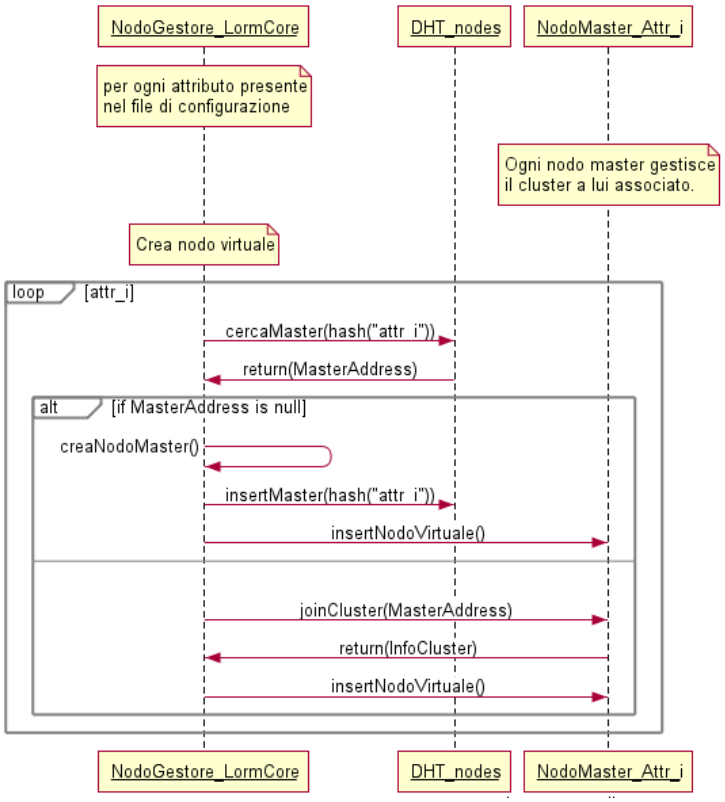


Figura 4.4: Creazione dei nodi virtuali nella DHT

Gestore il quale farà la richiesta di connessione alla DHT globale, diventandone a tutti gli effetti uno dei nodi partecipanti. Durante la richiesta di *join*, il nodo contattato inoltrerà tale richiesta a tutti i nodi conosciuti i quali faranno altrettanto, in modo che tutti vengano a conoscenza del nuovo nodo e possano inviargli le informazioni necessarie a costruirsi la propria routing table.

Il passo successivo (Fig. 4.4) prevede la creazione dei nodi master nel caso in cui il nodo fisico venga configurato per gestire degli attributi, il cui relativo cluster non è ancora stato creato. Successivamente alla creazione di eventuali nodi master il nodo gestore ordinerà la creazione di un nodo virtuale all'interno del cluster relativo ad un attributo che il nodo fisico dovrà gestire.

Listing 4.1: Pseudo codice di un nodo che vuole accedere alla DHT

```
CreaNodi(lista di attributi da gestire) {
       creo il Nodo Gestore nella DHT Globale
       for (ogni attributo in lista) {
              verifico se presente il Master
              if (MasterPresente) {
                     creo il nodo virtuale per l'attributo
              } else {
                     aggiungo il Master alla lista dei Master da creare
                     creo il cluster e il nodo per l'attributo
              }
       }
       for (ogni Master nella lista dei Master) {
              creo il Master dell'attributo corrispondente nella DHT Globale
              creo il nodo virtuale per l'attributo
       }
}
```

Inserimento di un dato

Per eseguire l'inserimento di un dato all'interno della DHT (Fig. 4.6) si devono differenziare due casi in cui si può trovare il nodo fisico che vuole effettuare l'operazione. Nel primo caso il Nodo Fisico gestisce tra i suoi Nodi Virtuali un nodo appartenente al cluster dell'attributo di cui si vuole inserire il valore, nel secondo caso il Nodo Fisico non gestisce questo tipo di nodo e deve adoperarsi per trovare un altro Nodo Fisico che possa finalizzare l'operazione di inserimento. Il nodo Gestore cerca se lui stesso ha in gestione il nodo virtuale interno al cluster dell'attributo coinvolto. Se il gestore trova il nodo allora può utilizzarlo per effettuare l'inserimento del dato e delle informazioni relative alla risorsa a cui è associato. Nel caso in cui il nodo che effettua l'inserimento non abbia tra i suoi Nodi Virtuali un nodo facente parte del cluster relativo all'attributo di cui fa parte il dato, il nodo gestore invierà alla DHT una richiesta di inserimento al nodo master il cui ID è ottenuto dall'hash del nome dell'attributo. Il nodo master che riceverà la richiesta potrà gestire o meno l'attributo, nel caso non lo gestisca ritornerà indietro un errore di "attributo non esistente", altrimenti effettuerà l'operazione di inserimento interna al cluster. In entrambi in casi l'inserimento sarà effettuato all'interno del nodo virtuale presente nel cluster il cui Id è numericamente più vicino all'indice ciclico ottenuto dalla funzione hash applicata al valore.

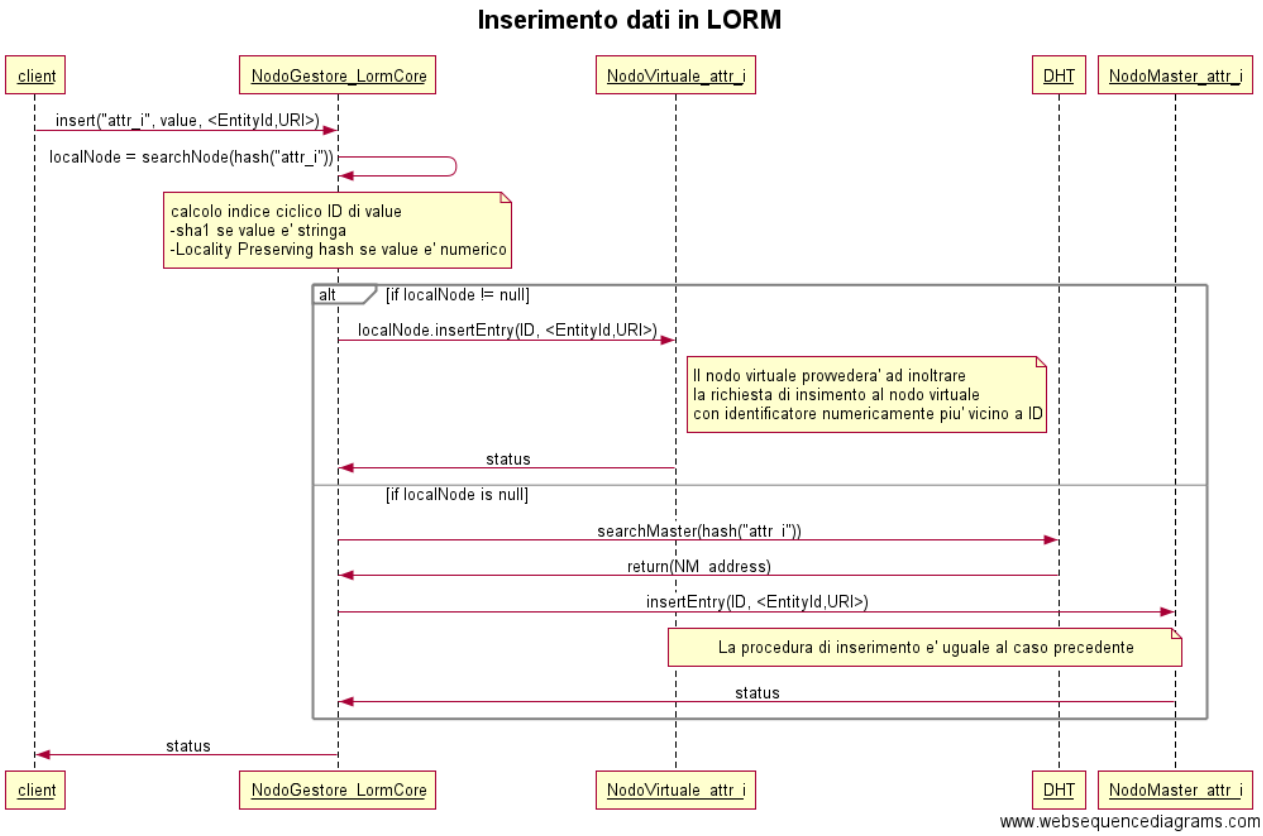


Figura 4.5: Inserimento di un dato in LORM

Listing 4.2: Pseudo codice di una operazione di inserimento

```
InserisciElemento(attributo, valore, contenuto) {
       cerco un nodo locale facente parte del cluster 'attributo'
       if (nodoTrovato) {
              inserisco l'elemento in posizione 'localityPreservingHash(valore)'
       } else {
              invio il dato nella DHT Globale al nodo con identificatore
              hash(attributo)
              // in modo asincrono aspetto la risposta di inserimento effettuato
       }
}
```

Lookup

L'operazione di lookup è identica nel comportamento all'operazione di inserimento. Il Nodo Gestore controlla se tra i suoi Nodi Virtuali ce n'è uno che può effettuare l'operazione di lookup. Come per l'inserimento si possono verificare due casi: il nodo gestore può eseguire il lookup tramite un nodo da lui gestito oppure deve inviare la richiesta all'interno della DHT e aspettare in modo asincrono la risposta contenente i dati richiesti.

Range-Lookup

Il comportamento della Range Lookup nel caso in cui il nodo fisico non gestisca il tipo di attributo richiesto, è identico a quello di una operazione di inserimento o di lookup. Invia un messaggio all'interno della DHT per contattare il nodo master che può gestire la richiesta.

La differenza nel comportamento si ha nel modo in cui vengono effettuate le richieste all'interno dei cluster. Facendo riferimento alla Figura 4.6 si consideri il Nodo Gestore (NG), interno al Nodo Fisico, come il nodo che richiede una range lookup che ha come intervallo di valori $[Val_{min}, Val_{max}]$. NG invierà il messaggio al primo nodo interno al range, cioè il nodo che dovrebbe contenere $Lhash(Val_{min})$, con $Lhash()$ funzione hash locality preserving. Supponiamo che il primo nodo del range sia NVn (appartenente a NFn), quest'ultimo risponderà al NG inviando i dati che soddisfano il range e la parte del suo *leaf set* che include gli identificatori dei NV che potrebbero contenere dati interni al range. Una volta ricevuta parte del leaf set di NVn che comprende i nodi interni al range, NG invierà in parallelo tante richieste di Range Lookup quanti sono i nodi della lista. Ogni nodo che riceverà il messaggio risponderà con i suoi dati e il suo leaf set filtrato, cioè contenente solo gli identificatori dei nodi interni al range. NG analizzerà i nuovi leaf set e verificherà la presenza di nuovi nodi a cui mandare la richiesta di range lookup, se sono presenti manderà tante altre

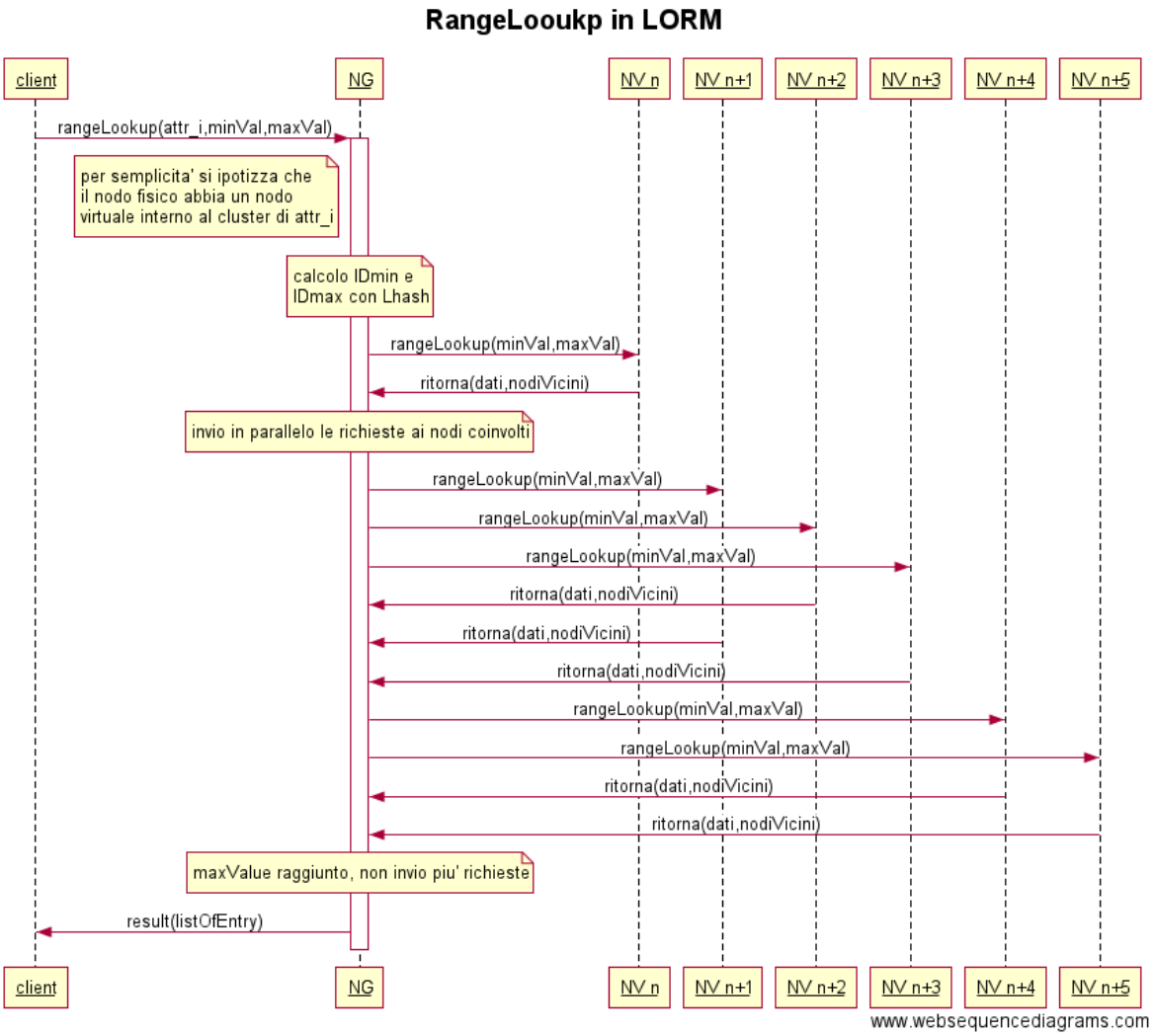


Figura 4.6: Range Lookup

richieste in parallelo per quanti essi sono, e così via ricorsivamente. L'algoritmo si fermerà se e solo se sono stati interrogati tutti i nodi all'interno del range, più uno o più nodi successivi ad esso dato che potrebbero contenere dei valori interni al range stesso.

Listing 4.3: Pseudo-codice di una operazione di Range Lookup

```
CercaElementiInRange(attributo, min, max) {
       cerco un nodo locale facente parte del cluster 'attributo'
       if (nodoTrovato) {
              invio richiesta di rangeLookup il al nodo in posizione
              'localityPreservingHash(min)'
              // avro' in risposta parte della sua Routing Table
              for (nodi nella Routing Table) {
                     Nuovo Thread: invio richiesta di rangeLookup al nodo[i]
                     // all'interno del thread il nodo aspetta le routing table
                     // di risposta e se trova nuovi nodi a cui non e' stata
                     // ancora inviata la richiesta, apre nuovi thread per i
                     // nuovi nodi, e cosi via ricorsivamente
              }
       } else {
              invio la richiesta di rangeLookup nella DHT Globale al nodo con
              identificatore hash(attributo)
              // in modo asincrono aspetto le risposte contenenti i contenuti
              // dei dati
       }
}
```

4.5.2 Richieste di ContextRegistration

Questa operazione permette la registrazione delle Context Entities, dei nomi dei loro attributi e di eventuali Metadata ad essi associati 4.7. In ogni richiesta di registrazione possono essere presenti una o più ContextRegistration caratterizzate ognuna da un EntityID una possibile lista di attributi, ognuno dei quali può avere dei metadata. Quando viene ricevuta una richiesta di registrazione dal REST Request Handler, il payload della richiesta conterrà la descrizione della richiesta in formato XML o JSON, la quale viene tradotta tramite un'operazione di *unmarshalling* in un oggetto Java che rappresenta la ContextRegistration. La context registration sarà memorizzata all'interno del local storage e le informazioni rilevanti della risorsa saranno inserite tramite un'operazione di *insert* nella DHT. Per rendere accessibili le ContextRegistration a tutte le istanze di *IoT-Discovery-Lorm* connesse alla DHT, dovrà essere inserito nella DHT l'EntityId (*Id;Type*) tramite l'operazione di *insert* resa disponibile da LORM. L'EntityId sarà inserito nel cluster di tipo descrittivo dedicato a questo *attributo*[1] secondo

[1]Attributo in corsivo, per riprendere la terminologia di LORM, dove ad ogni cluster è associato un attributo.

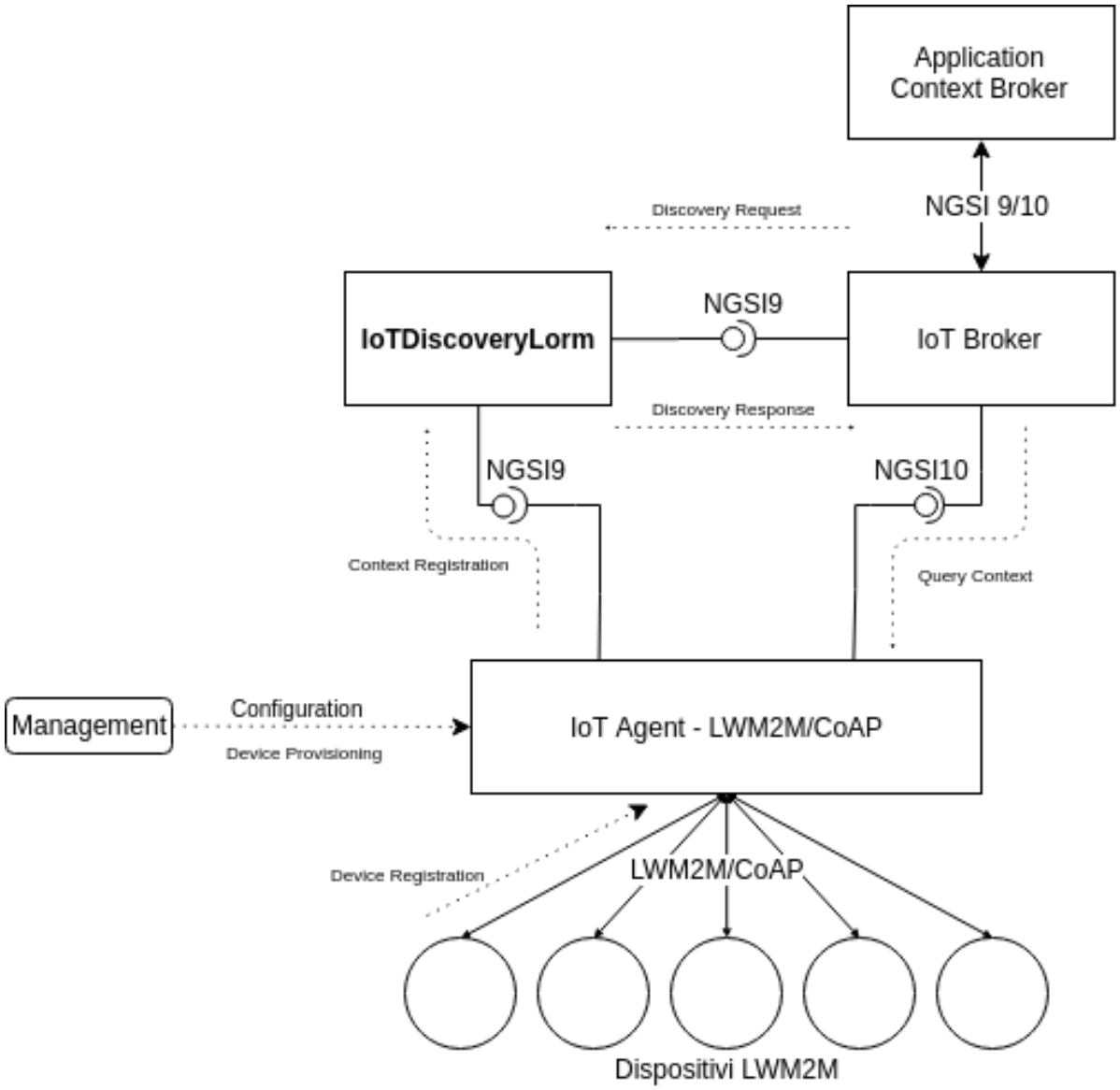


Figura 4.7: Richieste tra i vari IoT Generic Enablers

le modalità descritte in Fig. 4.5. Se nella Context Registration sono presenti dei metadata con valori numerici associati ad un attributo, su cui è prevista la possibilità di eseguire delle range-query (es. latitude, longitude), per ognuno di essi sarà inserito il valore del metadata all'interno della DHT. L'inserimento sarà effettuato all'interno del cluster relativo alla coppia attributo-metadata. Perché ciò sia possibile almeno una delle istanze di *IoT-Discovery-Lorm* deve essere configurata per poter creare e gestire tale cluster, dev'essere infatti noto a priori il valore minimo e il valore massimo accettati per calcolare l'indice cubico di ogni entry. L'operazione di inserimento prevede a tal proposito la possibile gestione di due situazioni di errore

- Coppia di attributo-metadato non conosciuta. Ovvero se nessuna istanza di *IoT-Discovery-Lorm* è configurata per gestire tale cluster
- Valore *OutOfRange*, cioè quando si tenta di inserire un valore del metadato che non è interno al range $[minVal, maxVal]$

In figura 4.8 è illustrato il diagramma di sequenza dell'operazione di registrazione.

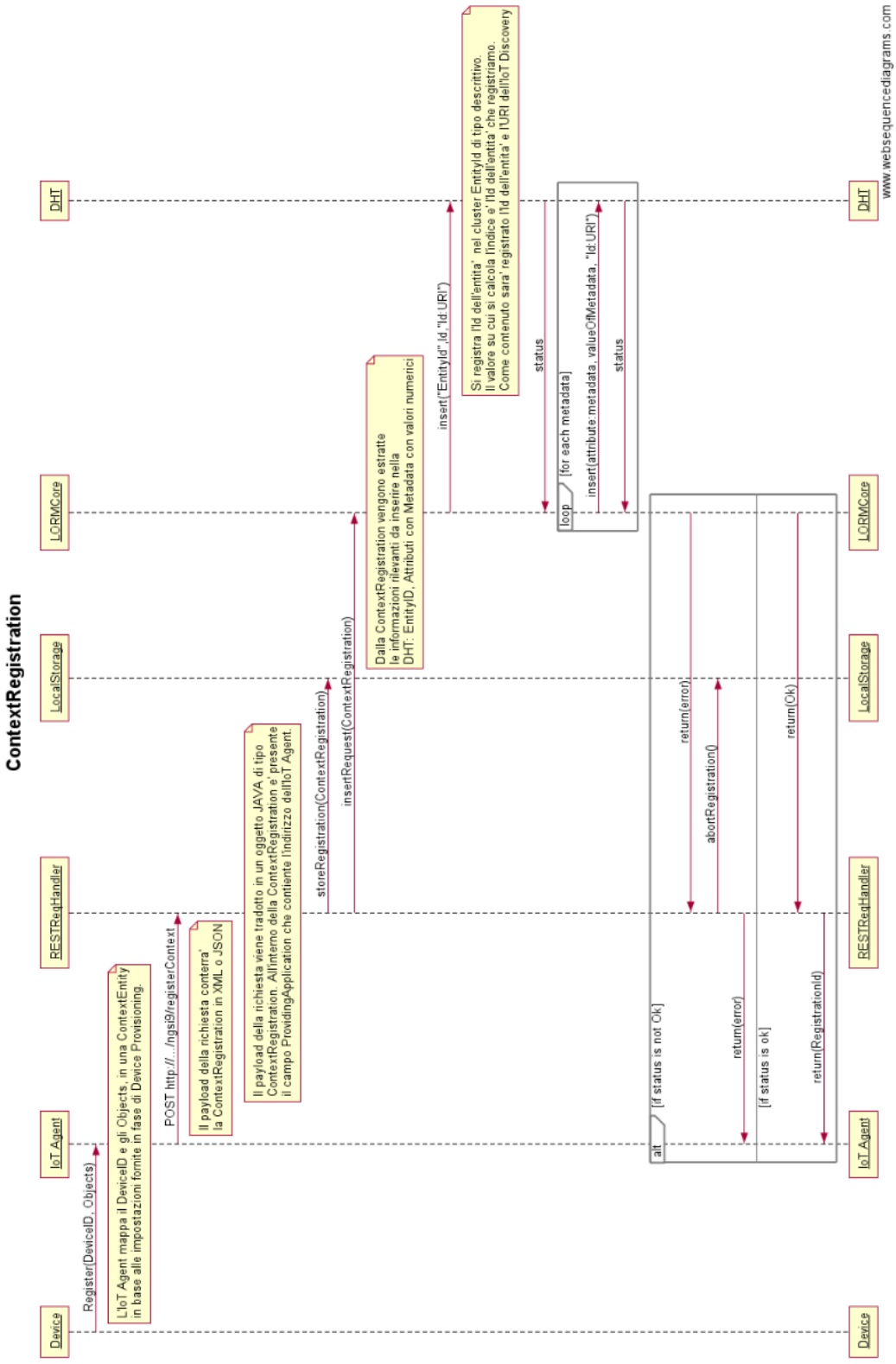


Figura 4.8: Context Registration

Listing 4.4: Esempio di richiesta NGSI9 di ContextRegistration in formato XML.

```
http://10.0.3.1:8080/ngsi9/registerContext
<?xml version="1.0"?>
<registerContextRequest>
       <contextRegistrationList>
              <contextRegistration>
                     <entityIdList>
                            <entityId type="robot" isPattern="false">
                                   <id>Robot1</id>
                            </entityId>
                     </entityIdList>
                     <contextRegistrationAttributeList>
                            <contextRegistrationAttribute>
                                   <name>temperature</name>
                                   <type>integer</type>
                                   <isDomain>false</isDomain>
                                   <metadata>
                                          <contextMetadata>
                                                 <name>longitude</name>
                                                 <type>integer</type>
                                                 <value>25</value>
                                          </contextMetadata>
                                   </metadata>
                            </contextRegistrationAttribute>
                            <contextRegistrationAttribute>
                                   <name>humidity</name>
                                   <type>integer</type>
                                   <isDomain>false</isDomain>
                                   <metadata>
                                          <contextMetadata>
                                                 <name>longitude</name>
                                                 <type>integer</type>
                                                 <value>35</value>
                                          </contextMetadata>
                                   </metadata>
                            </contextRegistrationAttribute>
                     </contextRegistrationAttributeList>
                     <providingApplication>http://131.114.59.22:4041/ngsi10</
                         providingApplication>
              </contextRegistration>
       </contextRegistrationList>
       <duration>P1M</duration>
</registerContextRequest>
```

4.5.3 Richieste di Discovery

Questa operazione permette potenzialmente di scoprire in modo sincrono un insieme di Context Entities e le relative Context Informations (attributi e metadati), che soddisfano i parametri presenti nella richiesta 4.7. In ogni richiesta di discovery possono essere presenti una lista di EntityId, una lista di attributi e una lista di restriction, per permettere varie tipologie di richieste. Entrando nel dettaglio, IoT Discovery permette di eseguire sul Local Storage delle richieste di discovery dove l'EntityId può essere indicato in modo esplicito (es. robot1:robot) oppure in forma di pattern (es. .*). Nel primo caso sarà ricercata la Context Registration che possiede quell'EntityId verificando che possieda tra i suoi attributi *almeno uno* di quelli specificati nella richiesta. Nel secondo caso, saranno ricercati nel Local storage tutte le Context Registration il cui EntityId soddisfa il pattern nella richiesta. Anche in questo caso ogni risultato ottenuto possieda *almeno uno* degli attributi specificati nella richiesta. Questo illustrato è il comportamento di questa operazione nel caso in cui non siano specificate delle restriction. Nell'implementazione di *IoT-Discovery-Lorm* è stata introdotta la possibilità di gestire due tipi di restriction *OperationScope*:

Discovery con Scope Globale (Fig. 4.9) in cui è possibile estendere la ricerca delle Context Entities anche ad altre istanze di *IoT-Discovery-Lorm* connesse alla stessa DHT-LORM. Quando viene specificata questa restriction, il comportamento dell'operazione di discovery risulta diverso. Per ogni EntityId specificato nella DiscoveryRequest sarà fatta una richiesta a LORM per una operazione di lookup sul cluster EntityId per l'Id specificato. Ogni risultato ricevuto conterrà l'EntityId ricercato e l'URI dell'istanza di *IoT-Discovery-Lorm* che possiede effettivamente la ContextRegistration relativa. Per rendere più efficiente la fase successiva, se sono presenti più risultati, questi saranno raggruppati per URI differenti. Per ognuno di essi sarà costruita una richiesta REST di tipo DiscoveryRequest locale in cui saranno specificati gli EntityId voluti. Caso particolare è quello in cui l'URI del risultato è quello locale e quindi la Context Registration sarà semplicemente ricercata all'interno del Local Storage. Ottenute le ContextRegistration di tutti i risultati, può essere confezionata la risposta da inviare al client.

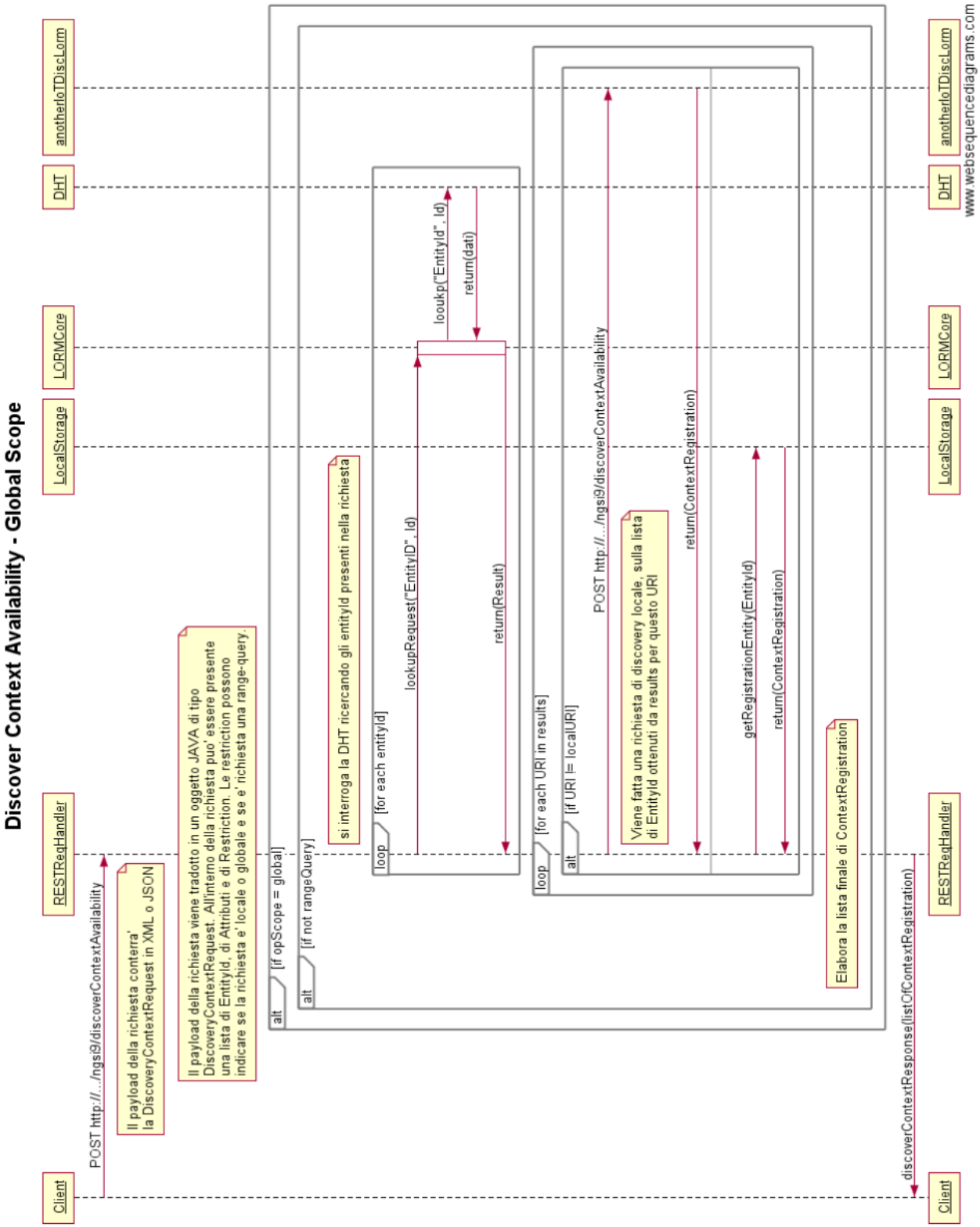


Figura 4.9: DiscoverContextAvailability - Global Scope

Discovery con Scope Globale e Range Query (Fig. 4.10) in cui è possibile eseguire delle interrogazioni su intervalli di valori di uno o più metadata associati ai loro attributi, in altre parole con questa restriction vengono implementate le range-query multiattributo. In questo caso il campo *scopeValue* di operationScope dovrà contenere le informazioni necessarie per eseguire la range query. Una range query restriction deve specificare l'attributo e il metadato associato e i valori dell'intervallo su cui dovrà essere fatta la query. Se è necessario eseguire una query su più coppie attributo-metadata, devono essere specificate più range-query restriction. Per ogni range-query il comportamento segue il caso illustrato precedentemente; viene fatta richiesta a LORM di eseguire una rangeLookup sul cluster identificato dalla coppia attributo-metadato, per l'intervallo di valori specificati nella restriction. Ogni rangeLookup restituirà un insieme di risultati, che al termine di tutte le range query, dovranno essere combinati tra loro secondo l'operatore logico AND, ottenendo l'insieme di entità che soddisfano la query di tipo range multi attributo. Ottenuto l'insieme di risultati finale, come nel caso precedente saranno contattate le varie istanze di *IoT-Discovery-Lorm* per ottenere le ContextRegistration da inviare al client. Nel caso in cui una delle range-lookup restituisca uno degli errori previsti, l'operazione di discovery sarà abortita, restituendo al client il motivo dell'errore.

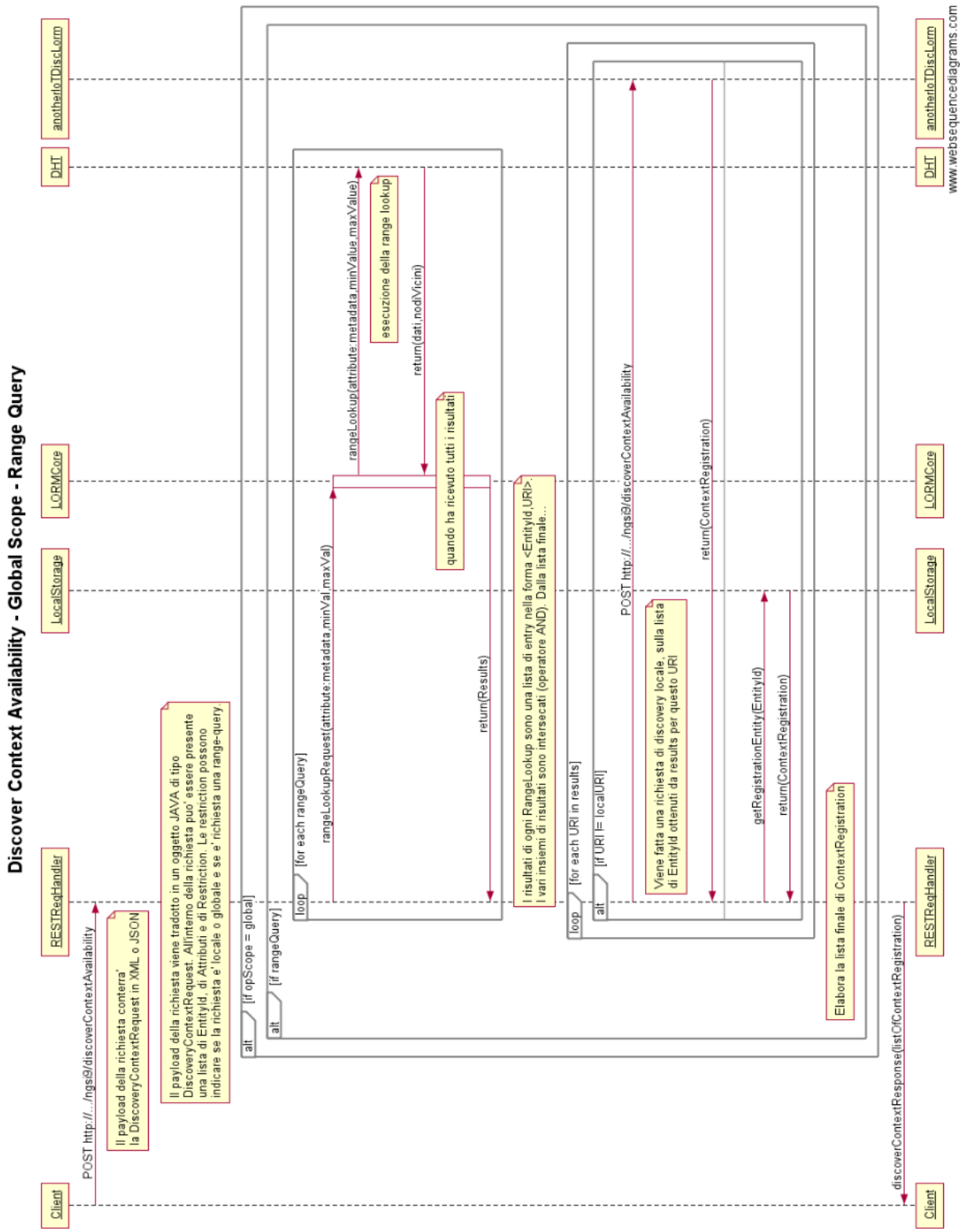


Figura 4.10: DiscoverContextAvailability - Global Scope - Range Query

Listing 4.5: Esempio di richiesta NGSI9 di DiscoveryContextAvailability in formato XML.

```
http://10.0.3.1:8080/ngsi9/discoverContextAvailability
<?xml version="1.0"?>
<discoverContextAvailabilityRequest>
       <entityIdList>
              <entityId type="" isPattern="true">
              <id>.*</id>
              </entityId>
       </entityIdList>
       <attributeList>
       </attributeList>
       <restriction>
              <attributeExpression></attributeExpression>
              <scope>
                     <operationScope>
                            <scopeType>discoveryType</scopeType>
                            <scopeValue>global</scopeValue>
                     </operationScope>
                     <operationScope>
                            <scopeType>RangeQuery</scopeType>
                            <scopeValue>
                                   <attribute>temperature</attribute>
                                   <metadata>longitude</metadata>
                                   <minValue>-20</minValue>
                                   <maxValue>30</maxValue>
                            </scopeValue>
                     </operationScope>
                     <operationScope>
                            <scopeType>RangeQuery</scopeType>
                            <scopeValue>
                                   <attribute>humidity</attribute>
                                   <metadata>longitude</metadata>
                                   <minValue>-89</minValue>
                                   <maxValue>40</maxValue>
                            </scopeValue>
                     </operationScope>
              </scope>
       </restriction>
</discoverContextAvailabilityRequest>
```

Capitolo 5

Test

La fase dei test è suddivisa in due parti. In primo luogo è stata effettuata una fase di validazione, in cui le funzionalità sviluppate in *IoT-Discovery-Lorm* vengono testate attraverso semplici casi d'uso. Nella seconda fase è stato verificato il corretto funzionamento di *IoT-Discovery-Lorm* nell'ecosistema IoT di FIWARE.

5.1 Test di Validazione

Il test di validazione è stato eseguito impostando un semplice scenario al fine di verificare il corretto comportamento delle funzionalità sviluppate in *IoT-Discovery-Lorm*. Lo scenario è stato impostato mettendo in esecuzione su 3 nodi fisici un'istanza di *IoT-Discovery-Lorm*, con la capacità di gestire risorse che possiedono come attributi *temperatura* e *umidità* a cui possono essere associati metadati relativi alla posizione geografica *latitudine* e *longitudine*. Le richieste NGSI-9 di *ContextRegistration* e di *Discovery* sono state simulate utilizzando un client REST, tramite il quale è stato possibile impostare nel payload la struttura XML opportuna.

Per l'analisi dei test di validazione si rimanda all'Appendice A.1

5.1.1 Sommario dei Test Case

LAB-TP-010-010 - Creazione di *IoT-Discovery-Lorm*

VALIDATION TEST REPORT				**Test Case ID**	LAB-TP-010-010		
Test Title	Creazione di *IoT-Discovery-Lorm*						
Test Input							
File di configurazione: web.xml File dei parametri della DHT: frepastry.params							
Result							
OK	X	NOT OK	.	Partially OK	.	Not Executed	.

LAB-TP-020-010 - ContextRegistration - EntityId - Attribute - No Metadata

VALIDATION TEST REPORT				**Test Case ID**	LAB-TP-020-010		
Test Title	ContextRegistration - EntityId - Attribute - No Metadata						
Test Input							
Richiesta NGSI9 tramite POST html - Payload: ContextRegistration in XML							
Result							
OK	X	NOT OK	.	Partially OK	.	Not Executed	.

LAB-TP-020-020 - ContextRegistration - EntityId - Attribute - Metadata

VALIDATION TEST REPORT				**Test Case ID**	LAB-TP-020-020		
Test Title	ContextRegistration - EntityId - Attribute - Metadata						
Test Input							
Richiesta NGSI9 tramite POST html - Payload: ContextRegistration in XML							
Result							
OK	X	NOT OK	.	Partially OK	.	Not Executed	.

LAB-TP-020-030 - ContextRegistration - Registrazione con EntityId già presente

VALIDATION TEST REPORT				**Test Case ID**	LAB-TP-020-030		
Test Title	ContextRegistration - Registrazione con EntityId già presente						
Test Input							
Richiesta NGSI9 tramite POST html - Payload: ContextRegistration in XML							
Result							
OK	X	NOT OK	.	Partially OK	.	Not Executed	.

LAB-TP-020-040 - ContextRegistration - Registrazione di una entità con attributo/metadato sconosciuto

VALIDATION TEST REPORT				**Test Case ID**	LAB-TP-020-040		
Test Title	ContextRegistration - Registrazione di una entità con attributo/metadato sconosciuto						
Test Input							
Richiesta NGSI9 tramite POST html - Payload: ContextRegistration in XML							
Result							
OK	X	NOT OK	.	Partially OK	.	Not Executed	.

LAB-TP-020-050 - ContextRegistration - Registrazione di una entità con valori del metadato OutOfRange

VALIDATION TEST REPORT				**Test Case ID**	LAB-TP-020-050		
Test Title	ContextRegistration - Registrazione di una entità con valori del metadato OutOfRange						
Test Input							
Richiesta NGSI9 tramite POST html - Payload: ContextRegistration in XML							
Result							
OK	X	NOT OK	.	Partially OK	.	Not Executed	.

LAB-TP-030-010 - Discovery - Lookup locale - EntityId specifico - Lista di attributi

VALIDATION TEST REPORT				**Test Case ID**	LAB-TP-030-010		
Test Title	Discovery - Lookup locale - EntityId specifico - Lista di attributi						
Test Input							
Richiesta NGSI9 tramite POST html - Payload: discoverContextAvailability in XML							
Result							
OK	X	NOT OK	.	Partially OK	.	Not Executed	.

LAB-TP-030-020 - Discovery - Lookup locale - Pattern - Lista di attributi

VALIDATION TEST REPORT				**Test Case ID**	LAB-TP-030-020		
Test Title	Discovery - Lookup locale - Pattern - Lista di attributi						
Test Input							
Richiesta NGSI9 tramite POST html - Payload: discoverContextAvailability in XML							
Result							
OK	X	NOT OK	.	Partially OK	.	Not Executed	.

LAB-TP-030-030 - Discovery - Lookup globale - EntityId Specifico - Nessuna Restriction

VALIDATION TEST REPORT				**Test Case ID**	LAB-TP-030-030		
Test Title	Discovery - Lookup globale - EntityId Specifico - Nessuna Restriction						
Test Input							
Richiesta NGSI9 tramite POST html - Payload: discoverContextAvailability in XML							
Result							
OK	X	NOT OK	.	Partially OK	.	Not Executed	.

LAB-TP-030-040 - Discovery - Lookup globale - Pattern - Range Query

VALIDATION TEST REPORT				**Test Case ID**	LAB-TP-030-040		
Test Title	Discovery - Lookup globale - Pattern - Range Query						
Test Input							
Richiesta NGSI9 tramite POST html - Payload: discoverContextAvailability in XML							
Result							
OK	X	NOT OK	.	Partially OK	.	Not Executed	.

LAB-TP-030-050 - Discovery - Lookup Globale - Range Query OutOfRange

VALIDATION TEST REPORT				**Test Case ID**	LAB-TP-030-050		
Test Title	Discovery - Lookup globale - Range Query OutOfRange						
Test Input							
Richiesta NGSI9 tramite POST html - Payload: discoverContextAvailability in XML							
Result							
OK	X	NOT OK	.	Partially OK	.	Not Executed	.

5.2 Test di uno scenario dimostrativo

In questa fase di testing è stato messo in opera il deployment dei Generic Enablers, necessari a realizzare uno scenario in cui dimostrare il corretto funzionamento di *IoT-Discovery-Lorm* nelle sue funzionalità e nella comunicazione con gli altri GE . Il GE realizzato deve ricevere e gestire le richieste di registrazione da parte di uno o più IoT Agent configurati per rivolgersi ad esso, come illustrato in figura 4.7. L'IoT Agent invia una *ContextRegistrationRequest* per ogni dispositivo per il quale riceve il *Device Provisioning*, attendendo in risposta l'esito della registrazione e l'identificativo di essa. Nel momento in cui un dispositivo viene effettivamente connesso all'Agent, sarà effettuato un update della registrazione utilizzando l'identificativo ottenuto nella fase precedente. Dal livello applicazione invece viene contattato l'IoT Broker per effettuare delle Query sui device. Il Broker richiederà tramite operazioni di Discovery verso *IoT-Discovery-Lorm*, le Context Information necessarie per identificare e raggiungere gli IoT Agent che possiedono i device che soddisfano la richiesta. Ottenute tali informazioni contatterà gli Agent coinvolti richiedendo operazioni di Query Context tramite l'interfaccia NGSI10 (Fig. 4.7).

In questo test si vuole inoltre dimostrare come l'utilizzo di un servizio di discovery distribuito, in cui vengono implementate operazioni range-query multi attributo, ottimizzi il processo descritto sopra rispetto ad un approccio che preveda più IoT Discovery centralizzati indipendenti tra loro.

5.2.1 Configurazione del Test

Tabella 5.1: Configurazione degli IoT Agent

IoT Agent Instance	**IoTDiscoveryLorm Instance**	**Device Id**	**Attributes**	**Metadata**
Agent1	IoTDiscoveryLorm1	sensor_i:agent1	temperature; humidity	latitude; longitude; accuracy; delaytime
Agent2	IoTDiscoveryLorm2	sensor_i:agent2	pressure; wind	latitude; longitude; accuracy; delaytime
Agent3	IoTDiscoveryLorm3	sensor_i:agent3	pollution;wind	latitude; longitude; accuracy; delaytime
Agent4	IoTDiscoveryLorm4	sensor_i:agent4	temperature; humidity	latitude; longitude; accuracy; delaytime
Agent5	IoTDiscoveryLorm1	sensor_i:agent5	pressure; wind	latitude; longitude; accuracy; delaytime
Agent6	IoTDiscoveryLorm2	sensor_i:agent6	pollution; temperature	latitude; longitude; accuracy; delaytime

Per verificare il funzionamento è stato realizzato uno scenario in cui i vari soggetti coinvolti fossero in uno stato consistente. Lo scenario prevede 4 istanze di *IoT-Discovery-Lorm* eseguite su altrettante macchine virtuali differenti operanti su un cluster basato su Openstack. Ogni istanza è stata configurata per gestire gli attributi e i metadata previsti dallo scenario, così che la DHT realizzata sia pronta a ricevere le richieste di inserimento e lookup. Per quanto riguarda gli IoT Agent sono state messe in esecuzione 6 istanze di *LWM2M-IoTAgent* fornite da FIWARE, su altrettante macchine virtuali appartenenti allo stesso cluster dei Discovery. Su queste istanze dovrà essere eseguito il device provisioning e il device registration per ogni device che si collegherà all'Agent (Par. 3.7.1). Per differenziare il tipo di Context Entities che ogni Agent registrerà ad ogni istanza è stata delegata la gestione di un tipo di device dotato di due risorse/attributi e quattro metadata.

Come si vede dalla tabella 5.1 e da Fig. 5.1 ogni IoT Agent è stato configurato per avere un solo IoT Discovery di riferimento su cui fare richieste di registrazione, da ciò ne consegue che in ogni istanza di *IoT-Discovery-Lorm* saranno registrate ContextRegistration diverse.

Per effettuare la fase di registrazione dei device sono stati realizzati degli script in Python in modo da automatizzare su ogni macchina virtuale il device provisioning. La generazione del valore dei metadata associati ad ogni device è random e per quanto riguarda latitudine e longitudine è stata ristretta ad un intervallo di valori delle coordinate tale da far risultare la posizione dei device nell'area corrispondente alla zona di Pisa. La fase di device registration prevede che venga effettuata la connessione del device all'IoT Agent. Non disponendo dei device fisici, LWM2M-IoTAgent mette a disposizione un client che simula tali

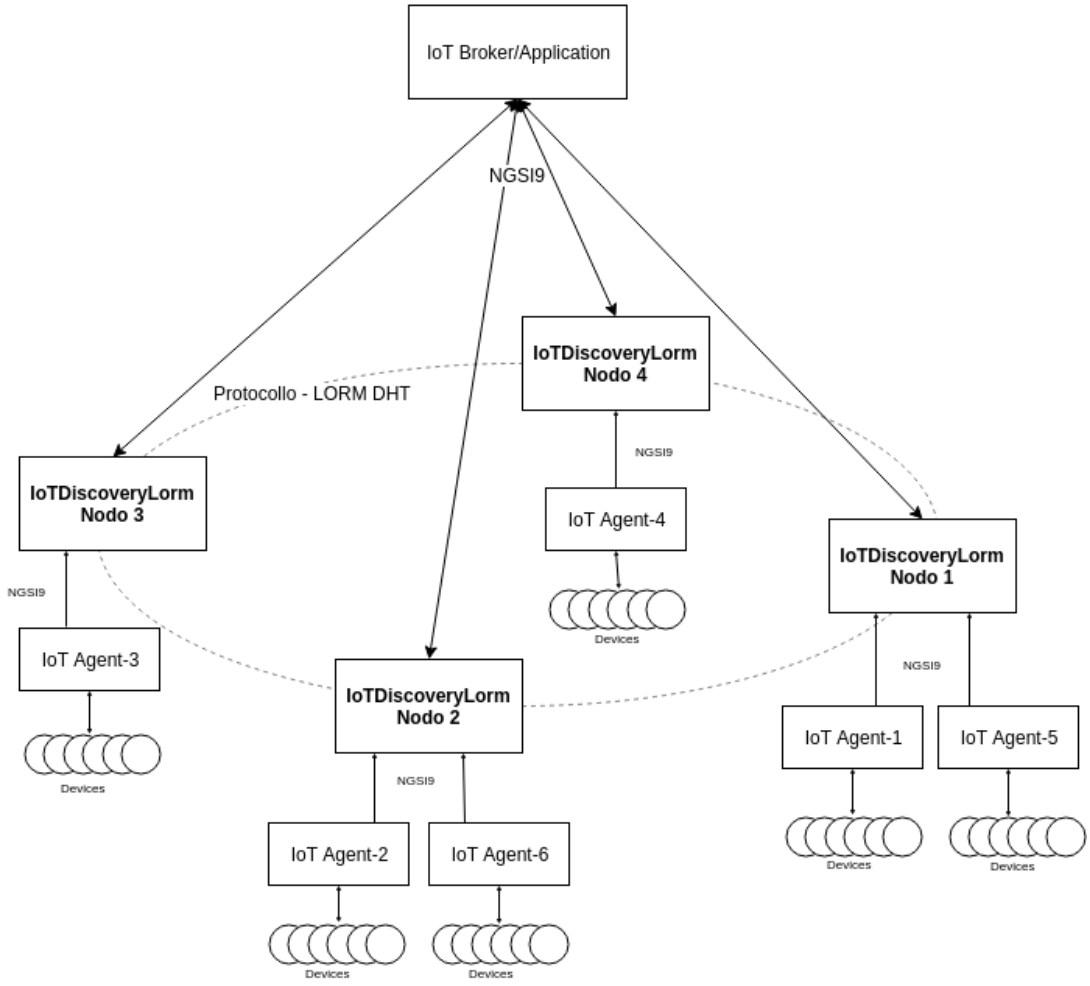


Figura 5.1: Test

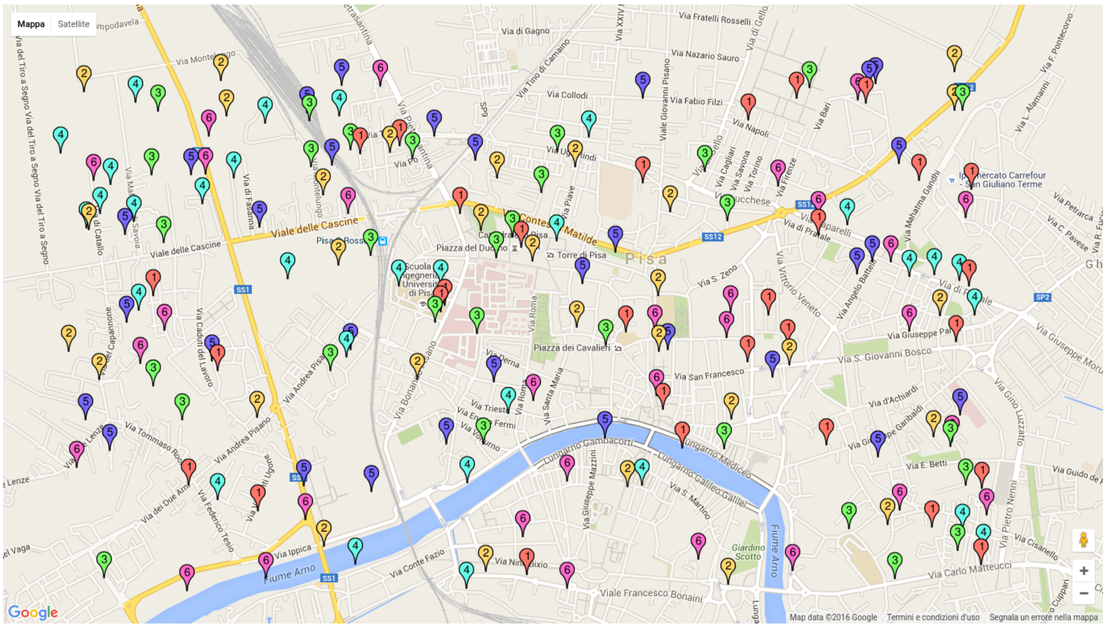


Figura 5.2: Mappa dei device

dispositivi e permette la loro registrazione e l'interazione con l'IoT Agent. Gli script sono stati configurati per registrare 30 device per IoT agent, per un totale di 180 device. Non è stato possibile registrarne di più perché per ogni device creato deve rimanere in esecuzione il client che lo simula, da ciò consegue che creando un numero maggiore di client l'utilizzo di tutta la memoria principale della macchina virtuale. Per dare riscontro visivo alla registrazione dei device di tutte le istanze dell'IoT Agent, effettuate verso i 4 *IoT-Discovery-Lorm* , è stata riportata su una mappa la posizione e l'IoT agent di appartenenza di ogni device (Fig. 5.2).

Range Query e Range Query Multiattributo

Per quanto riguarda le richieste provenienti dall'IoT Broker per operazioni di Discovery, sono state simulate utilizzando un Client REST che permetta di eseguire richieste di tipo Restful. In questo scenario sono state effettuate solamente richieste di discovery con scope Globale e Range Query. Altre forme di interrogazione rientrano nei casi d'uso usuali del Discovery e sono stati analizzati nei test di validazione.

Le richieste di *DiscoveryAvailability* sono state effettuate verso una delle quattro istanze di *IoT-Discovery-Lorm* senza che vi fosse una scelta dettata da una logica precisa. Questo per dimostrare che, chiunque sia il soggetto interessato da tale richiesta, il risultato resta immutato grazie all'implementazione

distribuita dell'IoT Discovery. A questo va aggiunto che le *ContextRegistration* ricevute in risposta non sono state necessariamente registrate nell'istanza di *IoT-Discovery-Lorm* che ha ricevuto la richiesta.

Qua sono riportati due esempi di Range Query. Nel primo caso è stata interrogata l'instanza *IoT-Discovery-Lorm*2 , richiedendo le ContextEntities che possiedono l'attributo *temperature* e che risiedono all'interno dell'area di Ingegneria. Come si vede da figura 5.3 in quell'area sono presenti 5 device, ma solo 4 di questi possiedono l'attributo temperatura e sono presenti nella risposta restituita da *IoT-Discovery-Lorm*. I due device dell'IoT Agent 1 erano stati registrati sul nodo 1, mentre i due device dell'IoT Agent 4 erano stati registrati sul nodo 4 dell'IoT Discovery. Nel secondo caso invece è stata interrogata l'istanza *IoT-Discovery-Lorm*1, richiedendo le ContextEntities che possiedono gli attributi *pressure* e *wind* in un area centrale di Pisa. In questo caso essendo presenti più attributi la risposta dovrà contenere quei device che soddisfano la range query e che possiedano *entrambi* gli attributi. In questo scenario i device che soddisfano questa query come si vede dalla figura 5.4, sono quelli registrati sugli IoT Agent 2 e 5, mentre i device che appartengono all'IoT Agent 3 non sono presi in considerazione perchè possiedono solo l'attributo *wind*. Nell'appendice A.2 sono stati riportati i risultati dettagliati delle due range query eseguite.

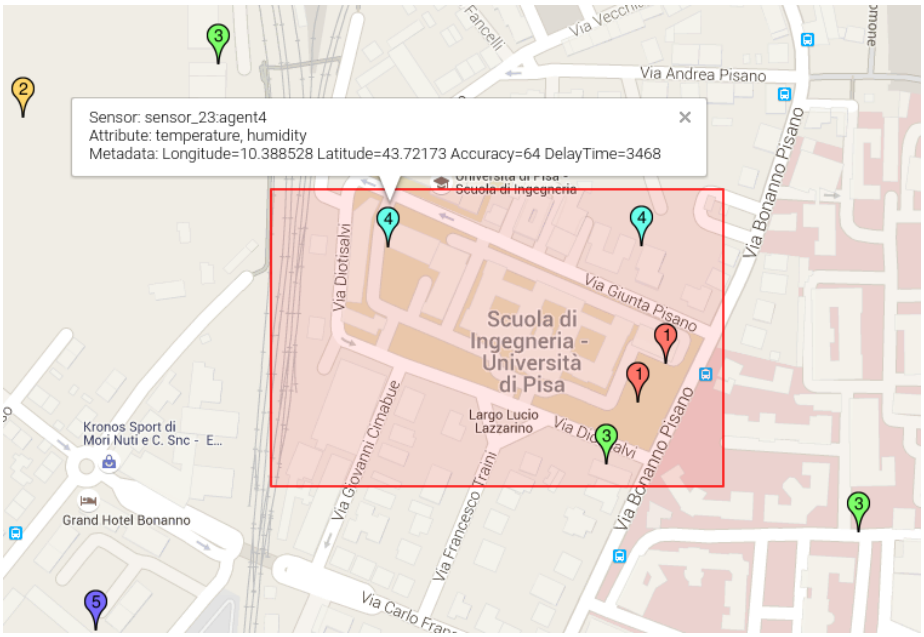


Figura 5.3: Range-Query su latitude e longitude per l'attributo temperature

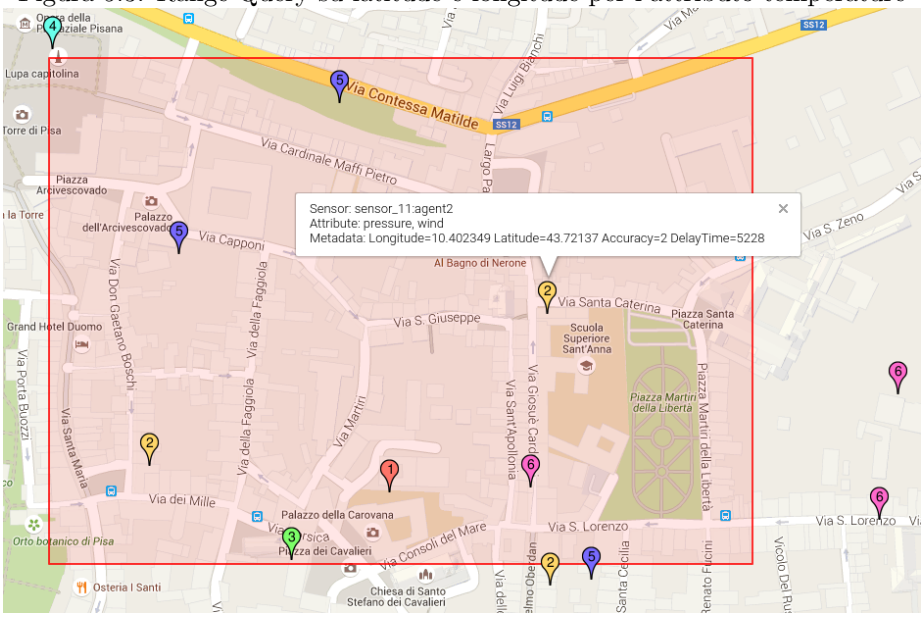


Figura 5.4: Range-query su latitude e longitude per gli attributi pressure e wind

5.2.2 Conclusioni sui test effettuati

Con questo test è stato dimostrato il corretto funzionamento di *IoT-Discovery-Lorm* in collaborazione con l'IoTAgent LWM2M fornito da FIWARE. I risultati ottenuti simulando il livello applicativo attraverso l'Advanced Client REST hanno restituito in seguito a richieste di Discovery con *scope globale* e di tipo *range*, le ContextRegistration previste e in formato conforme a quello previsto dallo standard NGSI9.

Analizzando lo scenario si possono trarre delle conclusioni anche in un'ottica di ottimizzazione del processo globale di interrogazione dei device da parte dell'IoT Broker. Confrontando lo stesso scenario utilizzando l'implementazione originale dell'IoT Discovery fornita da FIWARE piuttosto che la versione realizzata in questa tesi, avremmo avuto 4 istanze del discovery con repository centralizzato non comunicanti tra loro. Il Broker avrebbe dovuto eseguire 4 operazioni di discovery verso le 4 istanze per poter scoprire tutti i device che soddisfano i requisiti, prima di poter contattare gli IoT Agent interessati ed eseguire la Query. Inoltre l'implementazione dell'IoT Discovery GE non prevede la possibilità di eseguire le range-query, cio comporta che le ContextRegistration ottenute in risposta dai discovery non siano state filtrate. Eseguendo la stessa richiesta di discovery analizzata nel primo caso, il broker avrebbe ottenuto dai 4 IoT Discovery 90 ContextRegistration relative ai device con attributo temperatura. Per ottenere solo quelli dell'area evidenziata nell'esempio in fig. 5.3 l'Iot Broker dovrà analizzare l'insieme completo di risultati e filtrarli, prima di eseguire la Query nei confronti egli IoT Agent coinvolti.

Capitolo 6

Conclusioni

In questo lavoro è stato affrontato lo studio, la progettazione e l'implementazione di una soluzione per realizzare un servizio di discovery delle risorse IoT seguendo un approccio distribuito in modo da garantire scalabilità in un contesto come quello dell'Internet of Things, dove si ha a che fare con un numero molto elevato di dispositivi da identificare. Lo studio si è concentrato su architetture già esistenti basate sul paradigma del P2P-DHT. Utile allo scopo si è rivelato lo studio e l'implementazione di Lorm [7] che prevede una soluzione del servizio di discovery realizzata in modo distribuito attraverso una DHT a grafo composto, con la possibilità di eseguire delle lookup di tipo range multiattributo, mantenendo delle performance comparabili a quelle di DHT monoattributo e garantendo un alto livello di scalabilità e robustezza propria dei sistemi P2P-DHT [6]. In parallelo è stato oggetto di studio FIWARE che propone una piattaforma OpenSource per lo sviluppo di applicazioni per il Future Internet. L'attenzione principale è stata data ai servizi che offre nel contesto dell'Internet of Things. Il lavoro svolto si è concentrato nell'analizzare la struttura, il comportamento e le funzionalità del Generic Enablers che realizza il servizio di IoT Discovery, valutandone i possibili limiti e proponendo eventuali estensioni. In base a questo studio è emerso che il servizio di discovery realizzato da FIWARE segue un approccio centralizzato che non rispecchia le caratteristiche di scalabilità richieste in un contesto IoT dove sono presenti un grande numero di dispositivi distribuiti geograficamente in una area vasta.

La soluzione qua proposta ha avuto quindi lo scopo di estendere il Generic Enabler in modo che potesse operare in maniera distribuita sfruttando una rete globale come Internet. Per fare questo, come descritto nel capitolo 4, sono stati implementati all'interno di IoT Discovery i meccanismi e le operazioni della DHT *LORM*. L'integrazione delle nuove funzionalità è stata validata e testata come descritto nel capitolo 5, valutando il corretto funzionamento e i miglioramenti

che questa soluzione porta. Come lavoro futuro può essere interessante estendere la soluzione realizzata migliorando la struttura della DHT a grafo composto di LORM, in modo da poter implementare delle query con scope globale estese anche ai soli attributi di cui non è conosciuto il valore.

Capitolo 7

Ringraziamenti

Un grande ringraziamento al Professor Enzo Mingozzi per il supporto e la disponibilità costante e per i suoi consigli nello svolgere questo lavoro. Un ringraziamento altrettanto importante per Carlo Vallati e Giacomo Tanganelli per l'aiuto costante in questi mesi.

Un grazie speciale va a tutti i miei compagni di studi conosciuti in questi anni, per lo scambio di idee e per ogni progetto svolto insieme. Voglio ringraziare in particolare Gloria che mi ha sempre spinto a non mollare mai, sopportandomi e supportandomi nei momenti di difficoltà.

Infine il ringraziamento più importante alla mia famiglia che ha creduto in me, a mia madre e mio padre che mi hanno spinto a continuare anche dopo le delusioni dandomi fiducia e sicurezza, a mio fratello sempre pronto ad aiutarmi.

Appendice A

Test

A.1 Risultati dettagliati dei Test Case

LAB-TP-010-010 - Creazione di *IoT-Discovery-Lorm*

Dati in Input

La configurazione iniziale della DHT (compresi i nodi master da gestire) è contenuta nel file *freepastry.params*. I parametri della WebApplication sono contenuti nel file *web.xml*. I tre nodi fisici sono stati messi in esecuzione su 3 macchine virtuali, a cui sono stati associati gli indirizzi

- *10.0.3.1 - nodo di boot della DHT Lorm*
- *10.0.3.15*
- *10.0.3.133*

Risultati del test

Nodo 10.0.3.1 - Avvio della webapp e creazione interfaccia NGSI9, Creazione della DHT, Creazione nodi.

Questo nodo è il primo ad avviare l'istanza di *IoT-Discovery-Lorm*. Per permetterne la creazione della DHT di *Lorm* deve indicare se stesso come nodo di join.

```
1  repository_path in web.xml is valid
2  Opening Db4o database for Registration
3  /home/underhill84/Fiware-Lorm/ngsi9/repository/ngsiRegRepo.db4o
4  Opening Db4o database for Subscription
5  /home/underhill84/Fiware-Lorm/ngsi9/repository/ngsiSubRepo.db4o
6
7  ------------------------LORM MAIN LAUNCHED--------------------------------
8  Boot Node: /10.0.3.1:9001
9  Repository: /home/underhill84/Fiware-Lorm/lorm/repository/
10 Local Address: /10.0.3.1
```

```
11
12 Let's create the Manager Node
13 GENERATE_NODE_ID_FUNC: <0xD9F805163F032E6DDE1DB99E367A9D0AE03F911F..>
14 Storage Directory: /home/underhill84/Fiware-Lorm/lorm/repository/10.0.3.1/storage_Manager<0
       xD9F805163F032E6DDE1DB99E367A9D0AE03F911F..>
15
16 Creates local Nodes and verify if some master exist
17
18
19 Local Address: /10.0.3.1
20 Let's create a Virtual Node for Cluster: temperature:longitude
21 GENERATE_NODE_ID_FUNC: <0x65F0DE9349349DF03B00F52CE165685A499DFEA5..>
22 Storage Directory: /home/underhill84/Fiware-Lorm/lorm/repository/10.0.3.1/storage_Virtual_Node<0
       x65F0DE9349349DF03B00F52CE165685A499DFEA5..>
23 MASTER FOR temperature:longitude DOES NOT EXIST
24 LOCAL NODE FOR temperature:longitude DOES NOT EXIST
25 Connect to: /10.0.3.1
26 ::NODE BOOT from /10.0.3.1:9002
27
28
29 Local Address: /10.0.3.1
30 Let's create a Virtual Node for Cluster: humidity:latitude
31 GENERATE_NODE_ID_FUNC: <0x43B05096DF57C445B11330E16F9F7CF32C0BFE60..>
32 Storage Directory: /home/underhill84/Fiware-Lorm/lorm/repository/10.0.3.1/storage_Virtual_Node<0
       x43B05096DF57C445B11330E16F9F7CF32C0BFE60..>
33 MASTER FOR humidity:latitude DOES NOT EXIST
34 LOCAL NODE FOR humidity:latitude DOES NOT EXIST
35 Connect to: /10.0.3.1
36 ::NODE BOOT from /10.0.3.1:9003
37
38
39 Local Address: /10.0.3.1
40 Let's create a Virtual Node for Cluster: humidity:longitude
41 GENERATE_NODE_ID_FUNC: <0x247E0CBD8D8FC35832C6CB5C6DFE0559EAF4EDBC..>
42 Storage Directory: /home/underhill84/Fiware-Lorm/lorm/repository/10.0.3.1/storage_Virtual_Node<0
       x247E0CBD8D8FC35832C6CB5C6DFE0559EAF4EDBC..>
43 MASTER FOR humidity:longitude DOES NOT EXIST
44 LOCAL NODE FOR humidity:longitude DOES NOT EXIST
45 Connect to: /10.0.3.1
46 ::NODE BOOT from /10.0.3.1:9004
47
48
49 Local Address: /10.0.3.1
50 Let's create a Virtual Node for Cluster: entityid
51 GENERATE_NODE_ID_FUNC: <0xBA30A7FABBD3F3608D631401186DADA07733E977..>
52 Storage Directory: /home/underhill84/Fiware-Lorm/lorm/repository/10.0.3.1/storage_Virtual_Node<0
       xBA30A7FABBD3F3608D631401186DADA07733E977..>
53 MASTER FOR entityid DOES NOT EXIST
54 LOCAL NODE FOR entityid DOES NOT EXIST
55 Connect to: /10.0.3.1
56 ::NODE BOOT from /10.0.3.1:9005
57
58
59 Local Address: /10.0.3.1
60 Let's create a Virtual Node for Cluster: temperature:latitude
61 GENERATE_NODE_ID_FUNC: <0xE718E8B507CB3A90A487B42691137DCE79E4A76C..>
62 Storage Directory: /home/underhill84/Fiware-Lorm/lorm/repository/10.0.3.1/storage_Virtual_Node<0
       xE718E8B507CB3A90A487B42691137DCE79E4A76C..>
63 MASTER FOR temperature:latitude DOES NOT EXIST
64 LOCAL NODE FOR temperature:latitude DOES NOT EXIST
65 Connect to: /10.0.3.1
66 ::NODE BOOT from /10.0.3.1:9006
67
68 List of Master Node to be created: [temperature:longitude,-90,90, humidity:latitude,-180,180, humidity:longitude,-90,90, entityid
       , temperature:latitude,-180,180]
69
70
71 Let's create a master Node for attribute humidity:latitude
72
73 Let's create a master Node for attribute temperature:longitude
74
75 Let's create a master Node for attribute humidity:longitude
76
77 Let's create a master Node for attribute temperature:latitude
78 Local Address: /10.0.3.1
79 Local Address: /10.0.3.1
80 Local Address: /10.0.3.1
81 Storage Directory: /home/underhill84/Fiware-Lorm/lorm/repository/10.0.3.1/storage_Master<0
       x4E84F78C6A43198F333D7022B3D3648AB800A185..>
```

```
82
83  Let's create a master Node for attribute entityid
84  Storage Directory: /home/underhill84/Fiware-Lorm/lorm/repository/10.0.3.1/storage_Master<0
        xD2636C8B19D9F0DC5D38B440EB6F06939FA30812..>
85  Local Address: /10.0.3.1
86  ::MASTER_NODE for temperature:longitude BOOT from /10.0.3.1:9001
87  Storage Directory: /home/underhill84/Fiware-Lorm/lorm/repository/10.0.3.1/storage_Master<0
        xC1EC93536D47C16D704FFB1F4CF55294751EC969..>
88  Local Address: /10.0.3.1
89  Storage Directory: /home/underhill84/Fiware-Lorm/lorm/repository/10.0.3.1/storage_Master<0
        x601B99A4FBD2AFF864295381340C5D8CEE03A82B..>
90  ::MASTER_NODE for entityid BOOT from /10.0.3.1:9001
91  Storage Directory: /home/underhill84/Fiware-Lorm/lorm/repository/10.0.3.1/storage_Master<0
        xA13720FC21241F6C1E67B7342AEA2C3C379093DB..>
92  ::MASTER_NODE for humidity:longitude BOOT from /10.0.3.1:9001
93  ::MASTER_NODE for temperature:latitude BOOT from /10.0.3.1:9001
94  ::MASTER_NODE for humidity:latitude BOOT from /10.0.3.1:9001
95
96  Normal Node Created:
97  temperature:longitude 10.0.3.1 9002 -90 90 65F0DE9349349DF03B00F52CE165685A499DFEA5
98  humidity:latitude 10.0.3.1 9003 -180 180 43B05096DF57C445B11330E16F9F7CF32C0BFE60
99  humidity:longitude 10.0.3.1 9004 -90 90 247E0CBD8D8FC35832C6CB5C6DFE0559EAF4EDBC
100 entityid 10.0.3.1 9005 0 0 BA30A7FABBD3F3608D631401186DADA07733E977
101 temperature:latitude 10.0.3.1 9006 -180 180 E718E8B507CB3A90A487B42691137DCE79E4A76C
102
103 Master Created:
104 humidity:latitude,-180,180
105 temperature:longitude,-90,90
106 humidity:longitude,-90,90
107 temperature:latitude,-180,180
108 entityid
```

Nodo 10.0.3.15 e Nodo 10.0.3.133

Il comportamento di questi nodyi all'avvio è del tutto analogo al *nodo 10.0.3.1*. L'unica differenza è la mancata creazione di nodi master, essendo già stati creati dal primo nodo e quindi già presenti nell'anello superiore della DHT.

Gestione Eccezione - Nodo di boot alla DHT non raggiungibile

```
1   repository_path in web.xml is valid
2   Opening Db4o database for Registration
3   /home/underhill84/Fiware-Lorm/ngsi9/repository/ngsiRegRepo.db4o
4   Opening Db4o database for Subscription
5   /home/underhill84/Fiware-Lorm/ngsi9/repository/ngsiSubRepo.db4o
6
7   ------------------------LORM MAIN LAUNCHED---------------------------------
8   Boot Node: /10.0.3.2:9001
9   Repository: /home/underhill84/Fiware-Lorm/lorm/repository/
10  Local Address: /10.0.3.1
11
12  Let's create the Manager Node
13  GENERATE_NODE_ID_FUNC: <0x28DB3518D8F567B21960534ADC4A3834D933A5A8..>
14  Storage Directory: /home/underhill84/Fiware-Lorm/lorm/repository/10.0.3.1/storage_Manager<0
        x28DB3518D8F567B21960534ADC4A3834D933A5A8..>
15  0x28DB35:rice.pastry:20160519.111817.753:joinFailed(rice.pastry.JoinFailedException: Cannot join ring. All bootstraps are faulty
        .[/10.0.3.2:9001])
16  Exception occurred in LormMain
```

LAB-TP-020-010 - ContextRegistration - EntityId - Attribute - No Metadata

Dati in Input

```
POST http://10.0.3.1:8080/ngsi9/registerContext
Payload:
<?xml version="1.0"?>
<registerContextRequest>
  <contextRegistrationList>
    <contextRegistration>
      <entityIdList>
        <entityId type="robot" isPattern="false">
          <id>Robot10</id>
        </entityId>
        <entityId type="robot" isPattern="false">
          <id>Robot11</id>
        </entityId>
      </entityIdList>
      <contextRegistrationAttributeList>
        <contextRegistrationAttribute>
          <name>temperature</name>
          <type>integer</type>
          <isDomain>false</isDomain>
        </contextRegistrationAttribute>
        <contextRegistrationAttribute>
          <name>humidity</name>
          <type>integer</type>
          <isDomain>false</isDomain>
        </contextRegistrationAttribute>
      </contextRegistrationAttributeList>
      <providingApplication>131.114.59.22/robots</providingApplication>
  </contextRegistration>
  </contextRegistrationList>
  <duration>P1M</duration>
</registerContextRequest>
```

Risultati del test

Nodo 10.0.3.1 - Nodo che riceve la richiesta di ContextRegistration NGSI-9

La richiesta di *ContextRegistration* è analizzata e tradotta dal *Marshaller*. Le informazioni rilevanti della ContextRegistration, in questo caso *EntityId*, vengono inserite della DHT di *Lorm*. Dal listato si può ricavare che in questo nodo fisico, viene salvata la entry relativa a *[Robot10;robot:10.0.3.1]*, il cui indice è dato dall'hash *23E1411F302BAEF20D64B4B2A17640253946B6B0* è numericamente più vicino al nodo vituale di indice *52A18F3EC5EE26223EEB19D527CC44C6F3F739E7* posseduto da questo nodo fisico

```
Marshalled XML Request
Registration ID not provided
Deleting possible duplicate registration

Load for node 52A18F3EC5EE26223EEB19D527CC44C6F3F739E7 = 0
cluster: entityid
content:[Robot10;robot:10.0.3.1] hash:[<0x23E1411F302BAEF20D64B4B2A17640253946B6B0..>]
 successfully stored at 1 locations.
Inserted!

------->Store 23E1411F302BAEF20D64B4B2A17640253946B6B0 to 52A18F3EC5EE26223EEB19D527CC44C6F3F739E7

INSERT LEAFSET -> [[SNH: <0xAEC79CCDAB4E7F5FA223F75D239AC83D27633955..>//10.0.3.15:9003], [SNH: <0
        x69441A38EABA7A1BC80300260869DE5C1E6D9F62..>//10.0.3.133:9002]]
cluster: entityid
content:[Robot11;robot:10.0.3.1] hash:[<0x6AB70A0313AF526E1EF131949703AA13D720EC84..>]
 successfully stored at 1 locations.
Inserted!
```

Nodo 10.0.3.133 - Nodo remoto appartentente alla DHT

In questo nodo viene salvata la entry relativa a *[Robot11;robot:10.0.3.1]*, il cui indice è dato dall'hash *6AB70A0313AF526E1EF131949703AA13D720EC84* è numericamente più vicino al nodo vituale di indice *69441A38EABA7A1BC80300260869DE5C1E6D9F62* posseduto da questo nodo fisico

```
Load for node 69441A38EABA7A1BC80300260869DE5C1E6D9F62 = 0
------->Store 6AB70A0313AF526E1EF131949703AA13D720EC84 to 69441A38EABA7A1BC80300260869DE5C1E6D9F62
```

Risposta NGSI9 al richiedente

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<registerContextResponse>
       <duration>P1M</duration>
       <registrationId>UniS_KJGTOOeV3m</registrationId>
       <errorCode>
              <code>200</code>
              <reasonPhrase>OK</reasonPhrase>
              <details xsi:type="xs:string">Stored</details>
       </errorCode>
</registerContextResponse>
```

LAB-TP-020-020 - ContextRegistration - EntityId - Attribute - Metadata

Dati in Input

```
POST http://10.0.3.1:8080/ngsi9/registerContext
Payload:
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<registerContextRequest>
   <contextRegistrationList>
      <contextRegistration>
         <entityIdList>
            <entityId type="robot" isPattern="false">
               <id>Robot1</id>
            </entityId>
         </entityIdList>
         <contextRegistrationAttributeList>
            <contextRegistrationAttribute>
               <name>temperature</name>
               <type>integer</type>
               <isDomain>false</isDomain>
               <metadata>
                  <contextMetadata>
                     <name>longitude</name>
                     <type>integer</type>
                     <value>25</value>
                  </contextMetadata>
               </metadata>
            </contextRegistrationAttribute>
            <contextRegistrationAttribute>
               <name>humidity</name>
               <type>integer</type>
               <isDomain>false</isDomain>
               <metadata>
                  <contextMetadata>
                     <name>longitude</name>
                     <type>integer</type>
                     <value>35</value>
                  </contextMetadata>
               </metadata>
            </contextRegistrationAttribute>
         </contextRegistrationAttributeList>
```

```
38            <providingApplication>131.114.59.22/robots</providingApplication>
39         </contextRegistration>
40         <contextRegistration>
41            <entityIdList>
42               <entityId type="robot" isPattern="false">
43                  <id>Robot2</id>
44               </entityId>
45            </entityIdList>
46            <contextRegistrationAttributeList>
47               <contextRegistrationAttribute>
48                  <name>temperature</name>
49                  <type>integer</type>
50                  <isDomain>false</isDomain>
51                  <metadata>
52                     <contextMetadata>
53                        <name>longitude</name>
54                        <type>integer</type>
55                        <value>25</value>
56                     </contextMetadata>
57                  </metadata>
58               </contextRegistrationAttribute>
59               <contextRegistrationAttribute>
60                  <name>humidity</name>
61                  <type>integer</type>
62                  <isDomain>false</isDomain>
63                  <metadata>
64                     <contextMetadata>
65                        <name>longitude</name>
66                        <type>integer</type>
67                        <value>60</value>
68                     </contextMetadata>
69                  </metadata>
70               </contextRegistrationAttribute>
71            </contextRegistrationAttributeList>
72            <providingApplication>131.114.59.22/robots</providingApplication>
73         </contextRegistration>
74         <contextRegistration>
75            <entityIdList>
76               <entityId type="robot" isPattern="false">
77                  <id>Robot3</id>
78               </entityId>
79            </entityIdList>
80            <contextRegistrationAttributeList>
81               <contextRegistrationAttribute>
82                  <name>temperature</name>
83                  <type>integer</type>
84                  <isDomain>false</isDomain>
85                  <metadata>
86                     <contextMetadata>
87                        <name>longitude</name>
88                        <type>integer</type>
89                        <value>60</value>
90                     </contextMetadata>
91                  </metadata>
92               </contextRegistrationAttribute>
93               <contextRegistrationAttribute>
94                  <name>humidity</name>
95                  <type>integer</type>
96                  <isDomain>false</isDomain>
97                  <metadata>
98                     <contextMetadata>
99                        <name>longitude</name>
100                       <type>integer</type>
101                       <value>35</value>
102                    </contextMetadata>
103                 </metadata>
104              </contextRegistrationAttribute>
105           </contextRegistrationAttributeList>
106           <providingApplication>131.114.59.22/robots</providingApplication>
107        </contextRegistration>
108        <contextRegistration>
109           <entityIdList>
110              <entityId type="robot" isPattern="false">
111                 <id>Robot4</id>
112              </entityId>
113           </entityIdList>
114           <contextRegistrationAttributeList>
115              <contextRegistrationAttribute>
116                 <name>temperature</name>
```

```
                <type>integer</type>
                <isDomain>false</isDomain>
                <metadata>
                    <contextMetadata>
                        <name>longitude</name>
                        <type>integer</type>
                        <value>7</value>
                    </contextMetadata>
                </metadata>
            </contextRegistrationAttribute>
            <contextRegistrationAttribute>
                <name>humidity</name>
                <type>integer</type>
                <isDomain>false</isDomain>
                <metadata>
                    <contextMetadata>
                        <name>longitude</name>
                        <type>integer</type>
                        <value>4</value>
                    </contextMetadata>
                </metadata>
            </contextRegistrationAttribute>
        </contextRegistrationAttributeList>
        <providingApplication>131.114.59.22/robots</providingApplication>
    </contextRegistration>
    <contextRegistration>
        <entityIdList>
            <entityId type="robot" isPattern="false">
                <id>Robot5</id>
            </entityId>
        </entityIdList>
        <contextRegistrationAttributeList>
            <contextRegistrationAttribute>
                <name>temperature</name>
                <type>integer</type>
                <isDomain>false</isDomain>
                <metadata>
                    <contextMetadata>
                        <name>longitude</name>
                        <type>integer</type>
                        <value>-15</value>
                    </contextMetadata>
                </metadata>
            </contextRegistrationAttribute>
            <contextRegistrationAttribute>
                <name>humidity</name>
                <type>integer</type>
                <isDomain>false</isDomain>
                <metadata>
                    <contextMetadata>
                        <name>longitude</name>
                        <type>integer</type>
                        <value>-85</value>
                    </contextMetadata>
                </metadata>
            </contextRegistrationAttribute>
        </contextRegistrationAttributeList>
        <providingApplication>131.114.59.22/robots</providingApplication>
    </contextRegistration>
    <contextRegistration>
        <entityIdList>
            <entityId type="robot" isPattern="false">
                <id>Robot6</id>
            </entityId>
        </entityIdList>
        <contextRegistrationAttributeList>
            <contextRegistrationAttribute>
                <name>temperature</name>
                <type>integer</type>
                <isDomain>false</isDomain>
                <metadata>
                    <contextMetadata>
                        <name>longitude</name>
                        <type>integer</type>
                        <value>45</value>
                    </contextMetadata>
                </metadata>
            </contextRegistrationAttribute>
            <contextRegistrationAttribute>
```

```
                <name>humidity</name>
                <type>integer</type>
                <isDomain>false</isDomain>
                <metadata>
                    <contextMetadata>
                        <name>longitude</name>
                        <type>integer</type>
                        <value>38</value>
                    </contextMetadata>
                </metadata>
            </contextRegistrationAttribute>
        </contextRegistrationAttributeList>
        <providingApplication>131.114.59.22/robots</providingApplication>
    </contextRegistration>
  </contextRegistrationList>
  <duration>P1M</duration>
</registerContextRequest>
```

Risultati del test

Nodo 10.0.3.1 - Nodo che riceve la richiesta di ContextRegistration NGSI-9

La richiesta di *ContextRegistration* è analizzata e tradotta dal *Marshaller*. Le informazioni rilevanti della ContextRegistration sono *EntityId* e il valore dei metadata associati agli attributi. Saranno questi ad essere inseriti nella DHT di *Lorm*.

```
Marshalled XML Request

Registration ID not provided
Deleting possible duplicate registration
INSERT LEAFSET -> [[SNH: <0xAEC79CCDAB4E7F5FA223F75D239AC83D27633955..>//10.0.3.15:9003], [SNH: <0
        x69441A38EABA7A1BC80300260869DE5C1E6D9F62..>//10.0.3.133:9002]]
cluster: entityid
content:[Robot1;robot:10.0.3.1] hash:[<0x711E95ADF27D8D9320C4174329FEACB7DC2BDE73..>]
 successfully stored at 1 locations.
Inserted!

INSERT LEAFSET -> [[SNH: <0x97A80A1CD6727211CA4B5413A18B43CE7F315752..>//10.0.3.133:9004], [SNH: <0
        x378D1A82A81FFD95291D4BDFA17ABFD57A0328FE..>//10.0.3.15:9004]]
cluster: temperature:longitude
content:[Robot1;robot:10.0.3.1] hash:[<0xA38E38E38E38E38E38E38E38E38E38E38E384..>]
 successfully stored at 1 locations.
Inserted!

cluster: humidity:longitude
content:[Robot1;robot:10.0.3.1] hash:[<0xB1C71C71C71C71C71C71C71C71C71C71BC..>]
successfully stored at 1 locations.
Inserted!

INSERT LEAFSET -> [[SNH: <0xAEC79CCDAB4E7F5FA223F75D239AC83D27633955..>//10.0.3.15:9003], [SNH: <0
        x69441A38EABA7A1BC80300260869DE5C1E6D9F62..>//10.0.3.133:9002]]
cluster: entityid
content:[Robot2;robot:10.0.3.1] hash:[<0xD61E31A2C263B692C6982A2E8534E451E3579909..>]
 successfully stored at 1 locations.
Inserted!

INSERT LEAFSET -> [[SNH: <0x97A80A1CD6727211CA4B5413A18B43CE7F315752..>//10.0.3.133:9004], [SNH: <0
        x378D1A82A81FFD95291D4BDFA17ABFD57A0328FE..>//10.0.3.15:9004]]
cluster: temperature:longitude
content:[Robot2;robot:10.0.3.1] hash:[<0xA38E38E38E38E38E38E38E38E38E38E38E384..>]
 successfully stored at 1 locations.
Inserted!

cluster: humidity:longitude
content:[Robot2;robot:10.0.3.1] hash:[<0xD55555555555555555555555555555555548..>]
successfully stored at 1 locations.
Inserted!

```

```
39 INSERT LEAFSET -> [[SNH: <0xAEC79CCDAB4E7F5FA223F75D239AC83D27633955..>//10.0.3.15:9003], [SNH: <0
       x69441A38EABA7A1BC80300260869DE5C1E6D9F62..>//10.0.3.133:9002]]
40 cluster: entityid
41 content:[Robot3;robot:10.0.3.1] hash:[<0xE13443552D21699A189A9DC4DF18473540D7BB75..>]
42  successfully stored at 1 locations.
43 Inserted!
44
45 INSERT LEAFSET -> [[SNH: <0x97A80A1CD6727211CA4B5413A18B43CE7F315752..>//10.0.3.133:9004], [SNH: <0
       x378D1A82A81FFD95291D4BDFA17ABFD57A0328FE..>//10.0.3.15:9004]]
46 Load for node C2F034DBCA2C746AF2D71E09797D3569C62F15C5 = 0
47 cluster: temperature:longitude
48 content:[Robot3;robot:10.0.3.1] hash:[<0xD5555555555555555555555555555555555555548..>]
49  successfully stored at 1 locations.
50 Inserted!
51
52 ------->Store D5555555555555555555555555555555555548 to C2F034DBCA2C746AF2D71E09797D3569C62F15C5
53 cluster: humidity:longitude
54 content:[Robot3;robot:10.0.3.1] hash:[<0xB1C71C71C71C71C71C71C71C71C71C71C71BC..>]
55 successfully stored at 1 locations.
56 Inserted!
57
58 INSERT LEAFSET -> [[SNH: <0xAEC79CCDAB4E7F5FA223F75D239AC83D27633955..>//10.0.3.15:9003], [SNH: <0
       x69441A38EABA7A1BC80300260869DE5C1E6D9F62..>//10.0.3.133:9002]]
59 cluster: entityid
60 content:[Robot4;robot:10.0.3.1] hash:[<0x87A19BEB7EE255A59CC01ACF83692422991953F9..>]
61  successfully stored at 1 locations.
62 Inserted!
63
64 INSERT LEAFSET -> [[SNH: <0x97A80A1CD6727211CA4B5413A18B43CE7F315752..>//10.0.3.133:9004], [SNH: <0
       x378D1A82A81FFD95291D4BDFA17ABFD57A0328FE..>//10.0.3.15:9004]]
65 cluster: temperature:longitude
66 content:[Robot4;robot:10.0.3.1] hash:[<0x89F49F49F49F49F49F49F49F49F49F49F49EC..>]
67  successfully stored at 1 locations.
68 Inserted!
69
70 cluster: humidity:longitude
71 content:[Robot4;robot:10.0.3.1] hash:[<0x85B05B05B05B05B05B05B05B05B05B05B05A8..>]
72 successfully stored at 1 locations.
73 Inserted!
74
75 INSERT LEAFSET -> [[SNH: <0xAEC79CCDAB4E7F5FA223F75D239AC83D27633955..>//10.0.3.15:9003], [SNH: <0
       x69441A38EABA7A1BC80300260869DE5C1E6D9F62..>//10.0.3.133:9002]]
76 cluster: entityid
77 content:[Robot5;robot:10.0.3.1] hash:[<0x9A5AABA582AED417BAC486D2C4BB8D74C730C977..>]
78  successfully stored at 1 locations.
79 Inserted!
80
81 INSERT LEAFSET -> [[SNH: <0x97A80A1CD6727211CA4B5413A18B43CE7F315752..>//10.0.3.133:9004], [SNH: <0
       x378D1A82A81FFD95291D4BDFA17ABFD57A0328FE..>//10.0.3.15:9004]]
82 cluster: temperature:longitude
83 content:[Robot5;robot:10.0.3.1] hash:[<0x6AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA4..>]
84  successfully stored at 1 locations.
85 Inserted!
86
87 cluster: humidity:longitude
88 content:[Robot5;robot:10.0.3.1] hash:[<0x071C71C71C71C71C71C71C71C71C71C71C71C..>]
89 successfully stored at 1 locations.
90 Inserted!
91
92 INSERT LEAFSET -> [[SNH: <0xAEC79CCDAB4E7F5FA223F75D239AC83D27633955..>//10.0.3.15:9003], [SNH: <0
       x69441A38EABA7A1BC80300260869DE5C1E6D9F62..>//10.0.3.133:9002]]
93 cluster: entityid
94 content:[Robot6;robot:10.0.3.1] hash:[<0x660E2EC4882692185 83C88EF93B41F43A5D5203C..>]
95  successfully stored at 1 locations.
96 Inserted!
97
98 INSERT LEAFSET -> [[SNH: <0x97A80A1CD6727211CA4B5413A18B43CE7F315752..>//10.0.3.133:9004], [SNH: <0
       x378D1A82A81FFD95291D4BDFA17ABFD57A0328FE..>//10.0.3.15:9004]]
99 Load for node C2F034DBCA2C746AF2D71E09797D3569C62F15C5 = 1
100 cluster: temperature:longitude
101 content:[Robot6;robot:10.0.3.1] hash:[<0xBFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF4..>]
102  successfully stored at 1 locations.
103 Inserted!
104
105 ------->Store BFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF4 to C2F034DBCA2C746AF2D71E09797D3569C62F15C5
106 cluster: humidity:longitude
107 content:[Robot6;robot:10.0.3.1] hash:[<0xB60B60B60B60B60B60B60B60B60B60B60B600..>]
108 successfully stored at 1 locations.
109 Inserted!
```

Nodo 10.0.3.15 - Nodo remoto appartentente alla DHT

Questo nodo gestisce i nodi master relativi a *temperature:latitude* e *humidity:longitude*. Come si vede dal listato gestisce la richiesta di inserimento ricevuta dal **nodo 10.0.3.1**.

```
INSERT 35 Robot1;robot:10.0.3.1 <0xB1C71C71C71C71C71C71C71C71C71C71C71BC..>
INSERT LEAFSET -> [[SNH: <0xD711D710E801592EB6157596EF00886BF21474DB..>//10.0.3.133:9006]]
cluster: humidity:longitude
content:[Robot1;robot:10.0.3.1] hash:[<0xB1C71C71C71C71C71C71C71C71C71C71C71BC..>]
successfully stored at 1 locations.
Load for node AEC79CCDAB4E7F5FA223F75D239AC83D27633955 = 0
------->Store D61E31A2C263B692C6982A2E8534E451E3579909 to AEC79CCDAB4E7F5FA223F75D239AC83D27633955

INSERT 60 Robot2;robot:10.0.3.1 <0xD55555555555555555555555555555555555548..>
INSERT LEAFSET -> [[SNH: <0xD711D710E801592EB6157596EF00886BF21474DB..>//10.0.3.133:9006]]
cluster: humidity:longitude
content:[Robot2;robot:10.0.3.1] hash:[<0xD55555555555555555555555555555555555548..>]
successfully stored at 1 locations.
Load for node AEC79CCDAB4E7F5FA223F75D239AC83D27633955 = 1
------->Store E13443552D21699A189A9DC4DF18473540D7BB75 to AEC79CCDAB4E7F5FA223F75D239AC83D27633955

INSERT 35 Robot3;robot:10.0.3.1 <0xB1C71C71C71C71C71C71C71C71C71C71C71BC..>
INSERT LEAFSET -> [[SNH: <0xD711D710E801592EB6157596EF00886BF21474DB..>//10.0.3.133:9006]]
cluster: humidity:longitude
content:[Robot3;robot:10.0.3.1] hash:[<0xB1C71C71C71C71C71C71C71C71C71C71C71BC..>]
successfully stored at 1 locations.

INSERT 4 Robot4;robot:10.0.3.1 <0x85B05B05B05B05B05B05B05B05B05B05B05A8..>
INSERT LEAFSET -> [[SNH: <0xD711D710E801592EB6157596EF00886BF21474DB..>//10.0.3.133:9006]]
Load for node 4AFFCB5523C823082E830E38100CDC1AAFCC0791 = 0
cluster: humidity:longitude
content:[Robot4;robot:10.0.3.1] hash:[<0x85B05B05B05B05B05B05B05B05B05B05B05A8..>]
successfully stored at 1 locations.
------->Store 85B05B05B05B05B05B05B05B05B05B05B05A8 to 4AFFCB5523C823082E830E38100CDC1AAFCC0791
Load for node AEC79CCDAB4E7F5FA223F75D239AC83D27633955 = 2
------->Store 9A5AABA582AED417BAC486D2C4BB8D74C730C977 to AEC79CCDAB4E7F5FA223F75D239AC83D27633955

INSERT -85 Robot5;robot:10.0.3.1 <0x071C71C71C71C71C71C71C71C71C71C71C71C..>
INSERT LEAFSET -> [[SNH: <0xD711D710E801592EB6157596EF00886BF21474DB..>//10.0.3.133:9006]]
cluster: humidity:longitude
content:[Robot5;robot:10.0.3.1] hash:[<0x071C71C71C71C71C71C71C71C71C71C71C71C..>]
successfully stored at 1 locations.

INSERT 38 Robot6;robot:10.0.3.1 <0xB60B60B60B60B60B60B60B60B60B60B60B60B600..>
INSERT LEAFSET -> [[SNH: <0xD711D710E801592EB6157596EF00886BF21474DB..>//10.0.3.133:9006]]
cluster: humidity:longitude
content:[Robot6;robot:10.0.3.1] hash:[<0xB60B60B60B60B60B60B60B60B60B60B60B60B600..>]
successfully stored at 1 locations.
```

Nodo 10.0.3.133 - Nodo remoto appartentente alla DHT

Questo nodo possiede solo *nodi Virtuali* nei quali vengono salvate le entry.

```
Load for node 69441A38EABA7A1BC80300260869DE5C1E6D9F62 = 1
------->Store 711E95ADF27D8D9320C4174329FEACB7DC2BDE73 to 69441A38EABA7A1BC80300260869DE5C1E6D9F62
Load for node 97A80A1CD6727211CA4B5413A18B43CE7F315752 = 0
------->Store A38E38E38E38E38E38E38E38E38E38E38E38E384 to 97A80A1CD6727211CA4B5413A18B43CE7F315752
Load for node D711D710E801592EB6157596EF00886BF21474DB = 0
------->Store B1C71C71C71C71C71C71C71C71C71C71C71BC to D711D710E801592EB6157596EF00886BF21474DB
Load for node 97A80A1CD6727211CA4B5413A18B43CE7F315752 = 1
20: Value already inserted, appending new content to old one
------->Store A38E38E38E38E38E38E38E38E38E38E38E38E384 to 97A80A1CD6727211CA4B5413A18B43CE7F315752
Load for node D711D710E801592EB6157596EF00886BF21474DB = 1
------->Store D55555555555555555555555555555555555548 to D711D710E801592EB6157596EF00886BF21474DB
Load for node D711D710E801592EB6157596EF00886BF21474DB = 2
20: Value already inserted, appending new content to old one
------->Store B1C71C71C71C71C71C71C71C71C71C71C71BC to D711D710E801592EB6157596EF00886BF21474DB
Load for node 69441A38EABA7A1BC80300260869DE5C1E6D9F62 = 2
------->Store 87A19BEB7EE255A59CC01ACF836924229919 53F9 to 69441A38EABA7A1BC80300260869DE5C1E6D9F62
Load for node 97A80A1CD6727211CA4B5413A18B43CE7F315752 = 1
------->Store 89F49F49F49F49F49F49F49F49F49F49F49EC to 97A80A1CD6727211CA4B5413A18B43CE7F315752
```

```
Load for node 97A80A1CD6727211CA4B5413A18B43CE7F315752 = 2
------->Store 6AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA4 to 97A80A1CD6727211CA4B5413A18B43CE7F315752
Load for node D711D710E801592EB6157596EF00886BF21474DB = 2
------->Store 071C71C71C71C71C71C71C71C71C71C71C71C71C to D711D710E801592EB6157596EF00886BF21474DB
Load for node 69441A38EABA7A1BC80300260869DE5C1E6D9F62 = 3
------->Store 660E2EC488269218583C88EF93B41F43A5D5203C to 69441A38EABA7A1BC80300260869DE5C1E6D9F62
Load for node D711D710E801592EB6157596EF00886BF21474DB = 3
------->Store B60B60B60B60B60B60B60B60B60B60B60B60B600 to D711D710E801592EB6157596EF00886BF21474DB
```

LAB-TP-020-030 - ContextRegistration - Registrazione con EntityId già presente

Dati in Input

```
POST http://10.0.3.1:8080/ngsi9/registerContext
Payload:
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<registerContextRequest>
    <contextRegistrationList>
        <contextRegistration>
            <entityIdList>
                <entityId type="robot" isPattern="false">
                    <id>Robot1</id>
                </entityId>
                <entityId type="robot" isPattern="false">
                    <id>Robot2</id>
                </entityId>
            </entityIdList>
            <contextRegistrationAttributeList>
                <contextRegistrationAttribute>
                    <name>temperature</name>
                    <type>integer</type>
                    <isDomain>false</isDomain>
                    <metadata>
                        <contextMetadata>
                            <name>longitude</name>
                            <type>integer</type>
                            <value>20</value>
                        </contextMetadata>
                    </metadata>
                </contextRegistrationAttribute>
            </contextRegistrationAttributeList>
            <providingApplication>http://mysensors.com/Rooms</providingApplication>
        </contextRegistration>
    </contextRegistrationList>
    <duration>P1M</duration>
</registerContextRequest>
```

Risultati del test

Nodo 10.0.3.1 - Nodo che riceve la richiesta di ContextRegistration NGSI-9

Ad ogni richiesta di registrazione ricevuta, viene preventivamente controllato se gli EntityId contenuti nella richiesta siano già presenti nello storage locale. In tal caso le informazioni associate vengono sovrascritte sia localmente, sia nella DHT.

```
Marshalled XML Request:

Registration ID not provided
Deleting possible duplicate registration
cluster: entityid
content:[Robot1;robot:10.0.3.1] hash:[<0x711E95ADF27D8D9320C4174329FEACB7DC2BDE73..>]
```

```
 successfully removed at 1 locations.
Removed!

cluster: temperature:longitude
content:[Robot1;robot:10.0.3.1] hash:[<0x9C71C71C71C71C71C71C71C71C71C71C71C68..>]
 successfully removed at 1 locations.
Removed!

INSERT LEAFSET -> [[SNH: <0xD8BE4A245EC9E51C5592E356CE23B5E1211164E8..>//10.0.3.15:9002]]
INSERT LEAFSET -> [[SNH: <0xC703A0D0F613AD4C8227DDB58656971C476EB7A0..>//10.0.3.15:9006]]
cluster: entityid
content:[Robot1;robot:10.0.3.1] hash:[<0x711E95ADF27D8D9320C4174329FEACB7DC2BDE73..>]
 successfully stored at 1 locations.
Inserted!

cluster: entityid
content:[Robot2;robot:10.0.3.1] hash:[<0xD61E31A2C263B692C6982A2E8534E451E3579909..>]
 successfully removed at 1 locations.
Removed!

cluster: temperature:longitude
content:[Robot2;robot:10.0.3.1] hash:[<0x9C71C71C71C71C71C71C71C71C71C71C71C68..>]
 successfully removed at 1 locations.
Removed!

cluster: temperature:longitude
content:[Robot1;robot:10.0.3.1] hash:[<0x9C71C71C71C71C71C71C71C71C71C71C71C68..>]
 successfully stored at 1 locations.
Inserted!

INSERT LEAFSET -> [[SNH: <0xD8BE4A245EC9E51C5592E356CE23B5E1211164E8..>//10.0.3.15:9002]]
INSERT LEAFSET -> [[SNH: <0xC703A0D0F613AD4C8227DDB58656971C476EB7A0..>//10.0.3.15:9006]]
cluster: entityid
content:[Robot2;robot:10.0.3.1] hash:[<0xD61E31A2C263B692C6982A2E8534E451E3579909..>]
 successfully stored at 1 locations.
Inserted!

cluster: temperature:longitude
content:[Robot2;robot:10.0.3.1] hash:[<0x9C71C71C71C71C71C71C71C71C71C71C71C68..>]
 successfully stored at 1 locations.
Inserted!
```

Nodo remoto appartentente alla DHT

```
Load for node D8BE4A245EC9E51C5592E356CE23B5E1211164E8 = 2
Load for node C703A0D0F613AD4C8227DDB58656971C476EB7A0 = 1
------->Remove 711E95ADF27D8D9320C4174329FEACB7DC2BDE73 to D8BE4A245EC9E51C5592E356CE23B5E1211164E8
------->Remove 9C71C71C71C71C71C71C71C71C71C71C71C68 to C703A0D0F613AD4C8227DDB58656971C476EB7A0
Load for node D8BE4A245EC9E51C5592E356CE23B5E1211164E8 = 1
Load for node D8BE4A245EC9E51C5592E356CE23B5E1211164E8 = 1
Load for node C703A0D0F613AD4C8227DDB58656971C476EB7A0 = 1
------->Remove D61E31A2C263B692C6982A2E8534E451E3579909 to D8BE4A245EC9E51C5592E356CE23B5E1211164E8
------->Store 711E95ADF27D8D9320C4174329FEACB7DC2BDE73 to D8BE4A245EC9E51C5592E356CE23B5E1211164E8
------->Remove 9C71C71C71C71C71C71C71C71C71C71C71C68 to C703A0D0F613AD4C8227DDB58656971C476EB7A0
Load for node C703A0D0F613AD4C8227DDB58656971C476EB7A0 = 0
------->Store 9C71C71C71C71C71C71C71C71C71C71C71C68 to C703A0D0F613AD4C8227DDB58656971C476EB7A0
Load for node C703A0D0F613AD4C8227DDB58656971C476EB7A0 = 1
Load for node D8BE4A245EC9E51C5592E356CE23B5E1211164E8 = 1
20: Value already inserted, appending new content to old one
------->Store D61E31A2C263B692C6982A2E8534E451E3579909 to D8BE4A245EC9E51C5592E356CE23B5E1211164E8
------->Store 9C71C71C71C71C71C71C71C71C71C71C71C68 to C703A0D0F613AD4C8227DDB58656971C476EB7A0
```

Risposta NGSI9 al richiedente

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<registerContextResponse>
       <duration>P1M</duration>
       <registrationId>UniS_XNOvaReZeN</registrationId>
       <errorCode>
              <code>200</code>
              <reasonPhrase>OK</reasonPhrase>
```

```
            <details xsi:type="xs:string">Stored</details>
      </errorCode>
</registerContextResponse>
```

LAB-TP-020-040 - ContextRegistration - Registrazione di una entità con attributo/metadato sconosciuto

La gestione di questa eccezione si ha nel caso in cui nessuna istanza di *IoT-Discovery-Lorm*, abbia creato un nodo master relativo alla coppia attributo-metadato che si tenta di registrare, in fase di inizializzazione del sistema.

Dati in Input

```
POST http://10.0.3.1:8080/ngsi9/registerContext
Payload:
<?xml version="1.0"?>
<registerContextRequest>
  <contextRegistrationList>
    <contextRegistration>
      <entityIdList>
        <entityId type="robot" isPattern="false">
          <id>Robot1</id>
        </entityId>
        <entityId type="robot" isPattern="false">
          <id>Robot2</id>
        </entityId>
      </entityIdList>
      <contextRegistrationAttributeList>
        <contextRegistrationAttribute>
          <name>temperature</name>
          <type>integer</type>
          <isDomain>false</isDomain>
          <metadata>
               <contextMetadata>
              <name>accuracy</name>
              <type>integer</type>
              <value>20</value>
            </contextMetadata>
          </metadata>
        </contextRegistrationAttribute>
      </contextRegistrationAttributeList>
      <providingApplication>http://mysensors.com/Rooms</providingApplication>
  </contextRegistration>
  </contextRegistrationList>
  <duration>P1M</duration>
</registerContextRequest>
```

Risultati del test

Nodo che riceve la richiesta di ContextRegistration NGSI-9

```
Registration ID not provided
Deleting possible duplicate registration
INSERT LEAFSET -> [[SNH: <0xB44B7F6B153F7C51BF228BAEAFC1360B12CFBA68..>//10.0.3.15:9004]]
Load for node 4DA3FBCD51C9A665A5FD5CD5EC2E1C94F4C0C011 = 0

temperature:accuracy Attribute unknown
temperature:accuracy Attribute unknown
Exception during Insert Request. temperature:accuracy Attribute unknown
Abort registration
Load for node 4DA3FBCD51C9A665A5FD5CD5EC2E1C94F4C0C011 = 0


temperature:accuracy Attribute unknown
```

```
temperature:accuracy Attribute unknown
cluster: entityid
content:[Robot2;robot:10.0.3.1] hash:[<0xD61E31A2C263B692C6982A2E8534E451E3579909..>]
 successfully removed at 0 locations.
ERROR: Data not exists or not all instance have been removed
cluster: entityid
content:[Robot1;robot:10.0.3.1] hash:[<0x711E95ADF27D8D9320C4174329FEACB7DC2BDE73..>]
 successfully stored at 1 locations.
Inserted!

------->Store 711E95ADF27D8D9320C4174329FEACB7DC2BDE73 to 4DA3FBCD51C9A665A5FD5CD5EC2E1C94F4C0C011
temperature:accuracy Attribute unknown
temperature:accuracy Attribute unknown
------->Remove 711E95ADF27D8D9320C4174329FEACB7DC2BDE73 to 4DA3FBCD51C9A665A5FD5CD5EC2E1C94F4C0C011
cluster: entityid
content:[Robot1;robot:10.0.3.1] hash:[<0x711E95ADF27D8D9320C4174329FEACB7DC2BDE73..>]
 successfully removed at 1 locations.
Removed!
```

Nodo remoto appartentente alla DHT

```
Load for node B44B7F6B153F7C51BF228BAEAFC1360B12CFBA68 = 0
```

Risposta NGSI9 al richiedente

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<registerContextResponse>
       <registrationId />
       <errorCode>
              <code>404</code>
              <reasonPhrase>temperature:accuracy Attribute unknown</reasonPhrase>
              <details xsi:type="xs:string" />
       </errorCode>
</registerContextResponse>
```

LAB-TP-020-050 - ContextRegistration - Registrazione di una entità con valori del metadato OutOfRange

La gestione di questa eccezione si ha nel caso in cui si tenta di inserire una ContextRegistration che ha come valore di un metadato associato ad un attributo, un valore fuori dal range specificato in occasione dell'inizializzazione del sistema.

Dati in Input

```
POST http://10.0.3.1:8080/ngsi9/registerContext
Payload:
<?xml version="1.0"?>
<registerContextRequest>
  <contextRegistrationList>
    <contextRegistration>
      <entityIdList>
        <entityId type="robot" isPattern="false">
          <id>Robot1</id>
        </entityId>
        <entityId type="robot" isPattern="false">
          <id>Robot2</id>
        </entityId>
      </entityIdList>
      <contextRegistrationAttributeList>
```

```
16        <contextRegistrationAttribute>
17          <name>temperature</name>
18          <type>integer</type>
19          <isDomain>false</isDomain>
20          <metadata>
21               <contextMetadata>
22              <name>longitude</name>
23              <type>integer</type>
24              <value>180</value>
25            </contextMetadata>
26          </metadata>
27        </contextRegistrationAttribute>
28      </contextRegistrationAttributeList>
29      <providingApplication>http://mysensors.com/Rooms</providingApplication>
30    </contextRegistration>
31    </contextRegistrationList>
32    <duration>P1M</duration>
33  </registerContextRequest>
```

Risultati del test

Nodo che riceve la richiesta di ContextRegistration NGSI-9

```
1  Registration ID not provided
2  Deleting possible duplicate registration
3  INSERT LEAFSET -> []
4  Load for node 564C8DEDBB4885FF0C1CB2DA8490F1998CAE74D5 = 0
5  cluster: entityid
6  content:[Robot1;robot:10.0.3.1] hash:[<0x711E95ADF27D8D9320C4174329FEACB7DC2BDE73..>]
7   successfully stored at 1 locations.
8  Inserted!
9
10 Value out of range: Range -90 to 90
11 Exception during Insert Request. Value out of range: Range -90 to 90
12 Abort registration
13 ------->Store 711E95ADF27D8D9320C4174329FEACB7DC2BDE73 to 564C8DEDBB4885FF0C1CB2DA8490F1998CAE74D5
14
15 Value out of range: Range -90 to 90
16 Load for node 564C8DEDBB4885FF0C1CB2DA8490F1998CAE74D5 = 1
17 Load for node 564C8DEDBB4885FF0C1CB2DA8490F1998CAE74D5 = 1
18 cluster: entityid
19 content:[Robot2;robot:10.0.3.1] hash:[<0xD61E31A2C263B692C6982A2E8534E451E3579909..>]
20  successfully removed at 0 locations.
21 ERROR: Data not exists or not all instance have been removed
22 ------->Remove 711E95ADF27D8D9320C4174329FEACB7DC2BDE73 to 564C8DEDBB4885FF0C1CB2DA8490F1998CAE74D5
23 cluster: entityid
24 content:[Robot1;robot:10.0.3.1] hash:[<0x711E95ADF27D8D9320C4174329FEACB7DC2BDE73..>]
25  successfully removed at 1 locations.
26 Removed!
```

Risposta NGSI9 al richiedente

```
1  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2  <registerContextResponse>
3      <registrationId></registrationId>
4      <errorCode>
5          <code>404</code>
6          <reasonPhrase>Value out of range: Range -90 to 90</reasonPhrase>
7          <details xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
               instance"></details>
8      </errorCode>
9  </registerContextResponse>
```

LAB-TP-030-010 - Discovery - Lookup locale - EntityId specifico - Lista di attributi

La richiesta di Discovery con *scope* locale, interroga il *Local Storage* dell'IoT Discovery, alla ricerca delle ContextRegistration in possesso delle informazioni richieste.

Dati in Input

```
<?xml version="1.0"?>
<discoverContextAvailabilityRequest>
  <entityIdList>
    <entityId type="robot" isPattern="false">
      <id>Robot1</id>
    </entityId>
  </entityIdList>
  <attributeList>
  </attributeList>
  <restriction>
    <attributeExpression></attributeExpression>
    <scope>
      <operationScope>
        <scopeType>discoveryType</scopeType>
        <scopeValue>local</scopeValue>
      </operationScope>
    </scope>
  </restriction>
</discoverContextAvailabilityRequest>
```

Risultati del test

Nodo 10.0.3.1 - Nodo che riceve la richiesta di ContextRegistration NGSI-9

```
Content-Type is: xml
Accept: *
```

Risposta NGSI9 al richiedente

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<discoverContextAvailabilityResponse>
    <contextRegistrationResponseList>
        <contextRegistrationResponse>
            <contextRegistration>
                <entityIdList>
                    <entityId type="robot" isPattern="false">
                        <id>Robot1</id>
                    </entityId>
                </entityIdList>
                <contextRegistrationAttributeList>
                    <contextRegistrationAttribute>
                        <name>temperature</name>
                        <type>integer</type>
                        <isDomain>false</isDomain>
                        <metadata>
                            <contextMetadata>
                                <name>longitude</name>
                                <type>integer</type>
                                <value>25</value>
                            </contextMetadata>
                        </metadata>
                    </contextRegistrationAttribute>
```

```
                <contextRegistrationAttribute>
                    <name>humidity</name>
                    <type>integer</type>
                    <isDomain>false</isDomain>
                    <metadata>
                        <contextMetadata>
                            <name>longitude</name>
                            <type>integer</type>
                            <value>35</value>
                        </contextMetadata>
                    </metadata>
                </contextRegistrationAttribute>
            </contextRegistrationAttributeList>
            <providingApplication>131.114.59.22/robots</providingApplication>
        </contextRegistration>
    </contextRegistrationResponse>
</contextRegistrationResponseList>
<errorCode>
    <code>200</code>
    <reasonPhrase>OK</reasonPhrase>
    <details xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
        instance">Result</details>
</errorCode>
</discoverContextAvailabilityResponse>
```

LAB-TP-030-020 - Discovery - Lookup locale - Pattern - Lista di attributi

La richiesta di Discovery con *scope* locale, interroga il *Local Storage* dell'IoT Discovery, alla ricerca delle ContextRegistration in possesso delle informazioni richieste.

Dati in Input

```
<?xml version="1.0" encoding="UTF-8"?>
<discoverContextAvailabilityRequest>
    <entityIdList>
        <entityId type="robot" isPattern="true">
            <id>.*</id>
        </entityId>
    </entityIdList>
    <attributeList>
        <attribute>humidity</attribute>
        <attribute>temperature</attribute>
    </attributeList>
    <restriction>
        <attributeExpression></attributeExpression>
        <scope>
            <operationScope>
                <scopeType></scopeType>
                <scopeValue></scopeValue>
            </operationScope>
            <operationScope>
                <scopeType></scopeType>
                <scopeValue></scopeValue>
            </operationScope>
        </scope>
    </restriction>
</discoverContextAvailabilityRequest>
```

Risultati del test

Nodo 10.0.3.1 - Nodo che riceve la richiesta di ContextRegistration NGSI-9

```
Content-Type is: xml
Accept: *
```

Risposta NGSI9 al richiedente

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<discoverContextAvailabilityResponse>
    <contextRegistrationResponseList>
        <contextRegistrationResponse>
            <contextRegistration>
                <entityIdList>
                    <entityId type="robot" isPattern="false">
                        <id>Robot10</id>
                    </entityId>
                    <entityId type="robot" isPattern="false">
                        <id>Robot11</id>
                    </entityId>
                </entityIdList>
                <contextRegistrationAttributeList>
                    <contextRegistrationAttribute>
                        <name>temperature</name>
                        <type>integer</type>
                        <isDomain>false</isDomain>
                        <metadata/>
                    </contextRegistrationAttribute>
                    <contextRegistrationAttribute>
                        <name>humidity</name>
                        <type>integer</type>
                        <isDomain>false</isDomain>
                        <metadata/>
                    </contextRegistrationAttribute>
                </contextRegistrationAttributeList>
                <providingApplication>131.114.59.22/robots</providingApplication>
            </contextRegistration>
        </contextRegistrationResponse>
        <contextRegistrationResponse>
            <contextRegistration>
                <entityIdList>
                    <entityId type="robot" isPattern="false">
                        <id>Robot1</id>
                    </entityId>
                </entityIdList>
                <contextRegistrationAttributeList>
                    <contextRegistrationAttribute>
                        <name>temperature</name>
                        <type>integer</type>
                        <isDomain>false</isDomain>
                        <metadata>
                            <contextMetadata>
                                <name>longitude</name>
                                <type>integer</type>
                                <value>25</value>
                            </contextMetadata>
                        </metadata>
                    </contextRegistrationAttribute>
                    <contextRegistrationAttribute>
                        <name>humidity</name>
                        <type>integer</type>
                        <isDomain>false</isDomain>
                        <metadata>
                            <contextMetadata>
                                <name>longitude</name>
                                <type>integer</type>
                                <value>35</value>
                            </contextMetadata>
                        </metadata>
                    </contextRegistrationAttribute>
                </contextRegistrationAttributeList>
                <providingApplication>131.114.59.22/robots</providingApplication>
            </contextRegistration>
        </contextRegistrationResponse>
        <contextRegistrationResponse>
            <contextRegistration>
                <entityIdList>
                    <entityId type="robot" isPattern="false">
```

```
                    <id>Robot2</id>
                </entityId>
            </entityIdList>
            <contextRegistrationAttributeList>
                <contextRegistrationAttribute>
                    <name>temperature</name>
                    <type>integer</type>
                    <isDomain>false</isDomain>
                    <metadata>
                        <contextMetadata>
                            <name>longitude</name>
                            <type>integer</type>
                            <value>25</value>
                        </contextMetadata>
                    </metadata>
                </contextRegistrationAttribute>
                <contextRegistrationAttribute>
                    <name>humidity</name>
                    <type>integer</type>
                    <isDomain>false</isDomain>
                    <metadata>
                        <contextMetadata>
                            <name>longitude</name>
                            <type>integer</type>
                            <value>60</value>
                        </contextMetadata>
                    </metadata>
                </contextRegistrationAttribute>
            </contextRegistrationAttributeList>
            <providingApplication>131.114.59.22/robots</providingApplication>
        </contextRegistration>
    </contextRegistrationResponse>
    <contextRegistrationResponse>
        <contextRegistration>
            <entityIdList>
                <entityId type="robot" isPattern="false">
                    <id>Robot3</id>
                </entityId>
            </entityIdList>
            <contextRegistrationAttributeList>
                <contextRegistrationAttribute>
                    <name>temperature</name>
                    <type>integer</type>
                    <isDomain>false</isDomain>
                    <metadata>
                        <contextMetadata>
                            <name>longitude</name>
                            <type>integer</type>
                            <value>60</value>
                        </contextMetadata>
                    </metadata>
                </contextRegistrationAttribute>
                <contextRegistrationAttribute>
                    <name>humidity</name>
                    <type>integer</type>
                    <isDomain>false</isDomain>
                    <metadata>
                        <contextMetadata>
                            <name>longitude</name>
                            <type>integer</type>
                            <value>35</value>
                        </contextMetadata>
                    </metadata>
                </contextRegistrationAttribute>
            </contextRegistrationAttributeList>
            <providingApplication>131.114.59.22/robots</providingApplication>
        </contextRegistration>
    </contextRegistrationResponse>
    <contextRegistrationResponse>
        <contextRegistration>
            <entityIdList>
                <entityId type="robot" isPattern="false">
                    <id>Robot4</id>
                </entityId>
            </entityIdList>
            <contextRegistrationAttributeList>
                <contextRegistrationAttribute>
                    <name>temperature</name>
                    <type>integer</type>
```

```
                    <isDomain>false</isDomain>
                    <metadata>
                        <contextMetadata>
                            <name>longitude</name>
                            <type>integer</type>
                            <value>7</value>
                        </contextMetadata>
                    </metadata>
                </contextRegistrationAttribute>
                <contextRegistrationAttribute>
                    <name>humidity</name>
                    <type>integer</type>
                    <isDomain>false</isDomain>
                    <metadata>
                        <contextMetadata>
                            <name>longitude</name>
                            <type>integer</type>
                            <value>4</value>
                        </contextMetadata>
                    </metadata>
                </contextRegistrationAttribute>
            </contextRegistrationAttributeList>
            <providingApplication>131.114.59.22/robots</providingApplication>
        </contextRegistration>
    </contextRegistrationResponse>
    <contextRegistrationResponse>
        <contextRegistration>
            <entityIdList>
                <entityId type="robot" isPattern="false">
                    <id>Robot5</id>
                </entityId>
            </entityIdList>
            <contextRegistrationAttributeList>
                <contextRegistrationAttribute>
                    <name>temperature</name>
                    <type>integer</type>
                    <isDomain>false</isDomain>
                    <metadata>
                        <contextMetadata>
                            <name>longitude</name>
                            <type>integer</type>
                            <value>-15</value>
                        </contextMetadata>
                    </metadata>
                </contextRegistrationAttribute>
                <contextRegistrationAttribute>
                    <name>humidity</name>
                    <type>integer</type>
                    <isDomain>false</isDomain>
                    <metadata>
                        <contextMetadata>
                            <name>longitude</name>
                            <type>integer</type>
                            <value>-85</value>
                        </contextMetadata>
                    </metadata>
                </contextRegistrationAttribute>
            </contextRegistrationAttributeList>
            <providingApplication>131.114.59.22/robots</providingApplication>
        </contextRegistration>
    </contextRegistrationResponse>
    <contextRegistrationResponse>
        <contextRegistration>
            <entityIdList>
                <entityId type="robot" isPattern="false">
                    <id>Robot6</id>
                </entityId>
            </entityIdList>
            <contextRegistrationAttributeList>
                <contextRegistrationAttribute>
                    <name>temperature</name>
                    <type>integer</type>
                    <isDomain>false</isDomain>
                    <metadata>
                        <contextMetadata>
                            <name>longitude</name>
                            <type>integer</type>
                            <value>45</value>
                        </contextMetadata>
```

```
                    </metadata>
                </contextRegistrationAttribute>
                <contextRegistrationAttribute>
                    <name>humidity</name>
                    <type>integer</type>
                    <isDomain>false</isDomain>
                    <metadata>
                        <contextMetadata>
                            <name>longitude</name>
                            <type>integer</type>
                            <value>38</value>
                        </contextMetadata>
                    </metadata>
                </contextRegistrationAttribute>
            </contextRegistrationAttributeList>
            <providingApplication>131.114.59.22/robots</providingApplication>
        </contextRegistration>
    </contextRegistrationResponse>
  </contextRegistrationResponseList>
  <errorCode>
    <code>200</code>
    <reasonPhrase>OK</reasonPhrase>
    <details xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
        instance">Result</details>
  </errorCode>
</discoverContextAvailabilityResponse>
```

LAB-TP-030-030 - Discovery - Lookup globale - EntityId Specifico - Nessuna Restriction

La richiesta di Discovery con *scope* globale, interroga in primo luogo la DHT con una lookup per ogni *EntityId* presente nella richiesta. Per ogni risultato restituito si ottiene l'indirizzo relativo all'istanza di *IoT-Discovery-Lorm* che possiede tale *ContextRegistration* alla quale verrà richiesta.

Dati in Input

```
POST http://10.0.3.15:8080/ngsi9/discoverContextAvailability
Payload:
<?xml version="1.0"?>
<discoverContextAvailabilityRequest>
  <entityIdList>
    <entityId type="robot" isPattern="false">
      <id>Robot1</id>
    </entityId>
    <entityId type="robot" isPattern="false">
      <id>Robot3</id>
    </entityId>
    <entityId type="robot" isPattern="false">
      <id>Robot6</id>
    </entityId>
  </entityIdList>
  <attributeList>
  </attributeList>
  <restriction>
    <attributeExpression></attributeExpression>
    <scope>
      <operationScope>
        <scopeType>discoveryType</scopeType>
        <scopeValue>global</scopeValue>
      </operationScope>
    </scope>
  </restriction>
</discoverContextAvailabilityRequest>
```

Risultati del test

Nodo 10.0.3.15 - Nodo che riceve la richiesta di Discovery NGSI-9

```
INFORMAZIONI: 2016-05-20 14:28:11 10.0.3.1 - 10.0.3.15 8080 POST /ngsi9/discoverContextAvailability - 200 5456 682 5112 http
       ://10.0.3.15:8080 - -
Content-Type is: xml
Accept: *

Global Discovery Request

Successfully looked up
cluster: entityid
content:[Robot1;robot:10.0.3.1] hash:[<0x711E95ADF27D8D9320C4174329FEACB7DC2BDE73..>]
for key <0x711E95ADF27D8D9320C4174329FEACB7DC2BDE73..>.
[Robot1;robot:10.0.3.1]
EntityId Query: [Robot1;robot:10.0.3.1]

Successfully looked up
cluster: entityid
content:[Robot3;robot:10.0.3.1] hash:[<0xE13443552D21699A189A9DC4DF18473540D7BB75..>]
for key <0xE13443552D21699A189A9DC4DF18473540D7BB75..>.
[Robot3;robot:10.0.3.1]
EntityId Query: [Robot3;robot:10.0.3.1]

Successfully looked up
cluster: entityid
content:[Robot6;robot:10.0.3.1] hash:[<0x660E2EC488269218583C88EF93B41F43A5D5203C..>]
for key <0x660E2EC488269218583C88EF93B41F43A5D5203C..>.
[Robot6;robot:10.0.3.1]
EntityId Query: [Robot6;robot:10.0.3.1]

Final result:
 {10.0.3.1=[Robot1;robot, Robot3;robot, Robot6;robot]}
```

Nodo 10.0.3.1 - Nodo remoto appartentente alla DHT

Questo nodo riceve da parte di **10.0.3.15** una richiesta di discovery locale per ContextRegistration relative alle entità *[Robot1;robot, Robot3;robot, Robot6;robot]*

```
Content-Type is: xml
Accept: *
```

Risposta NGSI9 al richiedente

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<discoverContextAvailabilityResponse>
       <contextRegistrationResponseList>
              <contextRegistrationResponse>
                     <contextRegistration>
                                  <entityIdList>
                                         <entityId type="robot" isPattern="false">
                                                <id>Robot1</id>
                                         </entityId>
                                  </entityIdList>
                                  <contextRegistrationAttributeList>
                                         <contextRegistrationAttribute>
                                                <name>temperature</name>
                                                <type>integer</type>
                                                <isDomain>false</isDomain>
                                                <metadata>
                                                       <contextMetadata>
                                                              <name>longitude</name>
                                                              <type>integer</type>
                                                              <value>25</value>
                                                       </contextMetadata>
                                                </metadata>
```

```
23                                      </contextRegistrationAttribute>
24                                      <contextRegistrationAttribute>
25                                              <name>humidity</name>
26                                              <type>integer</type>
27                                              <isDomain>false</isDomain>
28                                              <metadata>
29                                                      <contextMetadata>
30                                                              <name>longitude</name>
31                                                              <type>integer</type>
32                                                              <value>35</value>
33                                                      </contextMetadata>
34                                              </metadata>
35                                      </contextRegistrationAttribute>
36                              </contextRegistrationAttributeList>
37                              <providingApplication>131.114.59.22/robots</providingApplication>
38                      </contextRegistration>
39              </contextRegistrationResponse>
40              <contextRegistrationResponse>
41                      <contextRegistration>
42                      <entityIdList>
43                              <entityId type="robot" isPattern="false">
44                                      <id>Robot2</id>
45                              </entityId>
46                      </entityIdList>
47                      <contextRegistrationAttributeList>
48                              <contextRegistrationAttribute>
49                                      <name>temperature</name>
50                                      <type>integer</type>
51                                      <isDomain>false</isDomain>
52                                      <metadata>
53                                              <contextMetadata>
54                                                      <name>longitude</name>
55                                                      <type>integer</type>
56                                                      <value>25</value>
57                                              </contextMetadata>
58                                      </metadata>
59                              </contextRegistrationAttribute>
60                              <contextRegistrationAttribute>
61                                      <name>humidity</name>
62                                      <type>integer</type>
63                                      <isDomain>false</isDomain>
64                                      <metadata>
65                                              <contextMetadata>
66                                                      <name>longitude</name>
67                                                      <type>integer</type>
68                                                      <value>60</value>
69                                              </contextMetadata>
70                                      </metadata>
71                              </contextRegistrationAttribute>
72                      </contextRegistrationAttributeList>
73                      <providingApplication>131.114.59.22/robots</providingApplication>
74              </contextRegistration>
75      </contextRegistrationResponse>
76      <contextRegistrationResponse>
77              <contextRegistration>
78                      <entityIdList>
79                              <entityId type="robot" isPattern="false">
80                                      <id>Robot6</id>
81                              </entityId>
82                      </entityIdList>
83                      <contextRegistrationAttributeList>
84                              <contextRegistrationAttribute>
85                                      <name>temperature</name>
86                                      <type>integer</type>
87                                      <isDomain>false</isDomain>
88                                      <metadata>
89                                              <contextMetadata>
90                                                      <name>longitude</name>
91                                                      <type>integer</type>
92                                                      <value>45</value>
93                                              </contextMetadata>
94                                      </metadata>
95                              </contextRegistrationAttribute>
96                              <contextRegistrationAttribute>
97                                      <name>humidity</name>
98                                      <type>integer</type>
99                                      <isDomain>false</isDomain>
100                                     <metadata>
101                                             <contextMetadata>
```

```
                                                    <name>longitude</name>
                                                    <type>integer</type>
                                                    <value>38</value>
                                                </contextMetadata>
                                            </metadata>
                                        </contextRegistrationAttribute>
                                    </contextRegistrationAttributeList>
                                    <providingApplication>131.114.59.22/robots</providingApplication>
                                </contextRegistration>
                        </contextRegistrationResponse>
                </contextRegistrationResponseList>
                <errorCode>
                        <code>200</code>
                        <reasonPhrase>OK</reasonPhrase>
                        <details xsi:type="xs:string">Result</details>
                </errorCode>
</discoverContextAvailabilityResponse>
```

LAB-TP-030-040 - Discovery - Lookup globale - Pattern - Range Query

La richiesta di Discovery con *scope* globale, interroga in primo luogo la DHT per ogni *Range Query* presente nella richiesta. Se sono presenti più range lookup, gli insiemi di risultati ottenuti saranno filtrati utilizzando **AND** come operatore logico. Per ogni risultato restituito si ottiene l'indirizzo relativo all'istanza dell'*IoT Discovery* che possiede la *ContextRegistration* che soddisfa la query e a cui ne verrà fatta richiesta.

Dati in Input

```
POST http://10.0.3.15:8080/ngsi9/discoverContextAvailability
Payload:
<?xml version="1.0"?>
<discoverContextAvailabilityRequest>
  <entityIdList>
    <entityId type="" isPattern="true">
      <id>.*</id>
    </entityId>
  </entityIdList>
  <attributeList>
  </attributeList>
  <restriction>
    <attributeExpression></attributeExpression>
    <scope>
      <operationScope>
        <scopeType>discoveryType</scopeType>
        <scopeValue>global</scopeValue>
      </operationScope>
      <operationScope>
        <scopeType>RangeQuery</scopeType>
        <scopeValue>
          <attribute>temperature</attribute>
          <metadata>longitude</metadata>
          <minValue>-20</minValue>
          <maxValue>30</maxValue>
        </scopeValue>
      </operationScope>
      <operationScope>
        <scopeType>RangeQuery</scopeType>
        <scopeValue>
          <attribute>humidity</attribute>
          <metadata>longitude</metadata>
          <minValue>-89</minValue>
          <maxValue>40</maxValue>
        </scopeValue>
      </operationScope>
```

```
37      </scope>
38    </restriction>
39  </discoverContextAvailabilityRequest>
```

Risultati del test

Nodo 10.0.3.15 - Nodo che riceve la richiesta di Discovery NGSI-9

```
1   INFORMAZIONI: 2016-05-20 14:28:11 10.0.3.1 - 10.0.3.15 8080 POST /ngsi9/discoverContextAvailability - 200 5456 682 5112 http
        ://10.0.3.15:8080 - -
2   Content-Type is: xml
3   Accept: *
4
5   Global Discovery Request
6
7
8   Range query for: temperature:longitude minValue: -20 maxValue: 30
9
10  Range query for: humidity:longitude minValue: -89 maxValue: 40
11  0x378D1A:lorm@:20160520.143449.595: Performing range lookup on 638E38E38E38E38E38E38E38E38E38E388 to
        AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAO
12  0x4AFFCB:lorm@:20160520.143449.595: Performing range lookup on 016C16C16C16C16C16C16C16C16C16C16C to
        B8E38E38E38E38E38E38E38E38E38E38E38D8
13  0x378D1A:lorm@:20160520.143449.625:Send RangeLookupMessage on the wire
14  ***LEAFSET_DELIVER of 378D1A82A81FFD95291D4BDFA17ABFD57A0328FE
15  97A80A1CD6727211CA4B5413A18B43CE7F315752;
16  C2F034DBCA2C746AF2D71E09797D3569C62F15C5;
17  ****
18  SEND[]
19  []
20  RangeLookup request from: 378D1A82A81FFD95291D4BDFA17ABFD57A0328FE
21  0x4AFFCB:lorm@:20160520.143449.645:Send RangeLookupMessage on the wire
22  ---------Node LeafSet:--------
23  97A80A1CD6727211CA4B5413A18B43CE7F315752
24  C2F034DBCA2C746AF2D71E09797D3569C62F15C5
25  ------------------------------
26
27  Out of Bound Node Added
28
29  ***NodeListForRangeLookup:[378D1A82A81FFD95291D4BDFA17ABFD57A0328FE, 97A80A1CD6727211CA4B5413A18B43CE7F315752,
        C2F034DBCA2C746AF2D71E09797D3569C62F15C5]
30  ---> Request Data from 97A80A1CD6727211CA4B5413A18B43CE7F315752
31  ---> Request Data from C2F034DBCA2C746AF2D71E09797D3569C62F15C5
32  SEND[]
33  []
34  Successfully looked up
35  cluster: humidity:longitude
36  content:[Robot4;robot:10.0.3.1] hash:[<0x85B05B05B05B05B05B05B05B05B05B05B05A8..>]
37   for key RANGE
38  SEND[Robot4;robot:10.0.3.1]
39  [Robot4;robot:10.0.3.1]
40  RangeLookup request from: 4AFFCB5523C823082E830E38100CDC1AAFCC0791
41  NEWLIST: [378D1A82A81FFD95291D4BDFA17ABFD57A0328FE, 97A80A1CD6727211CA4B5413A18B43CE7F315752,
        C2F034DBCA2C746AF2D71E09797D3569C62F15C5]
42  SEND[]
43  []
44  ---------Node LeafSet:--------
45  4AFFCB5523C823082E830E38100CDC1AAFCC0791
46  ------------------------------
47  ***NodeListForRangeLookup:[4AFFCB5523C823082E830E38100CDC1AAFCC0791]
48  Successfully looked up
49  cluster: humidity:longitude
50  content:[Robot5;robot:10.0.3.1] hash:[<0x071C71C71C71C71C71C71C71C71C71C71C71C..>]
51   for key RANGE
52  Successfully looked up
53  cluster: humidity:longitude
54  content:[Robot3;robot:10.0.3.1] hash:[<0xB1C71C71C71C71C71C71C71C71C71C71BC..>]
55  content:[Robot1;robot:10.0.3.1] hash:[<0xB1C71C71C71C71C71C71C71C71C71C71BC..>]
56   for key RANGE
57  Successfully looked up
58  cluster: humidity:longitude
59  content:[Robot6;robot:10.0.3.1] hash:[<0xB60B60B60B60B60B60B60B60B60B60B600..>]
60   for key RANGE
61  SEND[Robot4;robot:10.0.3.1, Robot5;robot:10.0.3.1, Robot3;robot:10.0.3.1, Robot1;robot:10.0.3.1, Robot6;robot:10.0.3.1]
```

```
[Robot4;robot:10.0.3.1, Robot5;robot:10.0.3.1, Robot3;robot:10.0.3.1, Robot1;robot:10.0.3.1, Robot6;robot:10.0.3.1]
NEWLIST: [378D1A82A81FFD95291D4BDFA17ABFD57A0328FE, 97A80A1CD6727211CA4B5413A18B43CE7F315752,
        C2F034DBCA2C746AF2D71E09797D3569C62F15C5]
Successfully looked up
cluster: temperature:longitude
content:[Robot5;robot:10.0.3.1] hash:[<0x6AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA4..>]
 for key RANGE
Successfully looked up
cluster: temperature:longitude
content:[Robot4;robot:10.0.3.1] hash:[<0x89F49F49F49F49F49F49F49F49F49F49F49F49EC..>]
 for key RANGE
Successfully looked up
cluster: temperature:longitude
content:[Robot2;robot:10.0.3.1] hash:[<0xA38E38E38E38E38E38E38E38E38E38E38E38E384..>]
content:[Robot1;robot:10.0.3.1] hash:[<0xA38E38E38E38E38E38E38E38E38E38E38E38E384..>]
 for key RANGE
SEND[Robot5;robot:10.0.3.1, Robot4;robot:10.0.3.1, Robot2;robot:10.0.3.1, Robot1;robot:10.0.3.1]
[Robot5;robot:10.0.3.1, Robot4;robot:10.0.3.1, Robot2;robot:10.0.3.1, Robot1;robot:10.0.3.1]
Final result:
 {10.0.3.1=[Robot5;robot, Robot4;robot, Robot1;robot]}
```

Nodo 10.0.3.1 - Nodo remoto appartentente alla DHT

Questo nodo riceve da parte di **10.0.3.15** una richiesta di discovery locale per ContextRegistration relative alle entità *[Robot5;robot, Robot4;robot, Robot1;robot]*

```
***LEAFSET_DELIVER of C2F034DBCA2C746AF2D71E09797D3569C62F15C5
378D1A82A81FFD95291D4BDFA17ABFD57A0328FE;
97A80A1CD6727211CA4B5413A18B43CE7F315752;
****
Content-Type is: xml
Accept: *
```

Nodo 10.0.3.133 - Nodo remoto appartentente alla DHT

```
***LEAFSET_DELIVER of D711D710E801592EB6157596EF00886BF21474DB
4AFFCB5523C823082E830E38100CDC1AAFCC0791;
****
***LEAFSET_DELIVER of 97A80A1CD6727211CA4B5413A18B43CE7F315752
378D1A82A81FFD95291D4BDFA17ABFD57A0328FE;
C2F034DBCA2C746AF2D71E09797D3569C62F15C5;
****
```

Risposta NGSI9 al richiedente


```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<discoverContextAvailabilityResponse>
        <contextRegistrationResponseList>
                <contextRegistrationResponse>
                        <contextRegistration>
                        <entityIdList>
                                <entityId type="robot" isPattern="false">
                                        <id>Robot5</id>
                                </entityId>
                        </entityIdList>
                        <contextRegistrationAttributeList>
                                <contextRegistrationAttribute>
                                        <name>temperature</name>
                                        <type>integer</type>
                                        <isDomain>false</isDomain>
                                        <metadata>
                                                <contextMetadata>
                                                        <name>longitude</name>
                                                        <type>integer</type>
```

```
20                                      <value>-15</value>
21                                  </contextMetadata>
22                          </metadata>
23                   </contextRegistrationAttribute>
24                   <contextRegistrationAttribute>
25                          <name>humidity</name>
26                          <type>integer</type>
27                          <isDomain>false</isDomain>
28                          <metadata>
29                          <contextMetadata>
30                                      <name>longitude</name>
31                                      <type>integer</type>
32                                      <value>-85</value>
33                                  </contextMetadata>
34                          </metadata>
35                   </contextRegistrationAttribute>
36            </contextRegistrationAttributeList>
37           <providingApplication>131.114.59.22/robots</providingApplication>
38            </contextRegistration>
39     </contextRegistrationResponse>
40     <contextRegistrationResponse>
41           <contextRegistration>
42                  <entityIdList>
43                         <entityId type="robot" isPattern="false">
44                                <id>Robot4</id>
45                         </entityId>
46                  </entityIdList>
47                  <contextRegistrationAttributeList>
48                         <contextRegistrationAttribute>
49                                <name>temperature</name>
50                                <type>integer</type>
51                                <isDomain>false</isDomain>
52                                <metadata>
53                                       <contextMetadata>
54                                              <name>longitude</name>
55                                              <type>integer</type>
56                                              <value>7</value>
57                                        </contextMetadata>
58                                </metadata>
59                          </contextRegistrationAttribute>
60                          <contextRegistrationAttribute>
61                                <name>humidity</name>
62                                <type>integer</type>
63                                <isDomain>false</isDomain>
64                                <metadata>
65                                       <contextMetadata>
66                                              <name>longitude</name>
67                                              <type>integer</type>
68                                              <value>4</value>
69                                        </contextMetadata>
70                                </metadata>
71                           </contextRegistrationAttribute>
72                    </contextRegistrationAttributeList>
73                   <providingApplication>131.114.59.22/robots</providingApplication>
74            </contextRegistration>
75     </contextRegistrationResponse>
76     <contextRegistrationResponse>
77           <contextRegistration>
78                  <entityIdList>
79                         <entityId type="robot" isPattern="false">
80                                <id>Robot1</id>
81                         </entityId>
82                  </entityIdList>
83                  <contextRegistrationAttributeList>
84                         <contextRegistrationAttribute>
85                                <name>temperature</name>
86                                <type>integer</type>
87                                <isDomain>false</isDomain>
88                                <metadata>
89                                       <contextMetadata>
90                                              <name>longitude</name>
91                                              <type>integer</type>
92                                              <value>25</value>
93                                        </contextMetadata>
94                                </metadata>
95                         </contextRegistrationAttribute>
96                         <contextRegistrationAttribute>
97                                <name>humidity</name>
98                                <type>integer</type>
```

```
                                        <isDomain>false</isDomain>
                                        <metadata>
                                                <contextMetadata>
                                                        <name>longitude</name>
                                                        <type>integer</type>
                                                        <value>35</value>
                                                 </contextMetadata>
                                        </metadata>
                                    </contextRegistrationAttribute>
                                </contextRegistrationAttributeList>
                                <providingApplication>131.114.59.22/robots</providingApplication>
                        </contextRegistration>
                </contextRegistrationResponse>
        </contextRegistrationResponseList>
        <errorCode>
                <code>200</code>
                <reasonPhrase>OK</reasonPhrase>
                <details xsi:type="xs:string">Result</details>
        </errorCode>
 </discoverContextAvailabilityResponse>
```

LAB-TP-030-050 - Discovery - Lookup Globale - Range Query OutOfRange

Dati in Input

```
POST http://10.0.3.1:8080/ngsi9/discoverContextAvailability
Payload:
<?xml version="1.0"?>
<discoverContextAvailabilityRequest>
  <entityIdList>
    <entityId type="" isPattern="true">
      <id>.*</id>
    </entityId>
  </entityIdList>
  <attributeList>
  </attributeList>
  <restriction>
    <attributeExpression></attributeExpression>
    <scope>
      <operationScope>
        <scopeType>discoveryType</scopeType>
        <scopeValue>global</scopeValue>
      </operationScope>
      <operationScope>
        <scopeType>RangeQuery</scopeType>
        <scopeValue>
          <attribute>temperature</attribute>
          <metadata>longitude</metadata>
          <minValue>-20</minValue>
          <maxValue>180</maxValue>
        </scopeValue>
      </operationScope>
    </scope>
  </restriction>
</discoverContextAvailabilityRequest>
```

Risultati del test

Nodo 10.0.3.1 - Nodo che riceve la richiesta di ContextRegistration NGSI-9

```
Content-Type is: xml
Accept: *

Global Discovery Request

```

```
6 Range query for: temperature:longitude minValue: -20 maxValue: 180
7 Values out of range: Range -90 to 90
8 Exception in Global Discovery Request. temperature:longitude: Values out of range: Range -90 to 90
```

Risposta NGSI9 al richiedente

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
2 <discoverContextAvailabilityResponse>
3       <errorCode>
4             <code>404</code>
5             <reasonPhrase>java.lang.Exception: temperature:longitude: Values out of range: Range -90 to 90</reasonPhrase>
6             <details xsi:type="xs:string" />
7       </errorCode>
8 </discoverContextAvailabilityResponse>
```

A.2 Risultati dettagliati dello scenario

A.2.1 Discovery - Lookup Globale - Range Query

Dati in Input

```
POST http://10.2.2.29:8080/ngsi9/discoverContextAvailability
Payload:
<?xml version="1.0"?>
<discoverContextAvailabilityRequest>
  <entityIdList>
    <entityId type="" isPattern="true">
      <id>.*</id>
    </entityId>
  </entityIdList>
  <attributeList>
  </attributeList>
  <restriction>
    <attributeExpression></attributeExpression>
    <scope>
      <operationScope>
        <scopeType>discoveryType</scopeType>
        <scopeValue>global</scopeValue>
      </operationScope>
      <operationScope>
        <scopeType>RangeQuery</scopeType>
        <scopeValue>
          <attribute>temperature</attribute>
          <metadata>longitude</metadata>
          <type>float</type>
          <minValue>10.3875</minValue>
          <maxValue>10.3915</maxValue>
        </scopeValue>
      </operationScope>
      <operationScope>
        <scopeType>RangeQuery</scopeType>
        <scopeValue>
          <attribute>temperature</attribute>
          <metadata>latitude</metadata>
          <type>float</type>
          <minValue>43.7202</minValue>
          <maxValue>43.7221</maxValue>
        </scopeValue>
      </operationScope>
    </scope>
  </restriction>
</discoverContextAvailabilityRequest>
```

Risultati del test

```
Status:
200: OK Loading time: 417 ms

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<discoverContextAvailabilityResponse>
   <contextRegistrationResponseList>
      <contextRegistrationResponse>
         <contextRegistration>
            <entityIdList>
               <entityId type="agent1" isPattern="false">
                  <id>sensor_29</id>
               </entityId>
            </entityIdList>
            <contextRegistrationAttributeList>
               <contextRegistrationAttribute>
                  <name>temperature</name>
                  <type>integer</type>
                  <isDomain>false</isDomain>
                  <metadata>
                     <contextMetadata>
                        <name>latitude</name>
                        <type>float</type>
```

```
23                          <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">43.72074</value>
24                      </contextMetadata>
25                      <contextMetadata>
26                          <name>longitude</name>
27                          <type>float</type>
28                          <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">10.390742</value>
29                      </contextMetadata>
30                      <contextMetadata>
31                          <name>accuracy</name>
32                          <type>integer</type>
33                          <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">93</value>
34                      </contextMetadata>
35                      <contextMetadata>
36                          <name>delaytime</name>
37                          <type>integer</type>
38                          <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">9597</value>
39                      </contextMetadata>
40                  </metadata>
41              </contextRegistrationAttribute>
42              <contextRegistrationAttribute>
43                  <name>humidity</name>
44                  <type>integer</type>
45                  <isDomain>false</isDomain>
46                  <metadata>
47                      <contextMetadata>
48                          <name>latitude</name>
49                          <type>float</type>
50                          <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">43.72074</value>
51                      </contextMetadata>
52                      <contextMetadata>
53                          <name>longitude</name>
54                          <type>float</type>
55                          <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">10.390742</value>
56                      </contextMetadata>
57                      <contextMetadata>
58                          <name>accuracy</name>
59                          <type>integer</type>
60                          <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">93</value>
61                      </contextMetadata>
62                      <contextMetadata>
63                          <name>delaytime</name>
64                          <type>integer</type>
65                          <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">9597</value>
66                      </contextMetadata>
67                  </metadata>
68              </contextRegistrationAttribute>
69          </contextRegistrationAttributeList>
70          <providingApplication>http://10.2.2.31:4041/ngsi10</providingApplication>
71      </contextRegistration>
72  </contextRegistrationResponse>
73  <contextRegistrationResponse>
74      <contextRegistration>
75          <entityIdList>
76              <entityId type="agent1" isPattern="false">
77                  <id>sensor_14</id>
78              </entityId>
79          </entityIdList>
80          <contextRegistrationAttributeList>
81              <contextRegistrationAttribute>
82                  <name>temperature</name>
83                  <type>integer</type>
84                  <isDomain>false</isDomain>
85                  <metadata>
86                      <contextMetadata>
87                          <name>latitude</name>
88                          <type>float</type>
89                          <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">43.720988</value>
90                      </contextMetadata>
91                      <contextMetadata>
92                          <name>longitude</name>
```

```
93                          <type>float</type>
94                          <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">10.390982</value>
95                      </contextMetadata>
96                      <contextMetadata>
97                          <name>accuracy</name>
98                          <type>integer</type>
99                          <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">4</value>
100                     </contextMetadata>
101                     <contextMetadata>
102                         <name>delaytime</name>
103                         <type>integer</type>
104                         <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">2409</value>
105                     </contextMetadata>
106                 </metadata>
107             </contextRegistrationAttribute>
108             <contextRegistrationAttribute>
109                 <name>humidity</name>
110                 <type>integer</type>
111                 <isDomain>false</isDomain>
112                 <metadata>
113                     <contextMetadata>
114                         <name>latitude</name>
115                         <type>float</type>
116                         <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">43.720988</value>
117                     </contextMetadata>
118                     <contextMetadata>
119                         <name>longitude</name>
120                         <type>float</type>
121                         <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">10.390982</value>
122                     </contextMetadata>
123                     <contextMetadata>
124                         <name>accuracy</name>
125                         <type>integer</type>
126                         <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">4</value>
127                     </contextMetadata>
128                     <contextMetadata>
129                         <name>delaytime</name>
130                         <type>integer</type>
131                         <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">2409</value>
132                     </contextMetadata>
133                 </metadata>
134             </contextRegistrationAttribute>
135         </contextRegistrationAttributeList>
136         <providingApplication>http://10.2.2.31:4041/ngsi10</providingApplication>
137     </contextRegistration>
138 </contextRegistrationResponse>
139 <contextRegistrationResponse>
140     <contextRegistration>
141         <entityIdList>
142             <entityId type="agent4" isPattern="false">
143                 <id>sensor_23</id>
144             </entityId>
145         </entityIdList>
146         <contextRegistrationAttributeList>
147             <contextRegistrationAttribute>
148                 <name>temperature</name>
149                 <type>integer</type>
150                 <isDomain>false</isDomain>
151                 <metadata>
152                     <contextMetadata>
153                         <name>latitude</name>
154                         <type>float</type>
155                         <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">43.72173</value>
156                     </contextMetadata>
157                     <contextMetadata>
158                         <name>longitude</name>
159                         <type>float</type>
160                         <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">10.388528</value>
161                     </contextMetadata>
162                     <contextMetadata>
```

```
163                         <name>accuracy</name>
164                         <type>integer</type>
165                         <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">64</value>
166                      </contextMetadata>
167                      <contextMetadata>
168                         <name>delaytime</name>
169                         <type>integer</type>
170                         <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">3468</value>
171                      </contextMetadata>
172                   </metadata>
173                </contextRegistrationAttribute>
174                <contextRegistrationAttribute>
175                   <name>humidity</name>
176                   <type>integer</type>
177                   <isDomain>false</isDomain>
178                   <metadata>
179                      <contextMetadata>
180                         <name>latitude</name>
181                         <type>float</type>
182                         <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">43.72173</value>
183                      </contextMetadata>
184                      <contextMetadata>
185                         <name>longitude</name>
186                         <type>float</type>
187                         <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">10.388528</value>
188                      </contextMetadata>
189                      <contextMetadata>
190                         <name>accuracy</name>
191                         <type>integer</type>
192                         <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">64</value>
193                      </contextMetadata>
194                      <contextMetadata>
195                         <name>delaytime</name>
196                         <type>integer</type>
197                         <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">3468</value>
198                      </contextMetadata>
199                   </metadata>
200                </contextRegistrationAttribute>
201             </contextRegistrationAttributeList>
202             <providingApplication>http://10.2.2.34:4041/ngsi10</providingApplication>
203          </contextRegistration>
204       </contextRegistrationResponse>
205       <contextRegistrationResponse>
206          <contextRegistration>
207             <entityIdList>
208                <entityId type="agent4" isPattern="false">
209                   <id>sensor_8</id>
210                </entityId>
211             </entityIdList>
212             <contextRegistrationAttributeList>
213                <contextRegistrationAttribute>
214                   <name>temperature</name>
215                   <type>integer</type>
216                   <isDomain>false</isDomain>
217                   <metadata>
218                      <contextMetadata>
219                         <name>latitude</name>
220                         <type>float</type>
221                         <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">43.721737</value>
222                      </contextMetadata>
223                      <contextMetadata>
224                         <name>longitude</name>
225                         <type>float</type>
226                         <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">10.390766</value>
227                      </contextMetadata>
228                      <contextMetadata>
229                         <name>accuracy</name>
230                         <type>integer</type>
231                         <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">31</value>
232                      </contextMetadata>
```

```
                        <contextMetadata>
                            <name>delaytime</name>
                            <type>integer</type>
                            <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">285</value>
                        </contextMetadata>
                    </metadata>
                </contextRegistrationAttribute>
                <contextRegistrationAttribute>
                    <name>humidity</name>
                    <type>integer</type>
                    <isDomain>false</isDomain>
                    <metadata>
                        <contextMetadata>
                            <name>latitude</name>
                            <type>float</type>
                            <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">43.721737</value>
                        </contextMetadata>
                        <contextMetadata>
                            <name>longitude</name>
                            <type>float</type>
                            <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">10.390766</value>
                        </contextMetadata>
                        <contextMetadata>
                            <name>accuracy</name>
                            <type>integer</type>
                            <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">31</value>
                        </contextMetadata>
                        <contextMetadata>
                            <name>delaytime</name>
                            <type>integer</type>
                            <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">285</value>
                        </contextMetadata>
                    </metadata>
                </contextRegistrationAttribute>
            </contextRegistrationAttributeList>
            <providingApplication>http://10.2.2.34:4041/ngsi10</providingApplication>
        </contextRegistration>
    </contextRegistrationResponse>
</contextRegistrationResponseList>
<errorCode>
    <code>200</code>
    <reasonPhrase>OK</reasonPhrase>
    <details xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
            instance">Result</details>
</errorCode>
</discoverContextAvailabilityResponse>
```

A.2.2 Discovery - Lookup Globale - Range Query Multiattributo

Dati in Input

```
POST http://10.2.2.19:8080/ngsi9/discoverContextAvailability
Payload:
<?xml version="1.0"?>
<discoverContextAvailabilityRequest>
  <entityIdList>
    <entityId type="" isPattern="true">
      <id>.*</id>
    </entityId>
  </entityIdList>
  <attributeList>
  </attributeList>
  <restriction>
    <attributeExpression></attributeExpression>
    <scope>
      <operationScope>
        <scopeType>discoveryType</scopeType>
```

```
        <scopeValue>global</scopeValue>
      </operationScope>
      <operationScope>
        <scopeType>RangeQuery</scopeType>
        <scopeValue>
          <attribute>pressure</attribute>
          <metadata>longitude</metadata>
          <type>float</type>
          <minValue>10.3959</minValue>
          <maxValue>10.4046</maxValue>
        </scopeValue>
      </operationScope>
      <operationScope>
        <scopeType>RangeQuery</scopeType>
        <scopeValue>
          <attribute>pressure</attribute>
          <metadata>latitude</metadata>
          <type>float</type>
          <minValue>43.7194</minValue>
          <maxValue>43.7234</maxValue>
        </scopeValue>
      </operationScope>
      <operationScope>
        <scopeType>RangeQuery</scopeType>
        <scopeValue>
          <attribute>wind</attribute>
          <metadata>longitude</metadata>
          <type>float</type>
          <minValue>10.3969</minValue>
          <maxValue>10.4046</maxValue>
        </scopeValue>
      </operationScope>
      <operationScope>
        <scopeType>RangeQuery</scopeType>
        <scopeValue>
          <attribute>wind</attribute>
          <metadata>latitude</metadata>
          <type>float</type>
          <minValue>43.7193</minValue>
          <maxValue>43.7235</maxValue>
        </scopeValue>
      </operationScope>
    </scope>
  </restriction>
</discoverContextAvailabilityRequest>
```

Risultati del test

```
Status:
200: OK Loading time: 356 ms

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<discoverContextAvailabilityResponse>
    <contextRegistrationResponseList>
        <contextRegistrationResponse>
            <contextRegistration>
                <entityIdList>
                    <entityId type="agent2" isPattern="false">
                        <id>sensor_24</id>
                    </entityId>
                </entityIdList>
                <contextRegistrationAttributeList>
                    <contextRegistrationAttribute>
                        <name>pressure</name>
                        <type>integer</type>
                        <isDomain>false</isDomain>
                        <metadata>
                            <contextMetadata>
                                <name>latitude</name>
                                <type>float</type>
                                <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                        /2001/XMLSchema-instance">43.72017</value>
                            </contextMetadata>
                            <contextMetadata>
                                <name>longitude</name>
```

```
27                          <type>float</type>
28                          <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">10.397994</value>
29                      </contextMetadata>
30                      <contextMetadata>
31                          <name>accuracy</name>
32                          <type>integer</type>
33                          <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">88</value>
34                      </contextMetadata>
35                      <contextMetadata>
36                          <name>delaytime</name>
37                          <type>integer</type>
38                          <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">2055</value>
39                      </contextMetadata>
40                  </metadata>
41              </contextRegistrationAttribute>
42              <contextRegistrationAttribute>
43                  <name>wind</name>
44                  <type>integer</type>
45                  <isDomain>false</isDomain>
46                  <metadata>
47                      <contextMetadata>
48                          <name>latitude</name>
49                          <type>float</type>
50                          <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">43.72017</value>
51                      </contextMetadata>
52                      <contextMetadata>
53                          <name>longitude</name>
54                          <type>float</type>
55                          <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">10.397994</value>
56                      </contextMetadata>
57                      <contextMetadata>
58                          <name>accuracy</name>
59                          <type>integer</type>
60                          <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">88</value>
61                      </contextMetadata>
62                      <contextMetadata>
63                          <name>delaytime</name>
64                          <type>integer</type>
65                          <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">2055</value>
66                      </contextMetadata>
67                  </metadata>
68              </contextRegistrationAttribute>
69          </contextRegistrationAttributeList>
70          <providingApplication>http://10.2.2.33:4041/ngsi10</providingApplication>
71      </contextRegistration>
72  </contextRegistrationResponse>
73  <contextRegistrationResponse>
74      <contextRegistration>
75          <entityIdList>
76              <entityId type="agent2" isPattern="false">
77                  <id>sensor_11</id>
78              </entityId>
79          </entityIdList>
80          <contextRegistrationAttributeList>
81              <contextRegistrationAttribute>
82                  <name>pressure</name>
83                  <type>integer</type>
84                  <isDomain>false</isDomain>
85                  <metadata>
86                      <contextMetadata>
87                          <name>latitude</name>
88                          <type>float</type>
89                          <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">43.72137</value>
90                      </contextMetadata>
91                      <contextMetadata>
92                          <name>longitude</name>
93                          <type>float</type>
94                          <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">10.402349</value>
95                      </contextMetadata>
96                      <contextMetadata>
```

```
97                          <name>accuracy</name>
98                          <type>integer</type>
99                          <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">2</value>
100                         </contextMetadata>
101                         <contextMetadata>
102                             <name>delaytime</name>
103                             <type>integer</type>
104                             <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">5228</value>
105                         </contextMetadata>
106                     </metadata>
107                 </contextRegistrationAttribute>
108                 <contextRegistrationAttribute>
109                     <name>wind</name>
110                     <type>integer</type>
111                     <isDomain>false</isDomain>
112                     <metadata>
113                         <contextMetadata>
114                             <name>latitude</name>
115                             <type>float</type>
116                             <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">43.72137</value>
117                         </contextMetadata>
118                         <contextMetadata>
119                             <name>longitude</name>
120                             <type>float</type>
121                             <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">10.402349</value>
122                         </contextMetadata>
123                         <contextMetadata>
124                             <name>accuracy</name>
125                             <type>integer</type>
126                             <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">2</value>
127                         </contextMetadata>
128                         <contextMetadata>
129                             <name>delaytime</name>
130                             <type>integer</type>
131                             <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">5228</value>
132                         </contextMetadata>
133                     </metadata>
134                 </contextRegistrationAttribute>
135             </contextRegistrationAttributeList>
136             <providingApplication>http://10.2.2.33:4041/ngsi10</providingApplication>
137         </contextRegistration>
138     </contextRegistrationResponse>
139     <contextRegistrationResponse>
140         <contextRegistration>
141             <entityIdList>
142                 <entityId type="agent5" isPattern="false">
143                     <id>sensor_19</id>
144                 </entityId>
145             </entityIdList>
146             <contextRegistrationAttributeList>
147                 <contextRegistrationAttribute>
148                     <name>pressure</name>
149                     <type>integer</type>
150                     <isDomain>false</isDomain>
151                     <metadata>
152                         <contextMetadata>
153                             <name>latitude</name>
154                             <type>float</type>
155                             <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">43.721844</value>
156                         </contextMetadata>
157                         <contextMetadata>
158                             <name>longitude</name>
159                             <type>float</type>
160                             <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">10.398315</value>
161                         </contextMetadata>
162                         <contextMetadata>
163                             <name>accuracy</name>
164                             <type>integer</type>
165                             <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">11</value>
166                         </contextMetadata>
```

```
167                         <contextMetadata>
168                             <name>delaytime</name>
169                             <type>integer</type>
170                             <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                        /2001/XMLSchema-instance">4960</value>
171                         </contextMetadata>
172                     </metadata>
173                 </contextRegistrationAttribute>
174                 <contextRegistrationAttribute>
175                     <name>wind</name>
176                     <type>integer</type>
177                     <isDomain>false</isDomain>
178                     <metadata>
179                         <contextMetadata>
180                             <name>latitude</name>
181                             <type>float</type>
182                             <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                        /2001/XMLSchema-instance">43.721844</value>
183                         </contextMetadata>
184                         <contextMetadata>
185                             <name>longitude</name>
186                             <type>float</type>
187                             <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                        /2001/XMLSchema-instance">10.398315</value>
188                         </contextMetadata>
189                         <contextMetadata>
190                             <name>accuracy</name>
191                             <type>integer</type>
192                             <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                        /2001/XMLSchema-instance">11</value>
193                         </contextMetadata>
194                         <contextMetadata>
195                             <name>delaytime</name>
196                             <type>integer</type>
197                             <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                        /2001/XMLSchema-instance">4960</value>
198                         </contextMetadata>
199                     </metadata>
200                 </contextRegistrationAttribute>
201             </contextRegistrationAttributeList>
202             <providingApplication>http://10.2.2.35:4041/ngsi10</providingApplication>
203         </contextRegistration>
204     </contextRegistrationResponse>
205     <contextRegistrationResponse>
206         <contextRegistration>
207             <entityIdList>
208                 <entityId type="agent5" isPattern="false">
209                     <id>sensor_14</id>
210                 </entityId>
211             </entityIdList>
212             <contextRegistrationAttributeList>
213                 <contextRegistrationAttribute>
214                     <name>pressure</name>
215                     <type>integer</type>
216                     <isDomain>false</isDomain>
217                     <metadata>
218                         <contextMetadata>
219                             <name>latitude</name>
220                             <type>float</type>
221                             <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                        /2001/XMLSchema-instance">43.723041</value>
222                         </contextMetadata>
223                         <contextMetadata>
224                             <name>longitude</name>
225                             <type>float</type>
226                             <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                        /2001/XMLSchema-instance">10.400068</value>
227                         </contextMetadata>
228                         <contextMetadata>
229                             <name>accuracy</name>
230                             <type>integer</type>
231                             <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                        /2001/XMLSchema-instance">10</value>
232                         </contextMetadata>
233                         <contextMetadata>
234                             <name>delaytime</name>
235                             <type>integer</type>
236                             <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                        /2001/XMLSchema-instance">103</value>
```

```
237                         </contextMetadata>
238                     </metadata>
239                 </contextRegistrationAttribute>
240                 <contextRegistrationAttribute>
241                     <name>wind</name>
242                     <type>integer</type>
243                     <isDomain>false</isDomain>
244                     <metadata>
245                         <contextMetadata>
246                             <name>latitude</name>
247                             <type>float</type>
248                             <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">43.723041</value>
249                         </contextMetadata>
250                         <contextMetadata>
251                             <name>longitude</name>
252                             <type>float</type>
253                             <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">10.400068</value>
254                         </contextMetadata>
255                         <contextMetadata>
256                             <name>accuracy</name>
257                             <type>integer</type>
258                             <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">10</value>
259                         </contextMetadata>
260                         <contextMetadata>
261                             <name>delaytime</name>
262                             <type>integer</type>
263                             <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
                                    /2001/XMLSchema-instance">103</value>
264                         </contextMetadata>
265                     </metadata>
266                 </contextRegistrationAttribute>
267             </contextRegistrationAttributeList>
268             <providingApplication>http://10.2.2.35:4041/ngsi10</providingApplication>
269         </contextRegistration>
270     </contextRegistrationResponse>
271 </contextRegistrationResponseList>
272 <errorCode>
273     <code>200</code>
274     <reasonPhrase>OK</reasonPhrase>
275     <details xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
            instance">Result</details>
276 </errorCode>
277 </discoverContextAvailabilityResponse>
```

Bibliografia

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Comput. Netw.*, vol. 54, pp. 2787–2805, Oct. 2010.

[2] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, I. S. Jubert, M. Mazura, M. Harrison, M. Eisenhauer, *et al.*, "Internet of things strategic research roadmap," *Internet of Things-Global Technological and Societal Trends*, pp. 9–52, 2011.

[3] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé, eds., *Vision and Challenges for Realising the Internet of Things*. Luxembourg: Publications Office of the European Union, 2010.

[4] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia, and M. Dohler, "Standardized protocol stack for the internet of (important) things," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 3, pp. 1389–1406, 2013.

[5] S. Cirani, L. Davoli, G. Ferrari, R. Léone, P. Medagliani, M. Picone, and L. Veltri, "A scalable and self-configuring architecture for service discovery in the internet of things," *Internet of Things Journal, IEEE*, vol. 1, no. 5, pp. 508–521, 2014.

[6] M. Lupi, "Progettazione ed implementazione di un servizio distribuito di look-up delle risorse per l'internet delle cose," Master's thesis, Università di Pisa, 2013.

[7] H. Shen and C.-Z. Xu, "Leveraging a compound graph-based dht for multi-attribute range queries with performance analysis," *Computers, IEEE Transactions on*, vol. 61, no. 4, pp. 433–447, 2012.

[8] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (coap)," 2014.

[9] B. C. Villaverde, R. De Paz Alberola, A. J. Jara, S. Fedor, S. K. Das, and D. Pesch, "Service discovery protocols for constrained machine-to-machine

communications," *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 1, pp. 41–60, 2014.

[10] H. Balakrishnan, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Looking up data in p2p systems," *Communications of the ACM*, vol. 46, no. 2, pp. 43–48, 2003.

[11] B. Fabian, "Implementing secure p2p-ons," in *Communications, 2009. ICC'09. IEEE International Conference on*, pp. 1–5, IEEE, 2009.

[12] N. Schoenemann, K. Fischbach, and D. Schoder, "P2p architecture for ubiquitous supply chain systems," 2009.

[13] H. Shen, C.-Z. Xu, and G. Chen, "Cycloid: a constant-degree and lookup-efficient p2p overlay network," *Performance Evaluation*, vol. 63, no. 3, pp. 195–216, 2006.

[14] "FIWARE Catalogue." `http://catalogue.fiware.org/`.

[15] "FIWARE Wiki - Documentation." `https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/Main_Page`.

[16] OMA, *NGSI Context Management*, Approved Version 1.0 ed., May 2012. Tecnical Specification.

[17] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *Middleware 2001*, pp. 329–350.