FONDO EUROPEO DI SVILUPPO REGIONALE
P.O.R. 2007 – 2013

PROGETTO QUIES

DOCUMENTO CON ESITI ANALISI SPECIFICHE SMARTDATANET

Partner:	Università degli Studi di Torino – Dipartimento di Informatica
Workpackage:	WP7 – Integrazione con Smartdatanet
Task:	Task 7.1 Raccolta e analisi specifiche della piattaforma regionale Smardatanet
Deliverable	Deliverable 7.1.1 Documento con esiti analisi specifiche Smartdatanet
Versione	1
Allegati	-
Data di rilascio	30 Gennaio 2015

1 Introduzione
1.1 Yucca platform
1.2 Architettura della piattaforma
2 Specifiche tecniche
2.1 Disponibilità dei dataset e dei flussi inviati su SDP
2.2 Panoramica di protocolli e dei formati
2.3 Specifiche per l'invio di dati alfanumerici
2.4 Protocollo per l'invio di dati multimediali
2.5 Specifiche per la fruizione di eventi esposti da SDP
2.6 Rest API per l'accesso a dati della SDP tramite servizi
2.7 Sicurezza
3 Sintesi operativa
3.1 Comunicazione dei sensori con SDP attraverso server intermedio
3.2 Variante del primo scenario (accesso non in streaming)
3.3 Comunicazione diretta dei sensori con la SDP
3.4 Struttura dati
3.5 Fruizione dei dati dalla piattaforma e dati di contesto
3.6 Scenario del progetto QUIES

1 Introduzione

L'ecosistema Smartdatanet permette di condividere e aggregare informazioni prodotte da differenti servizi in ottica multidisciplinare per creare nuove applicazioni, nella direzione di comunità più intelligenti e sostenibili. La piattaforma si rivolge principalmente ad imprese, Pubbliche Amministrazioni, Centri di ricerca e tutti i soggetti pubblici e privati interessati a condividere il loro patrimonio di dati ed a sviluppare nuove applicazioni.
I soggetti che aderiscono all'iniziativa possono diventare parte di un ambiente di lavoro collaborativo, dove è possibile:

- incrementare un patrimonio di dati comune;
- stimolare e realizzare idee innovative.

I flussi di dati che vengono utilizzati a questo scopo derivano dal mondo delle cose (Internet of things, come le telecamere, i sensori di traffico, le centraline meteo) e dal mondo delle persone (Internet of people, come i tweet, o le segnalazioni che possiamo inviare con uno smartphone).

Basato sullo scenario del Realtime Internet of EveryThings, ovvero l'interconnessione di oggetti e persone che producono dati e servizi finalizzati al miglioramento della realtà in cui viviamo, l'Ecosistema nel suo complesso è costituito da:

- soggetti pubblici e privati che collaborano condividendo applicazioni, device, dati, esperienze e infrastrutture tecnologiche abilitanti
- rete di sensori (Sensing Network) e di gateway che raccolgono e inviano dati mettendoli a disposizione di tutto l'Ecosistema
- Piattaforma regionale aperta e gratuita (Yucca Platform) che
    - dà acceso ai dati pubblici presenti nei portali http://www.dati.piemonte.it/ e http://www.geoportale.piemonte.it/
    - consente circolarità delle informazioni e integrazione con altre fonti dati
    - connette rapidamente oggetti, persone e sistemi, fornendo modelli di standardizzazione dei protocolli di raccolta e pubblicazione dei dati
    - facilita l'accesso a tecnologie innovative per la realizzazione di soluzioni end-to-end
- Marketplace (Smart Data Marketplace) delle applicazioni disponibili e condivise che utilizzano funzionalità, eventi e dati della piattaforma, ma eseguite in ambienti indipendenti. Sarà possibile trovare:
    - applicazioni sviluppate e messe a disposizione dagli sviluppatori (web application, mobile apps, cruscotti, dashboard analitiche, etc.)
    - business case e living labs
    - framework, librerie per gateway
    - esempi d'utilizzo
    - device, gateway, sensori utilizzabili.

L'obiettivo dell'ecosistema smartdatanet è andare nella direzione di comunità più intelligenti e sostenibili (*smart communities* o *smart cities*) :

1. Qualità dell'aria. Dispositivi wireless low-cost permettono di raccogliere dati in tempo reale sulla qualità dell'aria da una moltitudine di posizioni all'interno di una città.
2. I dati possono essere aggregati e trasformati in applicazioni utili, ad esempio un'app per aiutare ciclisti, corridori e pendolari a evitare zone particolarmente problematiche della città.

3. Qualità dell'acqua. Un sistema di sensori a domicilio, progettato per rilevare le perdite dell'impianto idraulico e allertare i proprietari della necessità di riparazioni, è un esempio di come le tecnologie possano avere un impatto sulla sostenibilità.
4. Flussi di traffico. I dati sul traffico nelle città possono essere raccolti e studiati per ottimizzare i segnali stradali, per ridurre incidenti e congestione.
5. Ad esempio, i ricercatori dell'Università Carlos III di Madrid hanno sperimentato sistemi di telerilevamento che consentono di misurare le emissioni dei veicoli in tempo reale. Non solo possono etichettare i veicoli dalle emissioni pesanti, ma possono anche misurare l'impatto delle diverse politiche di gestione del traffico e scegliere quella più efficace.
6. Illuminazione. Raccogliendo i dati sull'illuminazione nei centri abitati, attraverso appositi sensori, è possibile, ad esempio, ottimizzare l'illuminazione stradale, risparmiare energia elettrica e ottenere una segnalazione automatica dei guasti.
7. Rifiuti. Si possono applicare sensori per rilevare peso dei rifiuti, presenza di rifiuti non idonei o fuori dal cassonetto, identità del cittadino che li ha depositati. Questo avrà per risultato un efficientamento della raccolta, un'automazione del calcolo della tassa sui rifiuti, e una sensibilizzazione dei cittadini allo smaltimento differenziato.

1.1 Yucca platform

La piattaforma yucca costituisce il "cuore pulsante" dell'intero sistema smartdatanet, mediante cui è possibile realizzare soluzioni applicative basate su Internet of Things e Big Data.
Essa permette di:

- interconnettere applicazioni, social network, sistemi e oggetti distribuiti sul territorio;
- raccogliere dati (strutturati oppure semplici) da applicazioni,videocamere o altri dispositivi e consentirne l'elaborazione e l'analisi avanzata;
- fornire una mappa integrata dell'ecosistema digitale;
- abilitare la realizzazione di soluzioni end-to-end finali.

Le informazioni possono essere raccolte ed analizzate per trattare in "near-real-time" fenomeni fisici e sociali anche complessi (ad esempio, si può seguire l'evoluzione di una manifestazione o di una calamità naturale).
È possibile realizzare elaborazioni statistiche avanzate, fino alle moderne tecniche previsionali che consentono, dato un set di informazioni noto, di prevedere la trasformazione di uno o più fenomeni nel tempo, con possibilità di autoapprendimento (ad esempio, l'evoluzione dei flussi di traffico, dei consumi energetici in base al meteo, o di una specifica patologia tra la popolazione).

Di seguito vengono elencate le librerie mediante cui è possibile effettuare l'interazione con la piattaforma. I sorgenti delle librerie e delle reference implementation sono in open source su GITHUB.

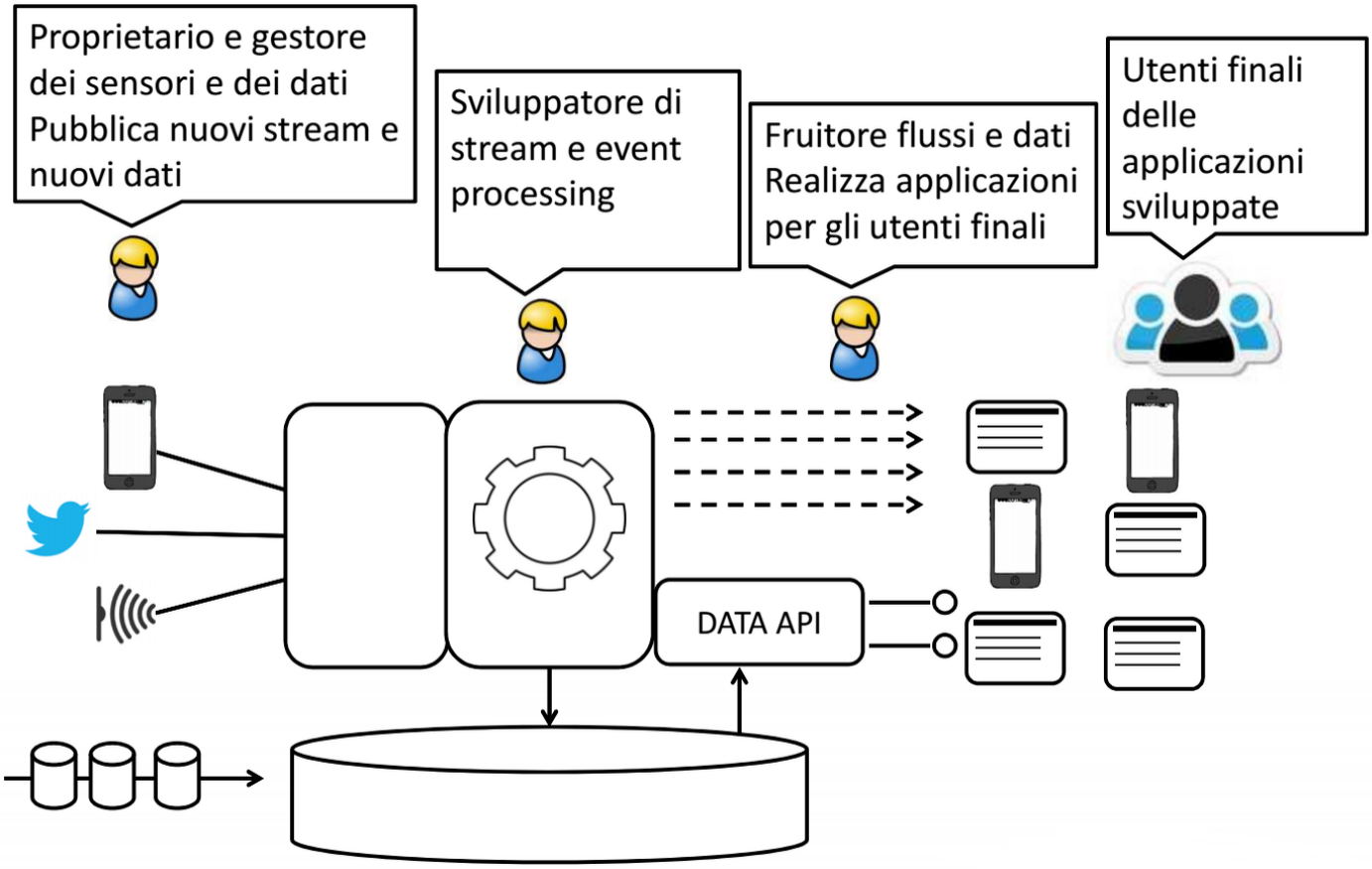
1. Linux Library C for SDP: libreria che fornisce le API per lo sviluppo di un fornitore o di un fruitore di dati verso e da Yucca Platform in MQTT su un sistema Linux
2. Arduino reference implementations: libreria che fornisce le API per lo sviluppo di un fornitore o di un fruitore di dati verso e da Yucca Platform in MQTT su un sistema Arduino, anche in modalità sicura. La modalità sicura non è ancora supportata dalla versione attuale (DEV-0.7) della Piattaforma
3. Server Reference implementation: soluzione che consente di emulare parzialmente la Piattaforma in locale, in

modo da poter semplificare lo sviluppo di smart object e web application

4. WebSocket Singleton Reconnecting Library: libreria javascript che aggiunge diverse funzionalità a STOMP.js come riconnessione, singleton, etc.

1.2 Architettura della piattaforma

Lo schema seguente raffigura l'architettura ed il funzionamento della piattaforma:



L'unità che elabora gli stream (filtra,salva e redireziona i flussi) provenienti dagli smart object è detta motore e permette di effettuare le interrogazioni sui dati in tempo reale. I dati già esistenti, chiamati anche dati offline in quanto non arrivano da un dispositivo (o applicazione), e gli stream vengono memorizzati nello storage dei big data detto "data lake".

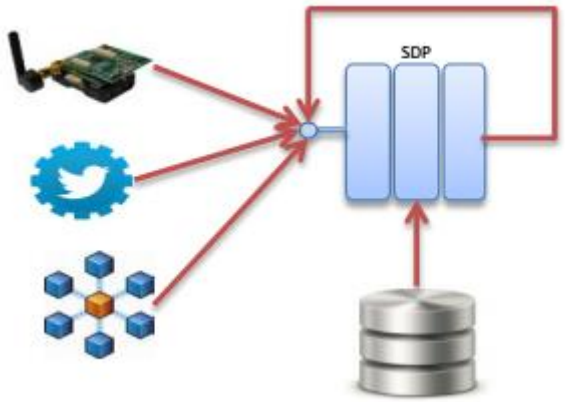
2 Specifiche tecniche

Sono previste 4 modalità di integrazione:

1. Invio di eventi verso SDP (tipicamente da sensori, feed social Twitter o applicazioni)
2. Invio di dataset completi su SDP (in modalità batch sulla piattaforma dataset nei formati più comuni: CSV, Excel, Dump di DB relazionali)
3. Fruizione di eventi in streaming esposti da SDP
4. Fruizione dei dati presenti nella SDP tramite servizi REST(in formato JSON/XML tramite http/s)

Le tipologie di dataset presenti nella piattaforma SDP sono le seguenti :

- dataset contenenti eventi provenienti dai sensori (un sensore non è altro che un dispositivo collegato ad internet che invia informazioni verso la piattaforma;
- dataset contenenti eventi provenienti da Twitter;
- dataset contenenti messaggi provenienti da applicazioni;
- dataset frutto di elaborazioni realtime/offline interne alla piattaforma;
- dataset provenienti da fonti esterne (quali ad esempio opendata).



2.1 Disponibilità dei dataset e dei flussi inviati su SDP

Importante è sottolineare che indipendentemente da dove arrivano le fonti di acquisizione dei flussi inviati si SDP, questi dovranno essere nella piena ed esclusiva disponibilità del fornitore. Questo significa naturalmente che sarà sua la responsabilità avere acquisito tutte le informazioni e consensi da soggetti terzi, necessarie ed indispensabili per poter integrare in SDP dataset e flussi di dati.
In caso di constatazioni sarà il fornitore a rispondere in caso di danni o richieste di risarcimento nel caso in cui per esempio venga violato qualche diritto per un uno non legittimo o non autorizzato dei dati. Il fornitore che deciderà di

esercitare questa facoltà dovrà adottare una o più licenze. Queste licenze dovranno essere pubblicate sulla piattaforma in modo tale da essere visibili a tutti quelli che utilizzeranno questa piattaforma.

2.2 Panoramica di protocolli e dei formati

Naturalmente questa piattaforma utilizzerà dei protocolli in ingresso e in uscita come si può vedere dall'immagine sottostante:

JSON: acronimo di JavaScript Object Notation, è un formato adatto per lo scambio dei dati in applicazioni client-server.
HTTP: L'HyperText Transfer Protocol (HTTP) (protocollo di trasferimento di un ipertesto) è usato come principale sistema per la trasmissione d'informazioni sul web con un'architettura di tipo client-server, il server generalmente resta in ascolto delle richieste dei client sulla porta 80 usando il protocollo TCP a livello di trasporto.
RTSP: Real Time Streaming Protocol è un protocollo di rete usato in sistemi informatici di comunicazione rivolto al controllo di server per lo streaming multimediale, ossia a stabilire e gestire sessione di streaming tra server e client. La maggior parte dei server RTSP si affida per la trasmissione di flussi multimediali al Real-Time Transport Protocol (RTP) in unione al Real-Time Control Protocol(RTCP).
MQTT : è un protocollo di messaggistica leggero posizionato in cima a TCP/IP. È stato disegnato per le situazioni in cui è richiesto un basso impatto e dove la banda è limitata

2.3 Specifiche per l'invio di dati alfanumerici

In questa categoria viene descritto il protocollo per l'invio dei dati alfanumerici. Gli eventi possono provenire dalle seguenti fonti:

1. Sensori o reti di sensori
2. Applicazioni
3. Feed Social (Twitter)

Per quanto riguarda gli eventi provenienti dai Feed Social (Twitter), essi saranno direttamente recuperati dalla piattaforma, a fronte di una configurazione opportuna.

Modello dei dati per le applicazioni

Un flusso di dati (definito come stream) è una sequenza di messaggi inviati da una applicazione. A questo flusso vengono associati i seguenti metadati;

Attributo	Descrizione	Tipo
id	Identificatore del flusso (univoco solo per una applicazione)	Stringa
application	Identificatore dell'applicazione	Stringa (7.1)
description	Descrizione libera	Stringa
tags	Tag utili per la ricerca	Vettore di stringhe
visibility	Visibilità del flusso	"public" "private"
domain	Il nome di un progetto/consorzio/network di appartenenza del flusso	Stringa
license	Tag Creative Commons che identifica la licenza sottoposta al flusso (e. "CC0 1.0", "CC BY 4.0")	Stringa
disclaimer	Dichiarazione di limitazione di responsabilità	Stringa
copyright	Nota di copyright	Stringa
components	Un vettore che contiene la descrizione di tutte le componenti del flusso	Vettore di oggetti Component

Tabella 1- Oggetto Stream

Lo stream contiene un vettore di oggetti component cosi definiti :

Attributo	Descrizione	Tipo
id	Identificatore della componente	Stringa
event	Nome dell'attributo	Stringa
type	Permette di risalire al dominio del dato (ad esempio numeri reali, interi positivi, enumerazione e stringhe)	Uno tra: int, long, double, float, string, boolean

Tabella 2- Oggetto Component

Ogni oggetto message contiene oltre allo stream, l'identificativo dell'applicazione e un vettore di oggetti business event.

Attributo	Descrizione	Tipo
stream	Identificatore del flusso	Stringa
application	Identificatore dell'applicazione	Stringa
values	Valori da inviare	Vettore di oggetti Business Event

Tabella 3- Oggetto Message

Attributo	Descrizione	Tipo
time	Data e ora dell'evento di business	Stringa ([ISO8601])
components	I valori associati alle diverse componenti	Mappa associativa in cui la chiave è l'id del componente e il valore è il valore dell'attributo di interesse

Tabella 4- Oggetto Business Event

Modello dei dati per gli eventi provenienti da sensori

Un sensore che invia dati numerici alla SDP (osservazioni) genera un flusso di dati.

	Dati provenienti da applicazioni	*Dati provenienti da sensori*
Metadata	Stream	Stream
Metadata	Component	Component
Metadata	Application	Sensor
Runtime data	Message	Observation
Runtime data	Business Event	Measure

A questo flusso vengono accostati i seguenti metadati:

Attributo	Descrizione	Tipo
id	Identificatore del flusso (univoco solo per un dato sensore)	Stringa
sensor	Identificatore del sensore	Stringa (7.1)
description	Descrizione libera	Stringa
tags	Tag utili per la ricerca	Vettore di stringhe
visibility	Visibilità del flusso	"public" "private"
fps	Associando al concetto di frame una singola osservazione, questo parametro (condiviso anche con i flussi multimediali) permette di dare evidenza della frequenza di acquisizione della misura	Numero
domain	Il nome di un progetto/consorzio/network di appartenenza del flusso (e. "MeteoNetwork")	Stringa
license	Tag Creative Commons che identifica la licenza sottoposta al flusso (e. "CC0 1.0", "CC BY 4.0")	Stringa
disclaimer	Dichiarazione di limitazione di responsabilità	Stringa
copyright	Nota di copyright	Stringa
components	Un vettore che contiene la descrizione di tutte le componenti del flusso	Vettore di oggetti Component

Tabella 5 Oggetto Stream

Nella descrizione di un flusso (Stream) è presente un elenco di component (a cui è associata un'unità di misura e la tolleranza) che collettivamente forniscono la rappresentazione di una misura complessa (Measure).

Attributo	Descrizione	Tipo
id	Identificatore della componente (univoco solo all'interno della misura)	Stringa
unit	Unità di misura	Stringa
allowance	Tolleranza espressa come valore assoluto dell'errore sulla misura	Numero
event	Tipo di fenomeno misurato (e. "Wind Speed")	Stringa
type	Permette di risalire al dominio del dato (ad esempio numeri reali, interi positivi, enumerazione e altri)	Uno tra: int, long, double, float, string, boolean

Tabella 6- Oggetto Component

Ogni oggetto Observation contiene lo stream definito precedentemente, l'identificativo del sensore e una sequenza di misure.

Attributo	Descrizione	Tipo
stream	Identificatore del flusso	Stringa
sensor	Identificatore del sensore	Stringa
values	Valori letti	Vettore di oggetti Measure

Tabella 7- Oggetto Observation

Ogni oggetto Measure riporta data e ora della misurazione, i valori delle singole componenti e il campo Validity che generalmente non deve essere fornito dal sensore, ma viene impostato dalla SDP sulla base di alcuni algoritmi di elaborazione;

Attributo	Descrizione	Tipo
time	Data e ora della misura	Stringa ([**ISO8601**])
components	I valori associati alle diverse componenti	Mappa associativa in cui la chiave è l'id del componente e il valore è la misura
validity	Validità della misura	"valid" "erroneous" "doubtful" "unknown"

Tabella 8- Oggetto Measure

2.4 Protocollo per l'invio di dati multimediali

I formati per la codifica audio/video supportati sono MPEG-4 Part 10/AVC Baseline/Main Profile (H.264), Motion JPEG.
I flussi video delle webcam devono essere messi a disposizione della piattaforma tramite un server RTSP o http (protocollo http-push).
Nel secondo caso si utilizza un oggetto MIME per distribuire il flusso MJPEG, mandato dal server al client senza esplicita richiesta di quest'ultimo.

2.5 Specifiche per la fruizione di eventi esposti da SDP

Per quanto riguarda la fruizione degli eventi, anche in questo caso, è necessario effettuare la divisione in alfanumerici o multimediali. Per quanto riguarda gli eventi alfanumerici, la piattaforma SDP consente la fruizione degli eventi in

modalità "stream": il fruitore si sottoscrive ad un flusso di informazioni e viene notificato dalla piattaforma a fronte di un nuovo evento (modalità publish-subscribe). La sottoscrizione può essere di due tipologie: "server" o "client". Nella sottoscrizione "server" è necessario dichiarare preventivamente e staticamente l'endpoint sottoscritto. In questo scenario la piattaforma richiama l'endpoint fornendo le informazioni dell'evento generato sul flusso sottoscritto. Invece nella tipologia "client" il fruitore è un client "finale" che effettua una sottoscrizione richiamando un' API specifica e attende le informazioni relative ai nuovi eventi.
Per quanto riguarda invece lo stream relativo agli eventi multimediali, viene esposto seguendo il formato webM.

2.6 Rest API per l'accesso a dati della SDP tramite servizi

La piattaforma sarà dotata di un sito divulgativo che consente la ricerca e la sottoscrizione dei dataset disponibili mediante delle apposite API. Queste consentono, scelto un dataset, di cercare e recuperare le informazioni di presenti (secondo la specifica OData v2) tramite http/s in formato JSON/XML.

2.7 Sicurezza

Naturalmente tutte le comunicazioni che vanno verso la piattaforma dovranno contenere le informazioni necessarie per autenticare la sorgente (da parte della SDP).
Nella versione attuale è possibile utilizzare l'autenticazione basic (ovvero quella basata sull'invio di username e password) su HTTP o MQTT.
In versioni successive i client avranno a disposizione il protocollo OAuth 2.0 ([OAUTH]) utilizzando un token d'accesso, con gli stessi protocolli esposti sopra.
Eventualmente HTTP e MQTT possono essere veicolati su SSL/TLS, questa configurazione prevede l'autenticazione del server e la cifratura del canale di comunicazione.
Nel caso dei sensori l'identificazione è realizzata tramite un UUID (Universally Unique Identifier) a 128 bit univoco per ognuno di essi.

3 Sintesi operativa

Il progetto QUIES prevede l'utilizzo di sensori a basso costo per il monitoraggio acustico. Sono stati stimati 30 sensori in totale intercambiabili tra fissi e mobili e un numero attualmente non quantificabile di dispositivi, su cui verrà installata l'applicazione in un'ottica di crowdsourcing.
Inoltre si monitoreranno zone differenti in tempi diversi, in modo da poter distribuire maggiormente i sensori all'interno di una zona.

La durata della batteria per i sensori fissi è di 10 giorni (con l'utilizzo del wifi), invece con i sensori mobili il problema dell'autonomia è poco rilevante, poichè vengono ricaricati periodicamente (prima e dopo l'itinerario effettuato dall'utilizzatore). Per i sensori fissi si utilizza un tempo di campionamento pari ad 1 minuto in modo da avere un maggiore risparmio energetico, poichè non necessita di geolocalizzazione (essendo la sua postazione fissa). Nel secondo caso (sensori mobili e smartphone) l'utente effettua uno spostamento per misurare il tratto effettuato, quindi è necessario campionare con una frequenza maggiore (tra 1 e 10 secondi) e ad ogni dato deve essere associata l'informazione relativa alla posizione del sensore mobile.

Di seguito vengono descritti tre possibili scenari per l'accesso alla piattaforma:

1. Viene previsto l'utilizzo di un server centrale che farà da "tramite" tra i sensori e la piattaforma;
2. Variante del primo scenario con accesso alternativo alla piattaforma.
3. Ogni sensore diventa uno smartobject che comunica direttamente i dati misurati alla piattaforma.

3.1 Comunicazione dei sensori con SDP attraverso server intermedio

In questo contesto i dispositivi non invieranno direttamente i dati alla piattaforma, in quanto la comunicazione sarà effettuata con l'ausilio di un server intermedio, che si occuperà anche di creare la mappa del rumore. Questo comporta che i sensori e gli smartphone possono utilizzare un proprio formato dati, comunicando con il server mediante un protocollo accordato tra le due parti. I dati verranno convertiti in formato json dal server e trasmessi in streaming sulla piattaforma utilizzando le API messe a disposizione mediante http. Opzionalmente, si potrebbe assumere che gli smartphone, essendo già dotati di un notevole potenza di calcolo, possano già effettuare la conversione del dato sorgente in formato json, andando a diminuire il carico di lavoro del server.
Si prevede di attivare uno smartobject di tipo Application he contterrà i dati unificati di tutti i sensori.

3.2 Variante del primo scenario (accesso non in streaming)

In questo scenario, come nel caso precedente, si fa utilizzo di un server intermedio tra i dispositivi e la piattaforma. La differenza sostanziale consiste nel tipo di accesso alla piattaforma smartdatanet: i dati non vengono inviati in modalità streaming, ma caricati tramite dataset offline. Il server intermedio si occuperà della creazione di un file csv,xls o xml contenente i dati ricevuti dai sensori. Anche la fruizione avverrà offline scaricando l'intero dataset.

3.3 Comunicazione diretta dei sensori con la SDP

I sensori comunicheranno direttamente con la piattaforma smartdatanet attraverso uno dei protocolli messi a disposizione (dalla SDP) MQTT o HTTP e si occuperanno di convertire i dati raccolti in formato JSON.

La piattaforma smartdatanet si occuperà degli aspetti di rielaborazione e validazione del dato grezzo, poi ci sarà un server di raccolta dati che prenderà questi dati rielaborati per salvarli e realizzare la mappa del rumore.
Ci saranno 30 smartobject di tipo Sensor e un numero variabile di smartphone che comunicheranno con la piattaforma (stream di dati Application).

Riguardo agli smartphone un aspetto da considerare è il numero degli stessi, se questo è sufficientemente basso allora è plausibile la comunicazione diretta con smartdatanet (registrando ogni smartphone presso la piattaforma). Altrimenti è necessaria l'introduzione di un server intermedio per aggregare le misurazioni effettuate dai telefonini, per avere un unico flusso dati verso smartdatanet (oltre a quelli dei sensori).

Il vantaggio di un'architettura di questo tipo è di diminuire la dipendenza dal server intermedio per il backup dei dati, poichè si utilizza la SDP.

3.4 Struttura dati

Non è ancora stato definito il tipo di dato che verrà memorizzato sulla piattaforma, tuttavia si possono ipotizzare due tipologie di struttura dati. Nel primo caso viene inviato il dato grezzo acquisito direttamente dal sensore senza ulteriori elaborazioni.

La struttura è composta dai seguenti campi:

- Data: in formato anno, mese, giorno, ora, minuti, secondi, centesimi;
- Durata: tempo di campionamento della misura;
- LAeq,T: livello equivalente A del segnale sonoro campionato su un tempo T (per livelli bassi di rumore);
- LZeq,T: livello equivalente non ponderato del segnale sonoro campionato su un tempo T;
- Latitudine;
- Longitudine;
- Valori per frequenze: contiene la misurazione in decibel del rumore per lo spettro di frequenza tra 20Hz e 20KHz con bande percentuali costanti di 1/3 di ottava (anche 1/1 di ottava).

Questa struttura viene salvata sul server di raccolta in un file con estensione nbf. Per avere una maggiore precisione, si potrebbero integrare dei dati opzionali:

- LCeq,T; livello equivalente C del segnale sonoro campionato su un tempo T (per livelli alti di rumore);
- Segnali max,min,peak con livello equivalente A,Z,C con ponderazione temporale :
    - (S) slow = 1 secondo;
    - (F) fast = 125 millisecondi;
    - (I) impulse = 35 millisecondi quando il segnale aumenta e 1,5 secondi quando diminuisce.

Nel secondo caso il dato viene elaborato e validato dal server di raccolta e successivamente inviato a smartdatanet. I campi della struttura dati si ipotizzano simili a quelli del primo caso, probabilmente si escludono i campi opzionali e si

utilizza uno spettro ridotto di frequenze per il segnale sonoro. Le informazioni su data, durata e geolocalizzazione rimangono invariate.

3.5 Fruizione dei dati dalla piattaforma e dati di contesto

Le misure salvate degli stream saranno interrogabili tramite oData Service API per un periodo di 6 mesi e comunque per non più di 1 milione di misure distinte per lo stream dei sensori fissi (fino a 10 milioni per lo smartobject con i dati unificati). Si stimano due applicazioni distinte per l'interrogazione dei dataset tramite API rest (con una frequenza superiore alle 10 chiamate al secondo) e altrettante come fruitrici agli stream in uscita dalla piattaforma SDP.

In una fase iniziale si utilizzerà la fruizione dei dati tramite streaming per verificare la corretta interazione tra server intermedio e la piattaforma smartdatanet. Inoltre questi dati potranno eventualmente essere processati per la creazione della mappa del rumore.

Si prevede inoltre il caricamento di 20 dataset (tra 10MB e 100MB) di contesto (ad esempio il traffico in una certa zona, informazioni di geolocalizzazione dei singoli sensori fissi) per arricchire i dati ricevuti dagli stream.

3.6 Scenario del progetto QUIES

Nell'ambito del progetto QUIES si è scelto il primo scenario utilizzando quindi un server centrale che farà da "tramite" tra i sensori e la piattaforma. Nel contesto in esame, comporta diversi vantaggi non indifferenti:

- Il sensore può avere una architettura più semplice non dovendo effettuare la conversione in formato json e non utilizzando i protocolli accettati dalla SDP;
- nel caso degli smartphone non è necessario registrarli direttamente presso smartdatanet, questo potrebbe essere problematico nel caso di un numero elevato di utilizzatori (dell'applicazione);
- la rielaborazione e la validazione dei dati è effettuata dal server intermedio, riducendo quindi il volume di dati inviati alla piattaforma;
- permette la diminuzione della mole dei dati nel caso di utilizzo dell'applicazione per smartphone da parte di un elevato numero di persone, potendo rimanere nei limiti di spazio allocabile sulla piattaforma;
- comoda fruizione in streaming dei dati raccolti (nelle fasi iniziali anche per finalità di testing);
- nel secondo scenario è comunque necessario un server aggiuntivo che preleva i dati dalla piattaforma per il backup degli stessi e per realizzare la mappa del rumore;
- grazie al server intermedio si ha una minore dipendenza dalla piattaforma smartdatanet e una maggiore sicurezza contro la perdita dei dati.

Per questi motivi e per come è stato definito il progetto, si è scelto che l'utilizzo di un server intermedio (tra i sensori e la SDP) sia il contesto ideale per procedere all'integrazione della rete di sensori con la piattaforma smartdatanet.