PROGETTO

TESSUTO DIGITALE METROPOLITANO

POR FESR 2014-2020, Azione 1.2.2

(Delibera 66/14 del 13.12.2016, Progetto Complesso area ICT della S3).

Deliverable

Deliverable D3.1 – REFERENCE DESIGN PER SENSORISTICA DIFFUSA (DESIGN+MANUALE)

Data di consegna prevista: 06/2018	**Data di consegna effettiva**: 06/2018
Natura: Rapporto + Open HW / Open SW	**Versione**: 1.0
Livello di Disseminazione (PU)	

Sommario

Uno degli obiettivi principali del progetto TDM è di realizzare un'architettura scalabile per l'acquisizione, l'integrazione e l'analisi di dati provenienti da sorgenti eterogenee in grado di gestire i dati generati da un'area metropolitana estesa. Una parte di questi dati proverrà da sensoristica diffusa sul territorio. Il progetto prevede la realizzazione di un reference design, di uso generale, per la gestione periferica dei sensori e la trasmissione dei segnali da essi raccolti. Questa piattaforma è denominata *Edge Gateway*. Questo documento descrive l'architettura hardware/software dell'Edge Gateway e delle sue componenti preposte all'acquisizione e pre-processamento delle misure acquisite dai sensori distribuiti del progetto, e il loro inoltro verso il sistema centrale di raccolta, memorizzazione e analisi. E' inoltre incluso in questo deliverable un manuale di installazione.

L'Edge Gateway qui descritto è di uso generale, ed è stato specificatamente realizzato per essere utilizzato con altre infrastrutture basate su FIWARE diverse da TDM. Può inoltre essere utilizzato su altri sistemi quali, ad esempio, i cloud IoT Amazon AWS, IBM Watson, Microsoft Azure per costruire soluzioni OASC-compliant. Tutto il software realizzato è reso disponibile su GITHUB all'indirizzo: https://github.com/tdm-project.

REGIONE AUTÒNOMA DE SARDIGNA
REGIONE AUTONOMA DELLA SARDEGNA

POR FESR SARDEGNA 2014-2020

UNIONE EUROPEA
Fondo europeo di sviluppo regionale

REPUBBLICA ITALIANA

Redazione			
	Name	Partner	Data
Autore	M. Gaggero	CRS4	01/06/2018
Approvato da	G. Zanetti & E. Gobbetti	CRS4	04/06/2018

Storia e Contributi			
Vers.	Data	Commento	Autori
V0.5	22/04/2018	Primo Draft condiviso	M. Gaggero (CRS4)
V0.6	11/05/2018	Prima versione completa con manuale	M. Gaggero, G. Zanetti (CRS4)
V0.7	14/05/2018	Sensoristica specializzata	L. Massidda, M. Marrocu (CRS4)
V0.8	22/05/2018	Revisione e finalizzazione manuale	G. Busonera / M. Delrio (CRS4)
V0.9	25/05/2018	Revisione globale e integrazione	M. Gaggero / E. Gobbetti / G. Zanetti (CRS4)
V1.0	01/06/2018	Versione finale	M. Gaggero / E. Gobbetti / G. Zanetti (CRS4)

Sommario

1. INTRODUZIONE

Uno degli obiettivi principali del progetto TDM è di realizzare un'architettura scalabile per l'acquisizione, l'integrazione e l'analisi di dati provenienti da sorgenti eterogenee in grado di gestire i dati generati da un'area metropolitana estesa. Una parte di questi dati proverrà da sensoristica diffusa sul territorio. Al fine di gestire al meglio questa sensoristica, il progetto prevede la realizzazione di un *reference design*, di uso generale, per una piattaforma per la gestione periferica dei sensori e la trasmissione dei segnali da essi raccolti. Questa piattaforma è denominata *Edge Gateway*.

Questo documento costituisce la descrizione dell'architettura hardware/software dell'Edge Gateway e delle sue componenti preposte all'acquisizione e pre-processamento delle misure acquisite dai sensori distribuiti del progetto, e il loro inoltro verso il sistema centrale di raccolta, memorizzazione e analisi.

L'Edge Gateway qui descritto è di uso generale, ed è stato specificatamente realizzato per essere utilizzato con altre infrastrutture basate su FIWARE diverse da TDM. Può inoltre essere utilizzato su altri sistemi quali, ad esempio, i cloud IoT Amazon AWS, IBM Watson, Microsoft Azure per costruire soluzioni OASC-compliant.

In particolare, il design descritto si basa su piattaforme hardware facilmente reperibili sul mercato e delle quali è disponibile liberamente tutta la documentazione per il loro funzionamento e integrazione in altri prodotti (la scheda BeagleBone Black, descritta dopo, ad esempio è distribuita come Open Hardware, ossia il suo intero design può essere usato nell'architettura di altri oggetti). Anche le stazioni di misura utilizzate assieme all'Edge Gateway sono distribuite, oltre che come prodotto finito, anche come Open Hardware mentre i sorgenti dei firmware relativi sono liberamente disponibili e distribuiti sotto licenza Open Source. L'architettura software pensata per l'Edge Gateway è a sua volta basata su Open Source, protocolli standard e il codice d'esempio è liberamente utilizzabili e modificabile. In particolare, le comunicazioni avvengono attraverso il protocollo standard MQTT, mentre il formato dei messaggi è compatibile con gli standard FIWARE e OASC.

Tutto il software realizzato è reso disponibile su GITHUB all'indirizzo: https://github.com/tdm-project.

Di seguito verrà presentata una breve panoramica dell'architettura generale del progetto, l'architettura sviluppata per il software, i requisiti per la piattaforma hardware i componenti scelti per il prototipo.

2. OVERVIEW DELL'ARCHITETTURA DI ACQUISIZIONE ED AGGREGAZIONE

Negli ultimi anni, l'offerta di servizi messi a disposizione dalle città ai propri i cittadini ha visto una rapida espansione grazie all'utilizzo, affianco ai tradizionali veicoli informativi, di nuovi strumenti digitali. Inoltre anche la quantità e la qualità di informazioni disponibili è notevolmente aumentata, in parte per il costante impegno delle istituzioni a pubblicare i propri *dataset* come *OpenData* e in parte per la creazione di nuove informazioni, elaborate o rappresentate in altre forme, partendo da dati iniziali. Il ruolo stesso del cittadino è ribaltato. Da *utente* dei servizi cittadini diviene *sorgente* delle informazioni che questi servizi utilizzano.

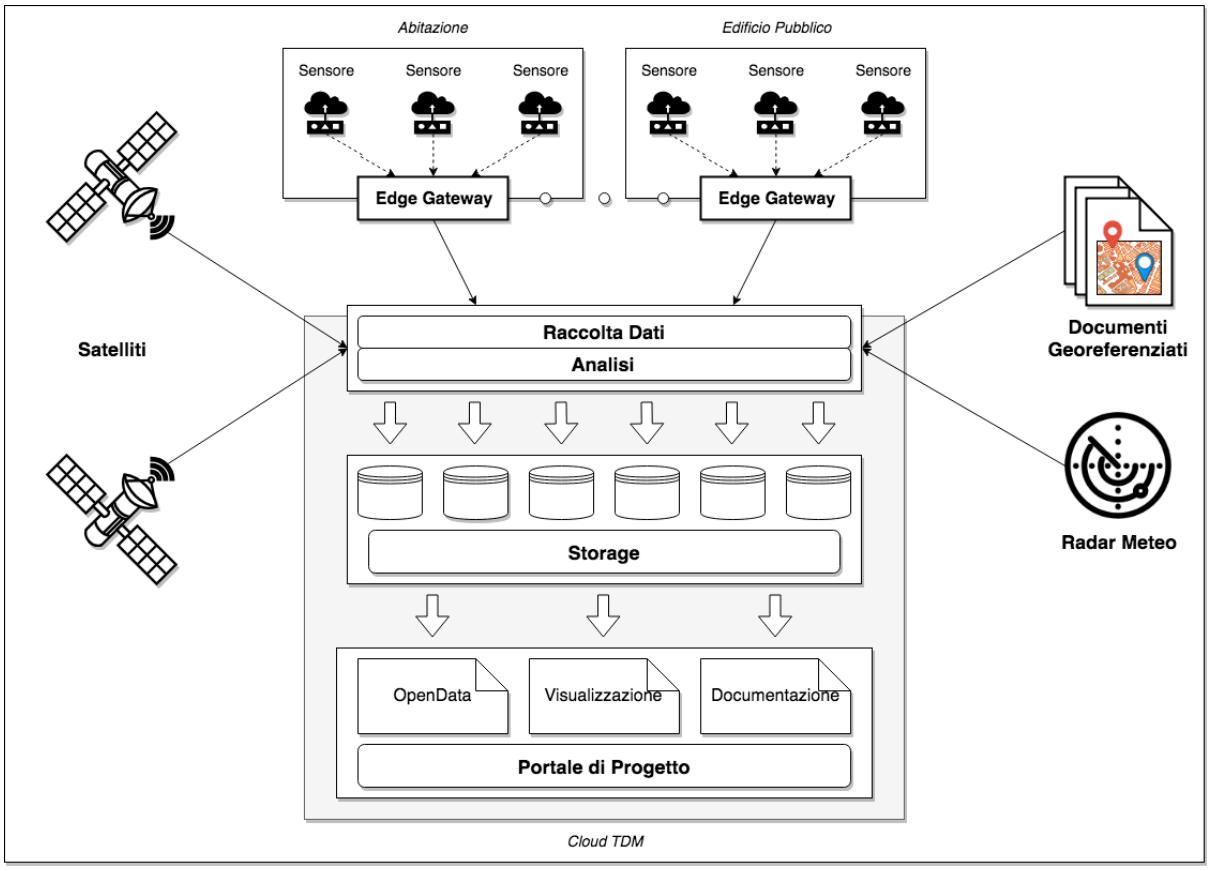
Temi quali sicurezza ambientale, consapevolezza energetica e del patrimonio storico/culturale coinvolgono sempre più il cittadino come *fruitore* dell'informazione, ma soprattutto come *fornitore* di dati. Il dato diviene quindi estremamente ubiquo, generato in modi diversi, con risoluzioni e qualità differenti. A questa nuova sorgente si affiancano poi le tradizionali fonti, spesso più precise e standard ma più costose e meno flessibili. Per trarre nuovi servizi occorre quindi porre assieme nuovi tipi di informazioni con informazioni tradizionali, nuovi canali con i classici.

Tra le attività di qualsiasi sistema scalabile di acquisizione dati c'è, quindi, l'utilizzo di piattaforme hardware a basso costo distribuibili sul territorio. Nell'ambito del progetto TDM, in particolare, utilizzeremo questi sensori per monitorare parametri meteo-ambientali e di consumo elettrico.

il metodo di acquisizione di tutti questi dati deve essere calibrato sul tipo di dato e sullo strumento che lo genera. In particolare le misure da sensori possono essere acquisite da dispositivi di tipo diverso e distribuite su un ampio territorio e si pone quindi il problema si trasmettere efficacemente la mole di dati generata in un formato di messaggio *standard* per la raccolta e l'immagazzinamento.

Si rende necessario perciò porre tra sensori sul campo e *cloud* di acquisizione un dispositivo di raccolta e inoltro dei dati che disaccoppi l'acquisizione della misura dall'invio e sopperisca ai limiti dei sensori stessi. Limitate capacità della rete, congestione, interruzioni, ridotte risorse computazionali e di memorizzazione, inadeguati o assenti strumenti di sicurezza nelle comunicazioni, dispositivi di connessione assenti sono alcuni dei problemi tecnologici che s'incontrano impiegando ad esempio sensori economici facilmente accessibili ai comuni cittadini. E di questi problemi occorre tenere conto in fase di progettazione di un sistema pensato per essere impiegato anche con dati e in campi non ancora immaginati. All'interno del progetto TDM, questo componente di raccolta locale e inoltro prende il nome di Edge Gateway.

L'integrazione di dati provenienti da un Edge Gateway all'interno di un'architettura scalabile di aggregazione, processamento e restituzione è esemplificata nella seguente figura, che mostra le diverse componenti del progetto TDM.



La raccolta centrale dei dati provenienti dall'Edge Gateway, così come di altre sorgenti quali immagini satellitari, radar meteorologico, mappe e altri dati statici georeferenziati avviene nel *cloud* TDM utilizzando una *Lambda Architecture*. I dati, appena giunti, vengono pre-processati in quella che è chiamata elaborazione *in stream*, e che risponde a necessità *real-time* date da aggiornamenti dei dati. Allo stesso tempo, i dati ricevuti, assieme a quelli pre-processati vengono salvati nei sistemi di *storage BigData* dove rimangono disponibili per lavorazioni successive. Parte di questi poi alimentano elaborazioni più lunghe, che richiedono più dati e risultati meno tempestivi, le cosiddette elaborazioni *in batch*. Elaborazioni batch sono ad esempio le previsioni meteorologiche a breve periodo (*Nowcasting*), le analisi storiche dei consumi energetici, la creazione di modelli di per la visualizzazione. Come per tutti gli altri dati, anche i risultati delle elaborazioni in batch sono archiviate nello storage. I dati, poi, possono essere restituiti attraverso componenti di visualizzazione integrate in un *portale integrato di progetto*.

L'Edge Gateway qui descritto è di uso generale, ed è stato specificatamente realizzato per essere utilizzato con altre infrastrutture basate su FIWARE diverse da TDM o con cloud IoT. In particolare, inserendo un edge gateway standardizzato, con capacità di processamento e memorizzazione locali, tra la sensoristica ed il cloud, si ottengono notevoli vantaggi, tra i quali la possibilità di una prima aggregazione locale dei segnali provenienti da più sensori e di trasmetterli verso l'infrastruttura di raccolta dati e analisi. Inoltre, assicurando la presenza negli aggregatori locali di una sufficiente potenza di calcolo e storage, a questi compiti si aggiunge anche quello di una prima memorizzazione dei dati ed elaborazione e filtraggio. Questa capacità, permette quindi di ridurre la quantità di dati trasferiti e la frequenza di invio, nonché di fornire agli stessi dispositivi collegati una risposta in tempo reale, per applicazioni di controllo e non solo monitoraggio. Inoltre, è possibile anche poter gestire un primo accesso al dato senza nessuna trasmissione in rete, il che permette di sviluppare diversi livelli di accesso e di controllo della privacy.

3. CARATTERISTICHE GENERALI DELLLA PIATTAFORMA HARDWARE DELL'EDGE GATEWAY

3.1 CONSIDERAZIONI GENERALI

La piattaforma hardware per l'Edge Gateway, l'Edge Gateway, deve essere in grado di poter far eseguire correttamente le funzionalità di aggregazione, processamento e trasmissione sopra descritta, e permettere l'eventuale integrazione di ulteriori componenti future.

Oltre a questi che sono requisiti tecnici, la scelta della piattaforma hardware deve tenere conto anche di altri requisiti legati all'utenza prevista. Il target di utenza per i sensori, oltre a quella istituzionale, è, infatti, sempre più costituito da studenti di scuola superiore e privati cittadini che volontariamente intendono acquistare o assemblare i dispositivi da sé. Specifiche sperimentazioni in questo senso saranno fatte nel progetto TDM, ma questa modalità di utilizzo è sempre più diffusa a livello mondiale.

E' necessario, quindi, che la piattaforma hardware abbia un costo ridotto e sia facilmente reperibile, possibilmente con una ampia comunità di sviluppatori e utenti possibile. Deve inoltre essere libera da vincoli commerciali su royalties e licenze, e possibilmente avere una connotazione 'didattica'.

In particolare due di questi requisiti generali, il basso costo e l'ampia comunità di utenti restringono il campo delle scelte a:

- Single Board Computer (SBC) hobbistiche per la piattaforma, e
- distribuzione Linux come sistema operativo.

3.2 REQUISITI I/O E CONNETTIVITÀ

All'Edge Gateway è assegnato come ruolo principale quello di raccogliere e aggregare misure e i dati acquisiti localmente dai diversi sensori e stazioni di rilevamento sul campo e predisporli per l'invio verso il cloud per la successiva archiviazione ed elaborazione.

Dal punto di vista delle capacità di I/O, la piattaforma hardware deve essere quindi in grado di comunicare con i sensori utilizzando sia semplici bus elettrici, sia protocolli di rete più sofisticati. In particolare, per i sensori direttamente collegati al dispositivo Edge, è possibile usare principalmente i protocolli I2C, USART e custom implementati su GPIO, mentre per le unità di rilevazione remote quali i misuratori di consumo energetico e le stazioni meteo/qualità dell'aria, la modalità più semplice ed efficace è la connettività WiFi attraverso rete domestica, qualora presente nel luogo di installazione, o eventualmente creandola attraverso l'Edge Gateway stesso. Aspetto da tenere in considerazione inoltre è quello della possibilità di future espansioni: requisito desiderato e non vincolante per la piattaforma hardware è quindi la presenza di ulteriori bus elettrici e protocolli di comunicazione tra dispositivi embedded.

Per quanto riguarda le comunicazioni dall'Edge Gateway verso un Cloud, quale quello del progetto TDM, si è scelto di usare connettività Internet nelle installazioni da questa raggiunta mentre per installazioni in aree rurali o non ancore servite è possibile prevedere l'uso di connettività LTE attraverso chip specifici o modem/dongle USB.

Riassumendo, la piattaforma hardware per l'Edge Gateway **deve** avere le seguenti periferiche:

- bus/porta ***I2C***;
- bus/porta ***UART/USART***;
- esporre pin ***GPIO***.

La presenza dei seguenti protocolli, sebbene non vincolante, è un desiderata:

- bus/porta ***CAN***;
- porte/periferiche ***ADC***;
- porte con uscita ***PWM***;
- uscite ***Video*** e/o controller ***LCD***;

Deve avere le periferiche di rete:

- ***WiFi***;
- ***Ethernet***;
- ***USB*** Host.

3.3 REQUISITI COMPUTAZIONALI

Per quanto riguarda i requisiti computazionali, ossia quei requisiti che permettono all'architettura software dell'Edge Gateway di poter essere eseguita efficacemente, questi sono di tre tipi: il processore e la potenza di calcolo, la quantità di memoria RAM, e la capacità di memorizzazione.

I requisiti per il processore dell'Edge Gateway sono i seguenti:

- deve essere supportato dal kernel Linux ed essere in grado di eseguire un sistema operativo Linux completo;
- avere un basso consumo energetico per permettere un utilizzo tramite accumulatore/batteria a bassa tensione;
- avere un buon supporto software e possibilmente un'ampia comunità di utenti e sviluppatori;
- avere SDK e compilatori disponibili gratuitamente e royalty free;
- avere un basso costo e schede di sviluppo complete accessibili;
- data la scelta architetturale di usare più microservizi in esecuzione concorrente, il processore deve possedere più di un core.

Molti processori per applicazioni embedded sono in grado di eseguire un sistema operativo Linux completo. Tra questi, la famiglia degli ARM-CORTEX offre diverse scelte, sia come processori che come schede di sviluppo, che soddisfano questi requisiti.

Per quanto riguarda la memoria RAM, valgono diversi tra i punti precedenti. La dotazione di RAM per l'Edge Gateway deve:

- essere sufficiente per eseguire un sistema operativo Linux completo;
- avere un ridotto consumo energetico;
- avere una dimensione sufficiente per servire più microservizi concorrenti in esecuzione su container Docker;
- supportare l'accesso da più core.

Per storage su Edge Gateway si intende il supporto di memorizzazione sia per il sistema operativo e gli applicativi Edge, sia per la memorizzazione dei dati raccolti nel tempo dai vari sensori e stazioni di misura collegati. Più in dettaglio, i requisiti per lo storage sono:

- economico, accessibile e flessibile, ossia soluzioni reperibili facilmente nel mercato, in diverse capacità e facilmente rimpiazzabili e aggiornabili;
- capacità sufficiente per contenere il sistema operativo, gli applicativi Edge e uno storico ragionevolmente ampio di dati raccolti;
- ampia velocità di trasferimento dei dati.

La soluzione più diffusa che risponde a queste caratteristiche è quella della tecnologia SD, e, pertanto si considera che la soluzione hardware per l'Edge Gateway dovrà possedere il supporto per schede SD/microSD. Per via empirica si è

stabilito che una dimensione di 8GB, classe 10, sia sufficiente per gli scopi dell'Edge Gateway, tuttavia, data la disponibilità di SD di capacità superiori con costi ridotti, si consiglia la dimensione minima da 16GB, classe 10.

3.4 DISCUSSIONE

I requisiti sopra indicati possono essere posseduti da molte soluzioni hardware, con vari rapporti costo-prestazioni. Alla seguente sezione, descriviamo l'architettura software realizzata, che si appoggia sui requisiti sopra esposti e non ha dipendenze sulla soluzione specifica scelta. Alla sezione 5 discuteremo, invece, l'implementazione hardware del prototipo di riferimento.

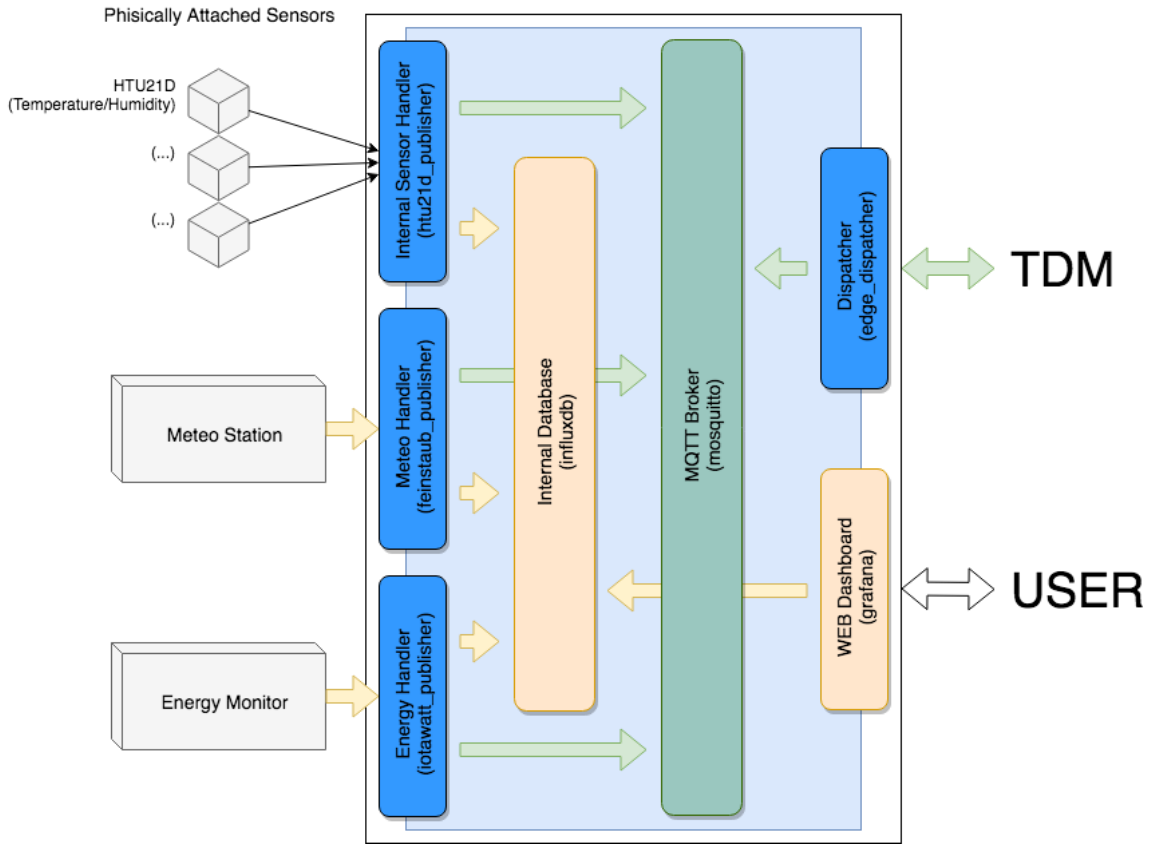
4. LA PIATTAFORMA SOFTWARE DELL'EDGE GATEWAY

4.1 ARCHITETTURA GENERALE

L'Edge Gateway ha il compito di interfacciarsi con i sensori locali o stazioni remote di misura presenti nell'edificio in cui è installato. Sebbene in via prototipale si sia identificato un set di questi da usare per sviluppo e test, lo scopo dell'Edge Gateway è quello di permettere una integrazione trasparente di nuovi sensori, tipi di dato, metodiche di interfacciamento. Il suo software deve perciò permettere una elevata flessibilità e al tempo stesso una buona resilienza ad errori e problemi indotti da nuove componenti.

L'architettura software scelta per la realizzazione dell'Edge Gateway è definita 'Choreography of loosely-coupled, containerized microservices', ossia è una composizione di microservizi non strettamente accoppiati distribuiti in application container. Il cuore dell'architettura è quindi costituita da una rete di microservizi collegati tra loro in maniera tale che la non disponibilità di uno o più di essi non infici il funzionamento dei restanti e del sistema nel suo insieme.

Una implementazione completa di riferimento è stata realizzata nel quadro del progetto TDM ed è resa disponibile su GITHUB all'indirizzo: https://github.com/tdm-project.



4.2 I MICROSERVIZI

In un'applicazione monolitica, tutte le funzionalità sono implementate in una singola unità eseguibile. Tutte le componenti interne condividono risorse quali ad esempio lo stesso spazio di memoria e gli stessi file handler, e un bug in uno dei moduli o dei thread può risultare nella corruzione delle risorse usate dal resto dell'applicazione, ossia nella propagazione del bug. L'aggiornamento e il bug-fixing inoltre deve essere sviluppato, testato e rilasciato per tutto l'applicativo, complicando tutte le fasi e introducendo nuovi punti rottura. Alla stessa maniera l'aggiunta di nuove

funzionalità comporta una revisione di tutta l'applicazione e la sua sostituzione, con tempi e costi non facilmente determinabili a priori.

In un'applicazione basata su microservizi, invece, le varie funzionalità sono separate ed assegnate ad applicazioni più piccole e dedicate a fornire solo quella funzionalità specifica. Questo porta diversi vantaggi:

- definiti i compiti e le interfacce di comunicazione, i vari microservizi possono essere sviluppati contemporaneamente e indipendentemente, rilasciati in momenti differenti e mantenuti separatamente;
- la riduzione delle dimensioni del singolo micro-servizio permette una più agevole revisione del codice diminuendo le possibilità di introduzione di bug e facilitando la loro identificazione;
- l'applicazione complessiva risulta più resiliente, riducendo la possibilità di single point-of-failure e la propagazione di guasti o interruzioni da un modulo agli altri;
- aggiornare o aggiungere nuovi moduli non richiede la sostituzione o lo spegnimento degli altri.

4.3 GLI APPLICATION CONTAINER

L'architettura software dell'Edge Gateway prevede la distribuzione dei microservizi attraverso application '*container*'. I container possono essere considerati dei pacchetti software contenenti l'applicazione principale che devono eseguire e tutte e sole le librerie e i file richiesta da questa. A differenza di una macchina virtuale, alla quale possono essere paragonati, non richiedono un sistema operativo completo e un kernel proprio in esecuzione, ma utilizzano le funzionalità del kernel in esecuzione nel sistema ospitante. A differenza di una macchina virtuale inoltre condividono le stesse risorse messe a disposizione dal sistema ospitante quali spazio disco, memoria e rete, evitando limitazioni e frammentazioni inutili e mantenendo al contempo la separazione dei processi in esecuzione e l'accesso esclusivo a alle risorse utilizzate.

I vantaggi di usare la containerizzazione per l'esecuzione dei microservizi diventa evidente però considerando l'aspetto della distribuzione. Applicazioni differenti possono richiedere, per la stessa libreria, versioni differenti. In un ambiente condiviso quale ad esempio il sistema operativo ospitante, ciò potrebbe portare ad una rottura della catena delle dipendenze: l'adozione di una versione impedisce l'esecuzione delle altre. Tale problema, seppur non presente inizialmente, può presentarsi successivamente nella vita di un prodotto nel momento in cui è necessario un aggiornamento: l'aggiornamento di una libreria richiesta da un micro-servizio può danneggiare un altro micro-servizio che non necessitava. Questo problema viene evitato con la distribuzione di una application assieme alle proprie copie di librerie in un ambiente dedicato e isolato. Altro problema del quale la containerizzazione rappresenta una soluzione è la riproducibilità. Il container è distribuito con l'applicazione, file e librerie con i quali è stato sviluppato e testato. Se l'applicazione funziona nel container quando viene rilasciata, dato che l'ambiente operativo ospitante non influenza il funzionamento del container, questa funzionerà una volta distribuita.

La tecnologia utilizzata per la containerizzazione dei microservizi nell'Edge Gateway è la piattaforma Docker. Gli strumenti messi a disposizione da Docker permettono la creazione, l'aggiornamento, il *versioning* e l'esecuzione di container per microservizi. In particolare, uno strumento, *docker-compose*, permette l'avvio e la composizione di diversi microservizi configurando aspetti come i collegamenti, lo storage e l'accesso a specifiche periferiche hardware del sottostante sistema ospitante, attraverso un unico file di configurazione.

4.4 COMUNICAZIONI INTERNE ED ESTERNE

I microservizi, contenuti nei container Docker e in esecuzione, necessitano di comunicare tra di loro per poter svolgere le funzioni assegnate all'Edge Gateway. Per evitare che la comunicazione interna possa rappresentare un vincolo per la flessibilità e l'affidabilità del sistema con latenze elevate, carico, timeout e propagazione di errori in presenza di problemi ad un solo dei componenti, si è adottato il paradigma di comunicazione asincrono Publish/Subscribe con boker. Nel Publish/Subscribe il mittente di un messaggio, il *publisher* invia il proprio messaggio ad un tramite centrale

detto *broker* senza necessità di specificare il destinatario ma solo il *topic*, una sorta di mailbox. I destinatari del messaggio, o meglio i consumatori, detti *subscriber* a loro volta si sottoscrivono presso il broker ad uno specifico topic, una sorta di abbonamento, per ricevere i messaggi inviati al relativo topic. Così come il publisher, anche il dispatcher non conosce il mittente del messaggio. Più publisher possono pubblicare sullo stesso topic, e più subscriber sottoscriversi allo stesso topic. O alternativamente, un subscriber può ricevere messaggi da più publisher e uno stesso messaggio può essere recapitato a più subscriber.

Esistono diversi protocolli e framework per che implementano il paradigma Publish/Subscribe con broker. Per l'Edge Gateway si è adottato il protocollo standard MQTT:

oltre ad essere uno dei protocolli maggiormente usati nell'Internet of Things, è estremamente leggero, flessibile e ha implementazioni in diversi linguaggi di programmazione e per diverse architetture. In particolare la sua flessibilità risiede nel fatto, tra le altre cose, di essere agnostico per quanto riguarda il contenuto del messaggio, occupandosi solo della sua trasmissione e ricezione.

Internamente quindi i microservizi produttori di dati quali gli handler dei sensori e delle stazioni pubblicano i propri messaggi su topic distinti in base al dominio di appartenenza, meteo, ambientale, energetico. I microservizi consumatori di tali messaggi, come il dispatcher, invece sottoscrivono su quei topic in attesa di messaggi. La comunicazione asincrona evita quindi ai microservizi di tenere costantemente un canale aperto tra di loro in attesa di dati o di conferme di ricezione. Inoltre, l'aggiunta di un nuovo publisher o di un nuovo subscriber è del tutto trasparente al sistema, in quanto non è necessario creare nuovi instradamenti o assegnare indirizzi o porte.

Per quanto riguarda la trasmissione verso l'esterno, vale lo stesso discorso. L'adozione del protocollo MQTT per le comunicazioni dall'Edge Gateway verso il sistema di raccolta centrale, oltre ai già citati vantaggi, offre la possibilità di autenticare e crittografare i messaggi attraverso SSL/TLS, una migliore efficienza della rete in ingresso al cloud e la scalabilità del servizio.

4.5 I MICROSERVIZI DELL'EDGE GATEWAY

L'Edge Gateway è costituito da tre categorie di microservizi:

- i '*sensor handler*' e '*station handler*' o semplicemente gli '*handler*';
- il '*dispatcher*';
- e gli '*ancillary service*'.

4.5.1 GLI HANDLER

Gli handler sono i microservizi incaricati di gestire i sensori collegati fisicamente all'Edge Gateway e i sensori singoli remoti ('*sensor handler*') o le comunicazioni con le stazioni di misura remote costituite da più sensori ('*station handler*'). In particolare i loro compiti sono, nel caso di un sensor handler

- configurare il canale o bus di comunicazione e inizializzare il sensore;
- interrogare periodicamente il sensore per avere le misure delle grandezze di interesse;
- inviare al broker locale il pacchetto di dati acquisiti.

Per quanto riguarda lo station handler (meteo, energia etc…), i compiti sono:

- instaurare il collegamento con la stazione remota o aprire un socket di rete di ascolto;
- ricevere i dati inviati dalle stazioni;
- inserire i dati nel database locale;
- convertire e inviare al broker locale il pacchetto di dati acquisiti.

Gli handler sono scritti in Python versione 3 e hanno una struttura del codice pressoché simile:

- possono essere configurati sia tramite file di configurazione che tramite opzioni a riga di comando;
- usano lo stesso tipo di file di configurazioni con campi comuni e campi specifici;
- tutti gli handler possono usare un unico file diviso in sezioni, ognuna col nome dell'handler che configura.

Meteo Handler

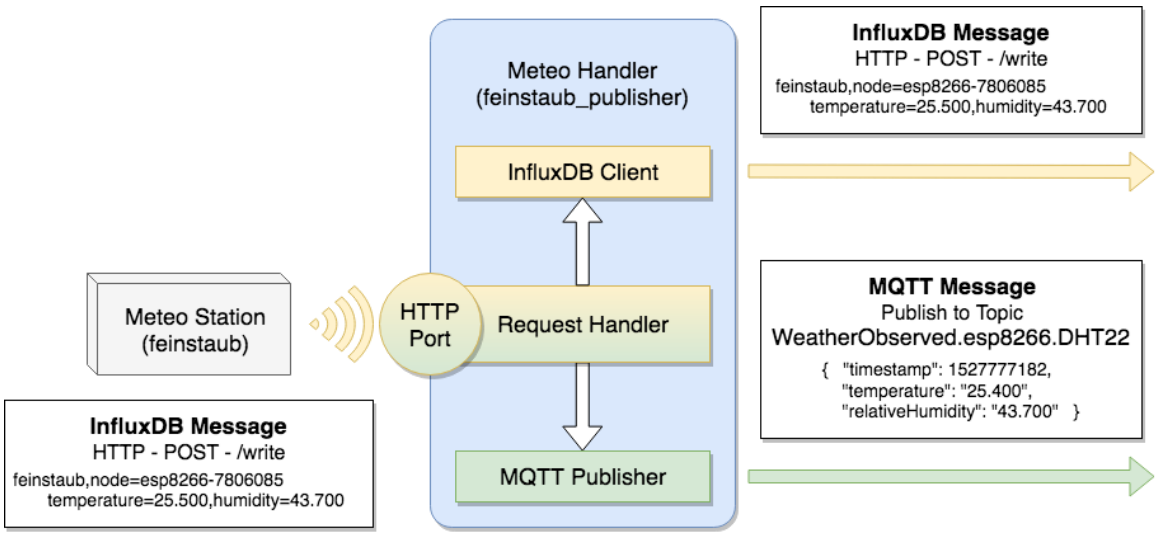
L'handler della stazione meteo remota prende il nome di Feinstaub Publisher dal nome della stazione meteo e qualità dell'aria adottata. Dato che tale stazione invia i propri dati tramite WiFi in formato e con protocollo atto all'inserimento in un database InfluxDB, il relativo handler simula una server InfluxDB in ascolto e intercetta i dati in arrivo. Questi dati sono quindi inseriti nel database locale InfluxDB per una successiva consultazione da parte degli altri microservizi e allo stesso tempo convertiti in un messaggio standard che viene inviato al broker MQTT locale per la gestione di eventi o l'inoltro verso l'esterno.

Energy Handler

L'handler della stazione remota di misura dell'energia prende il nome di Iotawatt Publisher dal nome del tipo di stazione adottata. Dato che anche tale stazione invia i propri dati tramite WiFi per l'inserimento in un database InfluxDB, il relativo handler è molto simile all'handler meteo di cui ne mutua gran parte del codice. Simula anch'essa un server InfluxDB in ascolto, ne riceve i dati, li inserisce nel database locale InfluxDB e li convertite in un messaggio standard che viene inviato al broker MQTT locale.

Internal Sensors Handler

L'Edge Gateway può avere dei sensori direttamente collegati ai propri connettori. In base al tipo di interfaccia o bus alla quale sono collegati, un micro-servizio diverso si occupa di instaurare il collegamento, configurare i sensori presenti sul quella interfaccia e interrogarli periodicamente. Per quanto riguarda il sensore usato in questo reference design, il sensore di temperatura e umidità HTU21D, l'handler prende il nome di HTU21D Publisher. Questo handler apre l'interfaccia sul bus I2C sul quale opera il sensore e ad intervallo di tempo prestabiliti nel file di configurazione o a riga di comando interroga il sensore e invia al broker MQTT locale il messaggio coi parametri rilevati, in questo caso temperatura, umidità e il punto di rugiada calcolato partendo dai primi due.



Esempio di flusso dati all'interno di un handler

4.5.2 IL DISPATCHER

Il *dispatcher* è un micro-servizio in ascolto sul broker locale, o *subscriber MQTT*, che riceve i messaggi inviati dagli handler. Il dispatcher ha il compito principale di inoltrare i dati acquisiti verso il sistema remoto di raccolta ed elaborazione. Esso risponde a due necessità principali:

- i sensori e le stazioni possono generare localmente quantità di dati e a frequenza tale da non poter essere inviati direttamente e in tempo reale al cloud remoto, pena la congestione e il sovraccarico sia della rete esterna, sia dei sistemi remoti; occorre quindi disciplinare l'invio in base alla capacità della rete e alla quantità di dati da inviare;
- per garantire la sicurezza e la confidenzialità della trasmissione al sistema remoto di raccolta attraverso Internet dei dati, è necessario autenticare e cifrare la comunicazione attraverso credenziali di accesso quali nome utente e password e certificati digitali; per maggior sicurezza, è preferibile che solo un micro-servizio abbia accesso e possa usare queste credenziali per presentarsi al broker remoto.

Il dispatcher svolge quindi il compito di:

- disaccoppiare la produzione dei dati dal loro invio al fine di poterli conservare temporaneamente, comprimerli e inviarli ad intervalli configurabili in base alle capacità della rete;
- presentare al cloud remoto le credenziali per l'accesso e l'invio;
- gestire localmente eventuali interruzioni delle comunicazioni o deterioramenti delle stesse.

4.5.3 I SERVIZI AUSILIARI

I servizi ausiliari sono microservizi che implementano delle funzionalità condivise cioè non proprie degli altri microservizi ma da questi utilizzate. Questi servizi ausiliari sono:

- il broker MQTT locale;
- il database locale;
- la dashboard locale.

Il broker MQTT locale

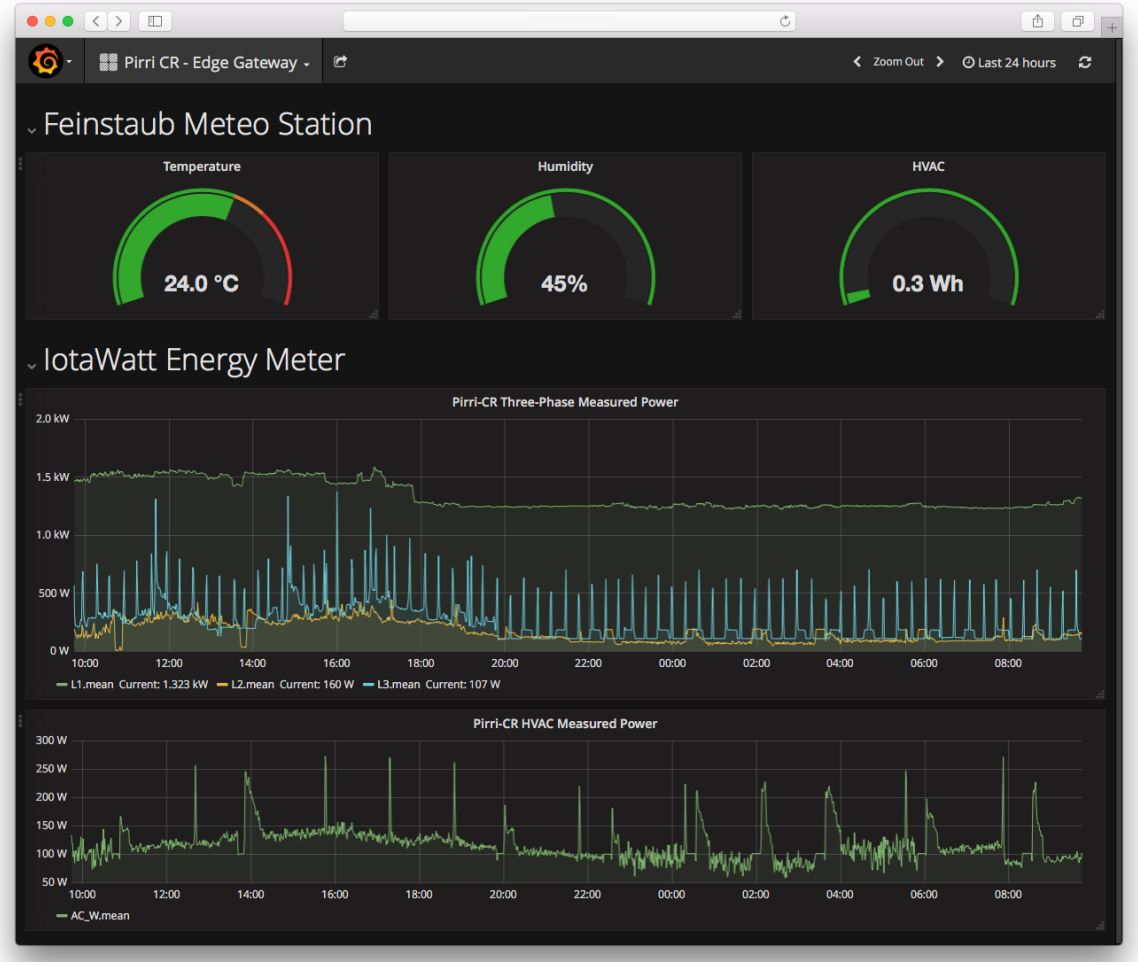
Il broker MQTT locale implementa il bus di comunicazione asincrono per i vari microservizi. In esso i *producer* di dati, gli handler, pubblicano i propri dati su appositi topic, ad esso i *consumer* di dati, il dispatcher ed altri eventuali microservizi di gestione di eventi, sottoscrivono la notifica e l'inoltro dei dati pubblicati. Il software usato per fornire il servizio di broker MQTT si chiama '*mosquitto*'. E' stato scelto in quanto è tra i più leggeri in termini di risorse richieste, implementa tutte le specifiche del protocolle MQTT, sebbene il suo uso interno non richieda particolari necessità, e si è dimostrato il più adatto a venire utilizzato in un ambiente con risorse limitate.

Il Database Locale

Il database locale ha il compito di immagazzinare i dati provenienti da sensori e stazioni di misura remote per la successiva visualizzazione da parte dell'utente, dell'interfaccia grafica locale, o da ulteriori microservizi di pre-processing. Il database utilizzato è InfluxDB ed è stato scelto in quanto database di destinazione generalmente utilizzato dalle stazioni di misurazione remote meteo e energetica.

La Dashboard Locale

La dashboard locale è il software che permette all'utente di visualizzare i dati raccolti e immagazzinati dal suo Edge Gateway. Il software utilizzato in questo caso è Grafana, un tool di visualizzazione WEB estremamente diffuso ed utilizzato per la visualizzazione e la creazione di grafici e cruscotti basati su serie storiche di dati. Grafana permette infatti all'utente, attraverso l'accesso da browser web, non solo di visualizzare i propri dati, ma anche di creare da se grafici e indicatori basati su l'elaborazione di questi.



Esempio di Dashboard Grafana su Edge Gateway

4.5.4 I MODELLI DI DATI

Il design dell'Edge Gateway prevede l'utilizzo di sensori e stazioni di misura di produttori e modelli diversi. Questi possono differire, dal punto di vista della trasmissione dei dati, sia nel protocollo usato, che nel formato dei dati stessi. Affinché dati dello stesso tipo ma prodotti da oggetti diversi e in formato diverso possano essere raccolti ed utilizzati dal progetto TDM, è necessaria una conversione in un formato comune e predefinito. In quanto nel progetto TDM si è scelto di aderire alle linee guida della iniziativa OASC e della sua piattaforma tecnologica di riferimento, l'ecosistema FIWARE, dove possibile e dove già definiti, si è scelto di usare i Modelli di Dato Armonizzato (Harmonized Data Models) di FIWARE. Per i domini dei sensori per i quali non è presente un Data Model Armonizzato FIWARE o non è stato possibile identificarne uno adeguato tra quelli presenti, si è deciso di definirne di nuovi sul modello di quelli esistenti.

Il messaggio di tipo 'Meteo' - WeatherObserved

Per le stazioni meteo si è scelto di usare il modello di dato armonizzato 'WeatherObserved' (https://fiware-datamodels.readthedocs.io/en/latest/Weather/WeatherObserved/doc/spec/index.html). Tuttavia, dato che parte dei sensori presenti nella stazione meteo inizialmente utilizzata riguardano anche misure della qualità dell'aria, e che il relativo modello non appariva sufficiente, si è esteso il 'WeatherObserved' aggiungendo i campi necessari.

Il modello dati 'WeatherObserved' è messaggio di tipo 'JSON' con campi obbligatori e campi opzionali. Il messaggio contenente i campi obbligatori ma in difetto di alcuni campi opzionali è comunque ritenuto valido. E' questo il caso di stazioni meteo che non posseggono tutti sensori richiesti. Non tutti i campi previsti nel modello sono però presenti nel messaggio. Alcuni di questi infatti vengono popolati e utilizzati dai componenti della piattaforma FIWARE nel cloud TDM durante l'elaborazione.

I campi obbligatori sono:

- **id**: identificativo univoco della stazione;
- **type**: tipo del messaggio, deve essere '**WeatherObserved**';
- **location**: posizione della stazione meteo (GeoJSON) obbligatorio se address non è presente;
- **address**: indirizzo civico della stazione;
- **dateObserved**: data e ore dell'osservazione (ISO8601 UTC).

I campi opzionali usati sono:

- **temperature**: temperatura dell'aria (numerico, gradi Celsius);
- **relativeHumidity**: umidità relativa dell'aria (numerico, percentuale);
- **precipitation**: quantità di pioggia (numerico, mm);
- **windDirection**: direzione del vento (numerico, gradi decimali Nord);
- **windSpeed**: velocità del vento (numeric m/s);
- **atmosphericPressure**: pressione atmosferica (numerico, hPa);
- **illuminance**: illuminazione (numerico, lux o lumen/mq).

Campi opzionali aggiunti:

- **timestamp**: timestamp della creazione del messaggio (numerico, UNIX Epoch);
- **infraredLight**: luce infrarossa (numerico, misura relativa al sensore);
- **CO**: quantità di CO (numerico, misura relativa al sensore);
- **NO**: quantità di NO (numerico, misura relativa al sensore);
- **NO2**: quantità di NO2 (numerico, misura relativa al sensore);
- **NOx**: quantità di NOx (numerico, misura relativa al sensore);
- **SO2**: quantità di SO2 (numerico, misura relativa al sensore);
- **PM10**: quantità di polveri PM 10 (numerico, misura relativa al sensore);
- **PM2.5**: quantità di polveri PM 2.5 (numerico, misura relativa al sensore).

Dato che il campo **dateObserved** può essere modificato dai componenti della piattaforma FIWARE, questo non viene inviato dagli EDGE ma è stato aggiunto il campo **timestamp** per comunicare il timestamp del rilevamento da parte dei sensori.

```
{
            "id":                                  "Cagliari-Pirri-ExDistilleria",
            "type":                                              "WeatherObserved",
            "dateObserved":                          "2016-11-30T07:00:00.00Z",
            "location":
```

```
            {
                "type":                                                 "Point",
                "coordinates":
                [            -4.754444444,            41.640833333            ]
            },
            "temperature":                                                 3.3,
            "relativeHumidity":                                             1,
            "atmosphericPressure":                                       938.9,
            "windDirection":                                              -45,
            "windSpeed":                                                    2,
            "precipitation":                                                0,
            "illuminance":                                               1000,
            "infraredLight":                                              850,
            "CO":                                                         500,
            "NO":                                                          45,
            "NO2":                                                         69,
            "NOx":                                                        139,
            "SO2":                                                         11,
            "PM10":                                                        11,
            "PM2.5":                                                       11,
}
```


Esempio del messaggio FIWARE 'WeatherObserved' usato.

Il messaggio di tipo 'Energia' - EnergyMonitor

Per le stazioni di monitoraggio energetico non è stato identificato un data model adeguato e si è provveduto a crearne uno ad-hoc che possa essere ampliato e sottoposto alla comunità FIWARE per un eventuale discussione e adozione. Come per il messaggio 'WeatherObserved', anche il messaggio 'EnergyMonitor' è in formato 'JSON' con campi obbligatori e campi opzionali, e un messaggio 'EnergyMonitor' contente campi obbligatori ma in difetto di alcuni campi opzionali è comunque ritenuto valido.

I campi obbligatori sono:

- **id**: identificativo univoco della stazione;
- **type**: tipo del messaggio, deve essere '**EnergyMonitor**';
- **location**: posizione della stazione meteo (GeoJSON) obbligatorio se address non è presente;
- **address**: indirizzo civico della stazione;
- **dateObserved**: data e ore dell'osservazione (ISO8601 UTC).

I campi opzionali usati sono:

- **timestamp**: timestamp della creazione del messaggio (numerico, UNIX Epoch);
- **voltage**: tensione della linea elettrica (numerico, Volt);
- **current**: corrente che attraversa la linea elettrica (numerico, Ampere);
- **apparentPower**: potenza apparente assorbita dal carico (numerico, Volt-Ampere);
- **realPower**: potenza reale assorbita dal carico (numerico, Watt);
- **powerFactor**: fattore di potenza (numerico, tra 0 e 1);
- **consumedEnergy**: energia consumata (numerico, Wh);
- **frequency**: frequenza della linea eletrica (numerico, Hz).

Dato che il campo **dateObserved** può essere modificato dai componenti della piattaforma FIWARE, questo non viene inviato dagli EDGE ma è stato aggiunto il campo **timestamp** per comunicare il timestamp del rilevamento da parte dei sensori.


```
{
            "id":                                  "Cagliari-Pirri-ExDistilleria",
            "type":                                                "EnergyMonitor",
            "dateObserved":                             "2016-11-30T07:00:00.00Z",
            "location":
            {
                "type":                                                    "Point",
                "coordinates":
                [-4.754444444, 41.640833333]

            },
            "voltage":                                                      235.90,
            "current":                                                        3.74,
            "apparentPower":                                                882.9,
            "realPower":                                                    250.8,
            "powerFactor":                                                    0.66
}
```


Esempio del messaggio NGSIv1 'EnergyMonitor' usato.

4.6 I TOPIC

Per poter inoltrare correttamente e secondo eventuali policy di trasmissione i messaggi, questi devono essere pubblicati internamente in specifici topic MQTT. Questi in generale hanno la forma:

TipoDelMessaggio/IdentificativoStazione.ModelloDelSensore

4.6.1 I TOPIC PER 'METEO'

I topic per i messaggi meteo iniziano con '*WeatherObserved*'. Ad esempio, per una stazione di tipo *feinstaub* i messaggi relativi a temperatura e umidità misurati dal sensore *DHT22* saranno pubblicati sul topic:

WeatherObserved/esp8266-XXXX.DHT22

4.6.2 I TOPIC PER 'ENERGIA'

I topic per i messaggi di tipo energetico iniziano con '*EnergyMonitor*'. Ad esempio, per una stazione di tipo *IotaWatt* i messaggi relativi a corrente, voltaggio, e altri parametri misurati sulla linea di un condizionatore, *HVAC*, saranno pubblicati sul topic:

EnergyMonitor/IOTAWATT.HVAC

5. LA PIATTAFORMA HARDWARE DELL'EDGE GATEWAY

5.1 SOLUZIONI HARDWARE CONSIDERATE

Tre Single Board Computer sono state prese in considerazione come piattaforme hardware embedded per il prototipo dell'Edge Gateway: la scheda BeagleBone Black e le schede Raspberry PI versione 2-B e 3-B. Queste tre rispondono ai requisiti iniziali di costo ridotto e reperibilità sul mercato, essendo acquistabili facilmente su e-commerce con costi tra i 30 e gli 80 euro, e possiedono un'ampia comunità di sviluppatori e utenti. Innumerevoli forum, portali dedicati e blog inoltre forniscono documentazione, assistenza ed esempi di applicazioni che ne fanno piattaforme ideali per scopi didattici.

5.1.1 BEAGLEBONE BLACK

La BeagleBone Black è un single-board computer prodotta da Texas Instruments assieme a Digi-Key and Newark Element14. Utilizza una CPU ARM Cortex-A8 a 32 bit, frequenza massima di 1GHz con 512 MB di RAM DDR3. Il processore ha un singolo core ma dispone di due unità embedded *Real-Time* programmabili. Lo storage è rappresentato da una memoria da 4GB onboard e un lettore per microSD card. Possiede un ampio set di periferiche a basso livello come UART, CAN, I2C/SPI, ADC, USB.

La connettività di rete è fornita da una scheda di rete ethernet integrata a 100 Mbit/s. Non possiede connettività wireless, che può tuttavia essere fornita attraverso un dispositivo esterno USB. L'assorbimento di corrente stimato è di 210-460 mA con alimentazione a 5V.

5.1.2 RASPBERRY PI 2

Le Raspberry Pi sono una serie di SBC sviluppate nel Regno Unito dalla Raspberry Pi Foundation al fine di fornire una piattaforma per l'insegnamento dei principi dell'informatica e della programmazione nelle scuole e nei paesi in via di sviluppo. La Raspberry PI 2 utilizza un System-on-Chip Broadcom basato sul processore ARM Cortex-A7 a 32 bit con frequenza massima di 900 MHz con 1 GB di RAM DDR2. Il processore della Raspberry PI 2 possiede 4 core. Lo storage consiste in un lettore per card microSD e non ha dispositivi memorizzazione on-board. La dotazione di periferiche a basso livello consiste in GPIO, UART, I2C/SPI, USB e interfacce per Camera e Display, mentre mancano convertitori Analogico/Digitali (ADC) e il controller CAN Bus presente invece nella BeagleBone Black.

La connettività di rete è fornita da una scheda di rete ethernet integrata a 100 Mbit/s. Neanche la Raspberry Pi 2 possiede connettività wireless, che può tuttavia essere fornita attraverso un dispositivo esterno USB. L'assorbimento di corrente stimato è di 800 mA con alimentazione a 5V.

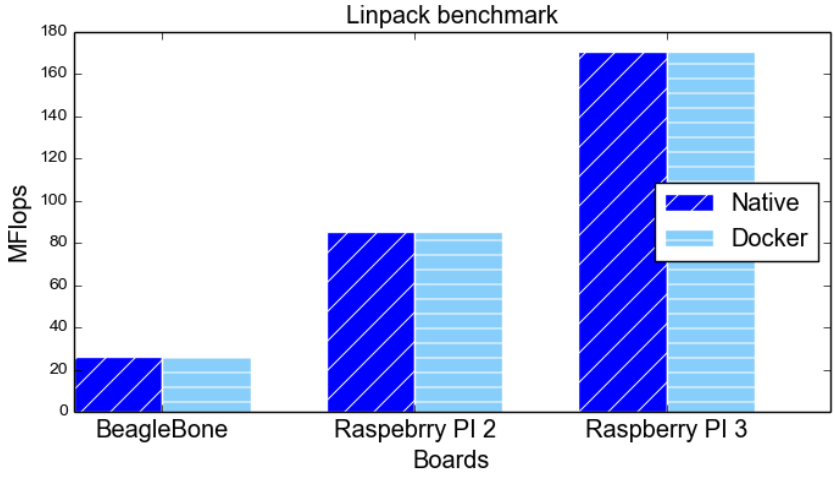
5.1.3 RASPBERRY PI 3

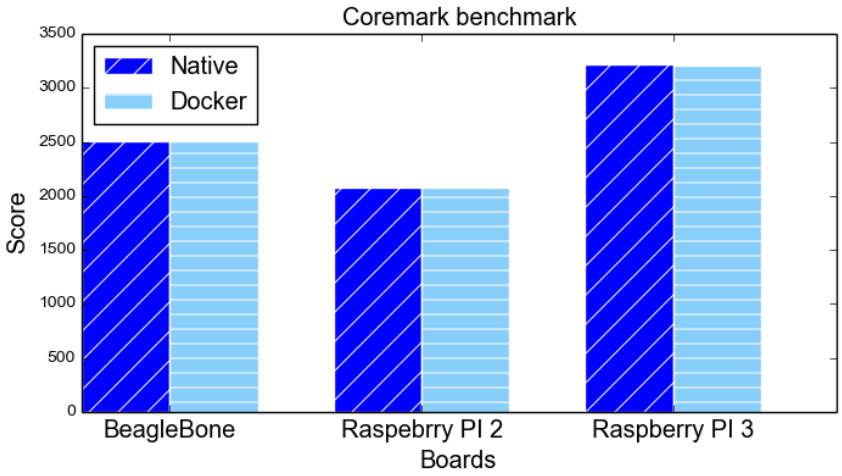
La Raspberry Pi 3 è una evoluzione della Raspberry PI 2. Le novità principali rispetto a quest'ultima sono il processore ARM Cortex- A53 a 64, sempre quad-core e la presenza di un chip WiFi 802.11n e Bluetooth BLE integrato. La frequenza massima della CPU passa a 1.2GHz, mentre la RAM resta ad 1 GB of DDR2. Anche il set di periferiche rimane lo stesso della Raspberry PI 2. Anche nella RPi 3 lo storage consiste in un lettore per card microSD e non ha dispositivi memorizzazione on-board. E' presente una scheda di rete integrata a 100 Mbit/s NIC. L'assorbimento di corrente stimato va da 300mA, 1.5W a riposo, fino a 1.34 A, 6.7 W sotto carico, con tensione di alimentazione a 5V. I produttori suggeriscono comunque di utilizzare alimentatori in grado di erogare almeno 2.5 A, onde evitare la corruzione del filesystem e comportamenti anomali.

5.2 BENCHMARKS E SCELTA DELLA PIATTAFORMA HARDWARE

Al fine di valutare le prestazioni delle tre schede dal punto di vista computazionale e per verificare se la dotazione di risorse è adeguata per eseguire l'architettura software prevista per l'Edge Gateway sono stati condotti diversi benchmark, riassunti di seguito.

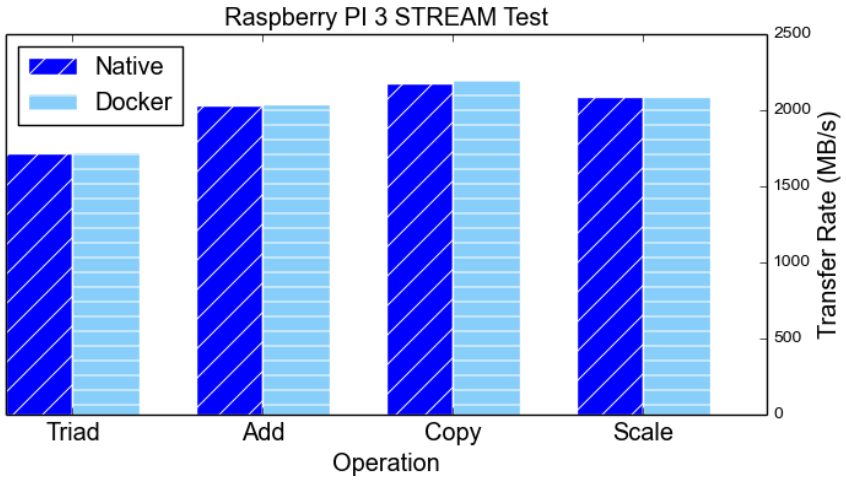
La capacità computazionale e soprattutto la compatibilità con il sistema di container Docker è stata valutata attraverso due benchmark consolidati: Linpack e Coremark. Il Linpack misura la potenza di calcolo rispetto ad operazioni floating-point mentre il Coremark si occupa testare algoritmi tipici nei sistemi embedded come, ad esempio, i CRC. Come mostrato dai 2 grafici seguenti, i test sono stati fatti sia in ambiente nativo (Arch Linux) che all'interno di un container Docker.



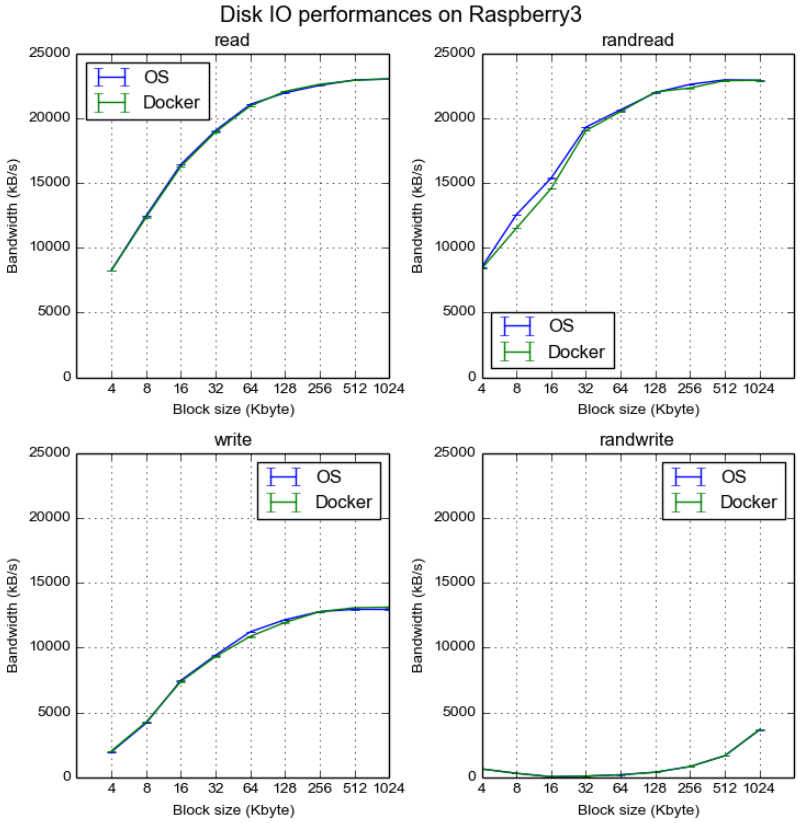


Da entrambi i benchmark risulta evidente che l'esecuzione del codice all'interno di un container Docker non è soggetta ad un deterioramento delle prestazioni. Dato che, come si può osservare, la Raspberry PI 3 sia risultata la piattaforma più potente per i test successivi verranno riportati solo i risultati di quelli eseguiti su questa piattaforma.

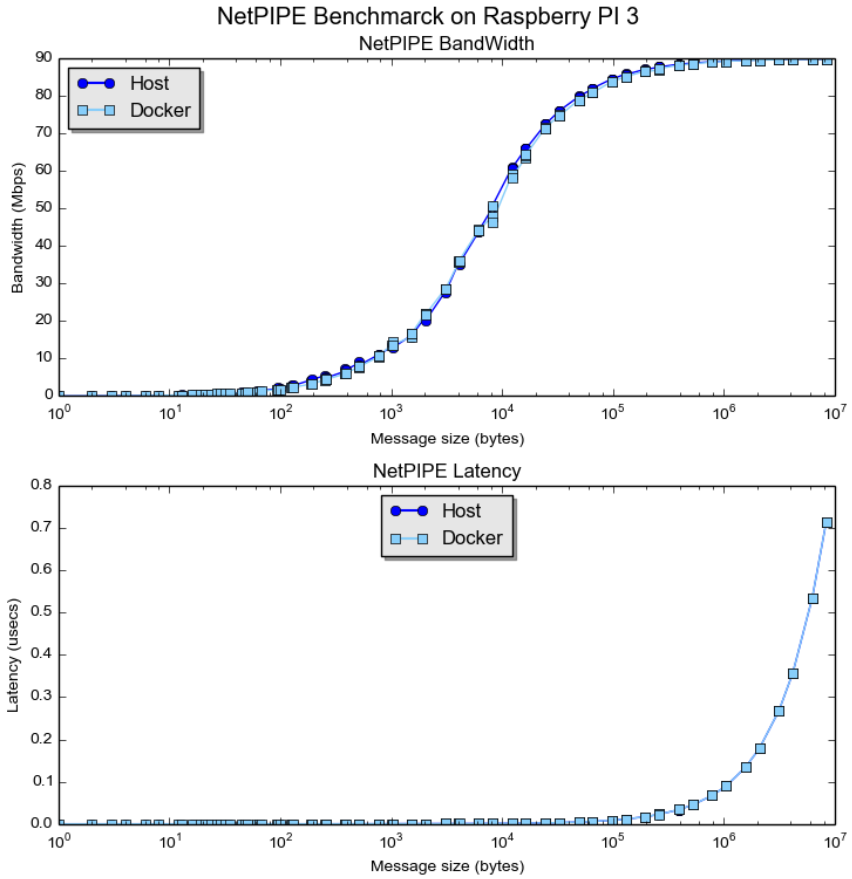
Le performance della memoria sono state misurate col benchmark STREAM. STREAM appartiene alla suite HPCC ed è utilizzato per valutare l'ampiezza di banda della memoria RAM rispetto a differenti tipi di accesso (ad esempio, Triad si riferisce ad un accesso del tipo: a[i] = b[i] + c[i]*scale e rappresenta il caso più oneroso). Dal diagramma, risulta una sostanziale equivalenza tra le operazioni compiute in ambiente nativo e all'interno di un container Docker.



Il test successivo è riferito al trasferimento di dati da e verso la memoria di massa, in questo caso una SD card di classe 10. Il test è stato effettuato utilizzando il benchmark sintetico FIO, un generatore di workload molto flessibile che permette di simulare vari scenari nel trasferimento dei dati. I grafici sono relativi ad operazioni di lettura e scrittura sia sequenziale che casuale. L'asse orizzontale indica il blocksize usato per le operazioni, mentra l'asse verticale l'ampiezza di banda espressa in kB/s. In questo test si ha solo una leggera differenza nell'uso di Docker per letture casuali con blocksize da 4KB fino a 64KB.



L'ultimo test si riferisce alle prestazioni di rete. Nei diagrammi di seguito è visualizzata l'ampiezza di banda e la latenza del benchmark NetPIPE. NetPIPE esegue le misure in modo indipendente dal protocollo di rete facendo una semplice comunicazione andata-ritorno tra due processi con messaggi di dimensioni crescenti. Dai risultati si ha la conferma dell'impatto sostanzialmente nullo di Docker sulle prestazioni della piattaforma scelta.



In conclusione, delle tre schede prese in esame, quella dotata di maggior capacità computazionale, anche alla luce dei test, è risultata la Raspberry Pi 3. Questa ha anche la connettività wireless integrata e non necessita quindi dell'acquisto di un dongle USB aggiuntivo. Manca però di convertitori A/D e di storage on-board, presenti nella BeagleBone Black. Le performance migliori costano però in termini di alimentazione, richiedendo la RPi 3 almeno 2.5 A di corrente.

L'utilizzo di container Docker non sembra aggiungere un sovraccarico significativo al sistema operativo ospitante, soprattutto a fronte dei vantaggi che offre. O, alternativamente, un processo in esecuzione all'interno di un container Docker su sistemi embedded ARM, non percepisce la differenza

5.3 SENSORISTICA

Per poter avviare lo sviluppo dell'Edge Gateway, così come per le altre attività verticali del progetto, sono stati identificati sensori e sistemi di misura iniziali da integrare con l'Edge Gateway.

5.3.1 STAZIONE DI MISURA METEO/AMBIENTALE

Tra i vari progetti open source e open hardware, sono state individuate tre differenti piattaforme per la sensoristica dei parametri ambientali e l'integrazione con l'Edge Gateway:

- lo Smart Citizen Kit, SCK (https://smartcitizen.me/), device portatile per la misura di temperatura, umidità, luce, suono, CO, NO2;
- l'Arduino Weather Station Project, AWSP (http://cactus.io/projects/weather/arduino-weather-station), piattaforma di misura fissa di tipo "tradizionale" con sensori per pressione atmosferica, temperatura, umidità, velocità e direzione de vento e precipitazione;
- lo Stuttgart Fine Dust Sensor, SFDS (https://luftdaten.info/en/construction-manual/) per la misurazione di umidità relativa, temperatura, e particolato PM10 e PM2.5.

Per lo sviluppo e il test di laboratorio dell'Edge Gateway si è scelto di usare la SFDS in quanto la sua architettura e software ha permesso una rapida integrazione con il prototipo dell'Edge Gateway. Inoltre, data la versatilità della scheda, parte dei sensori della AWSP sono stati integrati nella SFDS, in modo da avere una piattaforma unica, versatile ed espandibile.

La connettività della SFDS è rappresentata dal chip WiFi integrato, il quale può agire sia come Access Point per la prima configurazione, che come WiFi client per l'invio dei dati verso server di raccolta e visualizzazione remota. La SFDS supporta diversi server remoti tra cui il database InfluxDB e sistemi basati su API Rest. Non disponendo di uno storage locale, in assenza di connettività i dati non possono essere salvati e ri-trasmessi. In rete locale questo può essere superato dall''integrazione con l'Edge Gateway al quale si connette nativamente.

Grafici dei dati meteo come visualizzati da *Grafana*

5.3.2 ENERGY MONITOR

Come strumento iniziale per il monitoraggio dei consumi energetici si è scelto di utilizzare il sistema IotaWatt. Tale sistema è stato utilizzato anche nello sviluppo e test dell'Edge Gateway.

IotaWatt è un sistema di monitoraggio elettrico accurato, multicanale, open-hardware, open-source, a basso costo e facile da usare. Si basa su piattaforma IoT dotata di connettività WiFi ed impiega per il campionamento di tensione e corrente convertitori Analogici/Digitali a 12 bit capaci di raggiungere elevate frequenze di campionamento. Il dispositivo dispone di un Real Time Clock integrato e di una scheda SD per la memorizzazione delle misure. Supporta la funzione di interrogazione dei dati con il server web integrato e la trasmissione verso cloud.

Le caratteristiche principali:

- 14 canali di acquisizione;
- API REST per l'estrazione dei dati
- supporta sensori di corrente di marca, modello e capacità differenti
- supporta la definizione generica di qualsiasi sensore di corrente
- configurazione e visualizzazione basate su browser LAN locale
- Open Hardware/Software
- cloud supportati influxDB e Emoncms.org

Il software campiona i canali di ingresso alla velocità di 35-40 canali al secondo, registrando la tensione (V), la potenza (Watt) e l'energia (kWh) su scheda SD locale ogni cinque secondi. I dati possono essere visualizzati su rete WiFi o inviati su server remoto Emoncms o database influxDB. In caso di interruzioni del WiFi o del servizio Internet, IotaWatt continuerà a registrare localmente, per poi aggiornare il server quando il sistema WiFi viene ripristinato.

Il sistema IoTaWatt

6. IL PROTOTIPO REALIZZATO

Il prototipo di laboratorio dell'Edge Gateway consiste di tre componenti:

- l'Edge Gateway
- la stazione meteo/ambientale
- la stazione di misura energetica

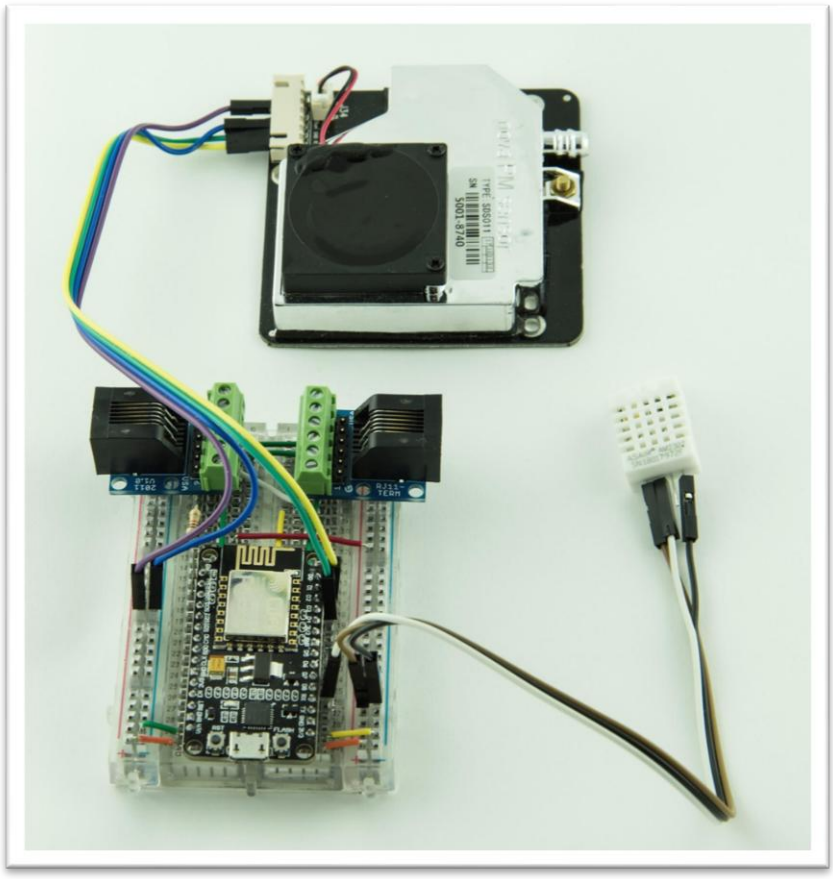
L'Edge Gateway a sua volta è formato da:

- scheda Raspberry Pi 3-B;
- sensore di temperatura/umidità I2C di test collegato direttamente;
- microSD da 16G classe 10 contenente
- Sistema Operativo Arch Linux per Raspberry Pi con le seguenti personalizzazioni per l'Edge Gateway TDM:
    - modalità *Access Point* che crea una WiFi Lan per la prima configurazione e la connessione delle stazioni di misura;
    - applicativi di gestione dei container Docker e dei repository Git preinstallati;
    - script per l'avvio automatico dei container;
- codice demo dei microservizi:
    - dispatcher (EDGE_dispatcher) per l'invio dei dati raccolti verso il cloud;
    - handler meteo (FEINSTAUB_publisher) per l'interfacciamento con le stazioni meteo/qualità dell'aria;
    - handler energia (IOTAWATT_publisher) per l'interfacciamento con le stazioni di misura energetica;
    - handler di test per il sensore cablato HTU21D (HTU21D_publisher) per misure di temperatura e umidità;
    - container per il database locale InfluxDB;
    - container per la visualizzazione locale dei dati Grafana;
    - container del broker locale per i messaggi MQTT degli handler Mosquitto.

Il prototipo è stato integrato con i seguenti sistemi di misura, i quali sono stati interfacciati, valutati e testati. In alcuni casi, come per la stazione meteo/ambientale, è stato modificato parte del firmware per integrare ulteriori funzionalità:

- stazione di monitoraggio energetica IotaWatt, con 5 sonde di corrente e una di tensione collegata a quadro di alimentazione generale e impianto HVAC;
- stazione meteo/ambientale Stuttgart Fine Dust Sensor (codice Feinstaub) con:
    - anemometro e segnavento Argent System;
    - anemometro e segnavento Davis;
    - misuratore direzione vento Argent System;
    - pluviometro a vasca basculante Argent System;
    - anemometro Davis;
    - misuratore direzione vento Davis;
    - pluviometro a infrarosso Hydreon RG11;
    - sensore temperatura e umidità DHT22;
    - sensore di particolato PM10 e PM 2.5 Nova Fitness SDS11.

Edge Gateway con il sensore di prova HTU21D

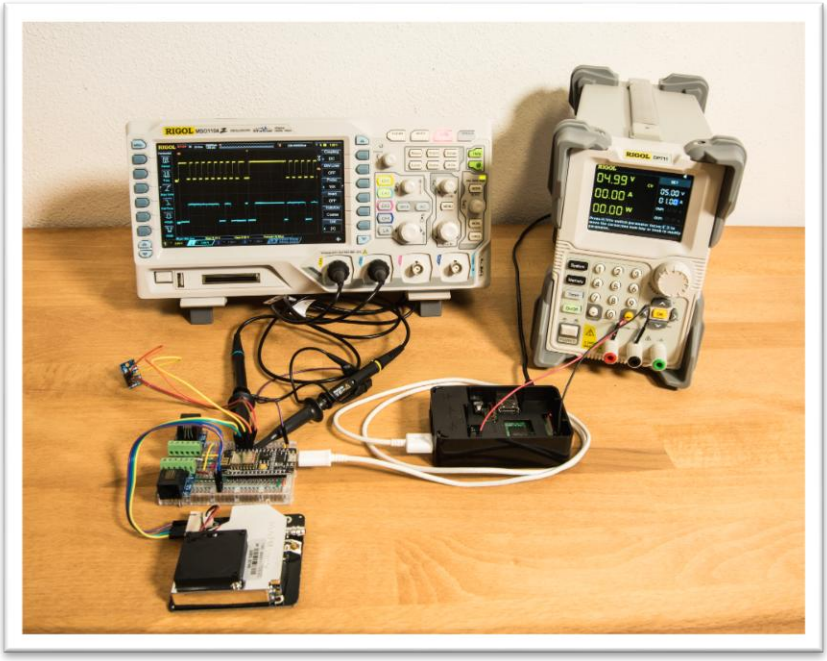


La stazione meteo/ambientale col sensore di particolato e t/h

Sensori vento e pioggia Argent System

Anemomentro e Segnavento Davis

La stazione di monitoraggio elettrico IotaWatt installata



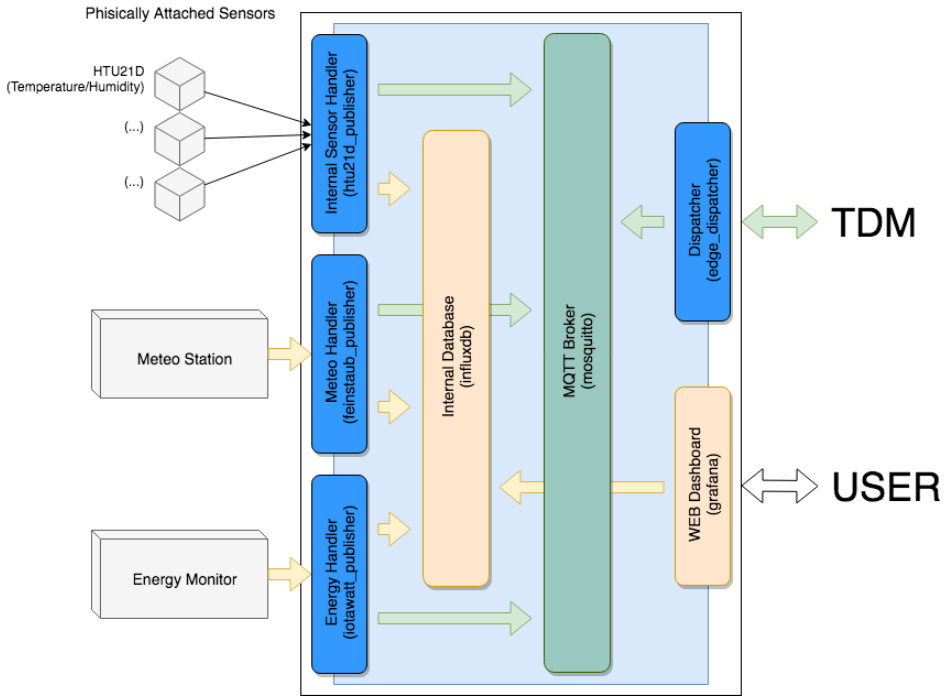
Testing in laboratorio della soluzione standalone Edge Gateway – Stazione Meteo

7. MANUALE DI INSTALLAZIONE E CONFIGURAZIONE

7.1 INTRODUZIONE

Lo scopo di questo documento è di descrivere i passi necessari per l'installazione e la configurazione di un Edge Gateway TDM, al fine di essere utilizzato assieme alle stazioni di misura previste nel documento di Design.

L'Edge Gateway consiste in una scheda Single Board Computer Raspberry Pi 3 con relativa alimentazione e scheda di memoria microSD ed eventuali sensori fisicamente collegati. Nel prototipo di laboratorio si è usato il sensore I2C di temperatura e umidità HTU21D, ma è possibile aggiungere altri dispositivi, dato che le specifiche della scheda e i sorgenti di esempio sono pubblici. La comunicazione con il resto delle stazioni di misura e il cloud avviene per mezzo della rete WiFi domestica.



Scheda Raspberry Pi 3 con microSD visibile sul lato inferiore (foto a sinistra) - Edge Gateway con sensore HTU21D (foto a destra)

7.2 INSTALLAZIONE

Per procedere all'installazione dell'Edge Gateway occorrono:

- un PC Linux con lettore per schede SD/microSD;
- privilegi di '*root*';
- un adattatore SD/microSD;
- accesso ad una rete WiFi;

La versione corrente del'Edge Gateway consiste di:

- una Raspberry Pi 3;
- un alimentatore 220 V - 5 V microUSB in grado di erogare almeno 2,5 A;
- una microSD da almeno 8 GB di capacità, consigliati 16 GB, classe 10.

I passi dell'installazione seguenti permettono di:

1. installare il sistema operativo personalizzato per l'Edge Gateway;
2. effettuare il primo accesso e la configurazione della rete WiFi casalinga;
3. installare i microservizi;
4. configurare il software TDM e avviare i servizi.

Inoltre viene mostrato il collegamento di un sensore di test per temperatura e umidità HTU21D.

7.2.1 INSTALLAZIONE DEL SISTEMA OPERATIVO SULL'EDGE GATEWAY

Nota: Per alimentare la Raspberry Pi 3 è raccomandato un alimentatore in grado di erogare almeno 2,5A: una corrente insufficiente infatti può causare malfunzionamenti e corruzione del filesystem.

Nota: La seguente procedura di installazione è basata in parte sulla procedura di installazione (in lingua inglese) di Arch Linux per architettura ARMv7. La procedura originale è disponibile all'indirizzo: https://archlinuxarm.org/platforms/armv8/broadcom/raspberry-pi-3 .

Le seguenti istruzioni consistono in comandi da digitare in un terminale o una console sul PC Linux. Esse, se impartite da utente diverso da '***root***' richiedono i privilegi di amministratore. Alternativamente si può usare una shell '***sudo***'.

Per avviare una shell sudo da un terminale Linux ed eseguire le successive istruzioni, digitare:

```
sudo -s
```

ed inserire la propria password.

ATTENZIONE: le operazioni che seguono distruggeranno i filesystem presenti nella scheda microSD, cancellandone il contenuto. Prestare la massima attenzione ai nomi dei dispositivi verso i quali i comandi sono indirizzati.

Di seguito sostituire ***sdX*** con il nome della periferica SD come mostrata sul computer. Per essere certi di agire sul dispositivo corretto si può utilizzare il comando *fdisk -l* prima e dopo l'inserimento della scheda SD per identificare il device id del dispositivo inserito.

1. inserire la microSD nel lettore (usando eventualmente l'adattatore);
2. prima di procedere al partizionamento è opportuno verificare che il sistema operativo non abbia montato eventuali partizioni già presenti sulla SD. A tal fine si può procedere consultando l'output del comando *mount.* Se nell'output è presente una o più partizioni con suffisso **sdX** (ad esempio **sdX1)** smontarla mediante il comando

```
umount /dev/sdX{numero_partizione}
```

3. Partizionare la microSD con ***fdisk***:

```
fdisk /dev/sdX
```

4. Al prompt di fdisk, eliminare le vecchie partizioni e crearne una nuova:
    a. Digitare **o** + **ENTER**. In questo modo si eliminano eventuali partizioni sull'unità.
    b. Digitare **p** + **ENTER** per elencare le partizioni. Non ci dovrebbero essere più partizioni.
    c. Digitare **n** + **ENTER**, poi **p** per primario, **1** per la prima partizione sull'unità, premere **ENTER** per accettare il primo settore predefinito, quindi digitare **+100M** per l'ultimo settore. Nel caso dovesse essere richiesto, premere **y** per rimuovere la firma *vfat* o *ext4* dalla prima partizione.
    d. Digitare **t** + **ENTER**, quindi **c** per impostare la prima partizione sul tipo W95 FAT32 (LBA).
    e. Digitare **n** + **ENTER**, poi **p** per primario, **2** per la seconda partizione sull'unità, quindi premere due volte **ENTER** per accettare il primo e l'ultimo settore predefiniti. Nel caso dovesse essere richiesto, premere **y** per rimuovere la firma ext4 dalla prima partizione.
    f. Digitare **w** + **ENTER** per scrivere la tabella delle partizioni e uscire da fdisk.
5. Creare e montare il filesystem FAT:

```
cd /tmp/

mkfs.vfat /dev/sdX1
```

```
mkdir boot
mount /dev/sdX1 boot
```

6. Creare e montare il filesystem ext4:

```
mkfs.ext4 /dev/sdX2
mkdir root
mount /dev/sdX2 root
```

6. Scaricare ed estrarre il filesystem di root (per preservare i permessi dei file, eseguire il comando come utente *root* e non utilizzando il comando *sudo*):

```
wget --content-disposition \

    https://space.crs4.it/s/ywvVDh3tuOBWCZ2/download
tar -xpf TDM-Arm-image-latest.tar.gz -C root
sync
```

Il comando '*tar*' potrebbe mostrare messaggi di errore quali:

- tar: Ignoring unknown extended header keyword 'SCHILY.fflags'
- tar: Ignoring unknown extended header keyword 'LIBARCHIVE.xattr.security.capability'

Su alcuni sistemi sono previsti e possono essere ignorati.

7. Spostare i file di avvio nella prima partizione:

```
mv root/boot/* boot
```

8. Smontare le due partizioni:

```
umount boot root
```

Ora è possibile uscire dalla shell sudo, se in uso, premendo **CTRL+D** o digitando

```
exit
```

7.2.2 PRIMO ACCESSO ALL'EDGE GATEWAY E CONFIGURAZIONE WIFI

Appena avviato l'Edge Gateway si presenta come un Access Point con una sua rete WiFi privata. Attraverso questa è possibile accedere al dispositivo.

1. Inserire la scheda microSD nella Raspberry Pi e applicare l'alimentazione 5V;
2. attendere alcuni secondi che si avvii il sistema operativo;
3. dal PC, cercare una rete WIFI che abbia un nome simile a 'TDM_XXXXXXXX' e collegarsi utilizzando la password '***tdmedgegateway***';

ATTENZIONE: lasciare attiva la rete WiFi locale dell'Edge senza cambiare la password di default potrebbe permettere l'accesso all'Edge a chiunque conosca tale password. Si consiglia di modificarla al primo accesso o disabilitare la rete WiFi locale dell'Edge con i commandi mostrati nella sezione '*Cambio password utente e passphrase WiFi*'.

4. Utilizzando un terminale SSH collegarsi all'indirizzo IP '***192.168.2.1***':
    - Effettuare il login come utente predefinito '*alarm*' con password '*alarm*':

```
ssh alarm@192.168.2.1
```

Appena entrati sull'Edge viene richiesto di cambiare obbligatoriamente la password di accesso:

- digitare la password corrente '*alarm*'
- digitare la nuova password, contenente almeno 8 caratteri
- ri-digitare la nuova password per conferma

Il collegamento SSH a questo punto viene terminato e si può procedere a ricollegarsi usando la nuova password.

Configurazione WiFi domestica sull'Edge Gateway

ATTENZIONE: per sicurezza, l'utente '*root*' è disabilitato, Le operazioni di amministrazione sono possibili solo attraverso l'uso del comando '***sudo***'.

5. Ottenere i privilegi di *amministratore* inserendo la password quando richiesto dopo aver digitato il comando:

```
sudo -s
```

6. Configurare l'Edge Gateway per collegarsi alla rete WIFI casalinga; sostituire <ESSID> con il nome della rete WIFI domestica e dopo aver digitato il seguente comando:

```
wpa_passphrase "<ESSID>" > /etc/wpa_supplicant/wpa_supplicant-wlan0.conf
```

7. il precedente comando non fornisce un *prompt*, cioè non stampa nessun messaggio di testo ma resta in attesa di input da parte dell'utente; inserire la password della rete WiFi domestica e premere **ENTER**;
8. Riavviare l'interfaccia di rete wireless per rendere effettive le modifiche:

```
systemctl restart wpa_supplicant@wlan0
```

9. Se tutto è andato correttamente, l'indirizzo IP assegnato dal router wireless casalingo può essere ricavato col comando:

```
ifconfig wlan0
```

Annotare l'IP in modo da poterlo usare per accedervi successivamente e per la configurazione delle stazioni di rilevamento.

10. Ora è possibile uscire dalla shell sudo, se in uso, premendo **CTRL+D** o digitando

```
exit
```

Cambio password utente e passphrase WiFi

Per modificare successivamente la password dell'utente 'alarm' digitare il comando:

```
passwd
```

e:

- digitare la password corrente
- digitare la nuova password, contenente almeno 8 caratteri
- ri-digitare la nuova password per conferma.

Per modificare la passphrase per la rete wireless creata dall'Edge Gateway, digitare:

```
sudo hostap_chpsk
```

- digitare la nuova passphrase, contenente minimo 8 caratteri e massimo 63
- ri-digitare la nuova passphrase per conferma.

Se non necessaria, è possibile disabilitare la rete wireless creata dall'Edge Gateway coi comandi:

```
sudo systemctl disable start_ap
sudo systemctl stop start_ap
```

7.2.3 INSTALLAZIONE SOFTWARE TDM

Per procedere alle operazioni che seguono occorre per prima cosa eseguire il login ssh all'indirizzo annotato in precedenza usando l'utente alarm. Ottenuto l'accesso passare all'utente root digitando il comando:

```
sudo -s
```

e inserendo la password dell'utente corrente.

2.3.1 Versione di sviluppo

```
cd /opt
git clone https://github.com/tdm-project/tdm-edge.git

cd tdm-edge
git checkout develop
```

7.3 CONFIGURAZIONE SOFTWARE TDM

Il file di configurazione unica per i microservizi TDM è:

- '***/opt/tdm-edge/configs/tdm/tdm.conf***'.

Tale file non esiste al momento dell'installazione e va creato al primo avvio con i comandi presenti nella prossima sezione.

Come esempio si riporta il suo contenuto completo:

```
[EDGE_dispatcher]
mqtt_local_host = mosquitto
mqtt_local_port = 1883
mqtt_remote_host = tdm-broker-be.crs4.it
mqtt_remote_port = 1883
logging_level = 0

[FEINSTAUB_publisher]
mqtt_host = mosquitto
mqtt_port = 1883
logging_level = 0
influxdb_host = influxdb
influxdb_port = 8086

[IOTAWATT_publisher]
mqtt_host = mosquitto
mqtt_port = 1883
logging_level = 0
influxdb_host = influxdb
influxdb_port = 8086

[HTU21D_publisher]
```

```
mqtt_host = mosquitto

mqtt_port = 1883

interval=10

logging_level = 0
```

Gli unici campi che possono necessitare di modifiche sono:

- Sezione '**EDGE_dispatcher**':
  - *mqtt_remote_host*: indirizzo del broker TDM, potrebbe cambiare, previa indicazione sul sito TDM;
  - *mqtt_remote_port*: posta del broker TDM, potrebbe cambiare, previa indicazione sul sito TDM;

7.3.1 CREAZIONE DEL FILE DI CONFIGURAZIONE

Il file di configurazione non è presente all'installazione e va creato successivamente:

```
cat > /opt/tdm-edge/configs/tdm/tdm.conf <<EOF
[EDGE_dispatcher]

mqtt_local_host = mosquitto

mqtt_local_port = 1883

mqtt_remote_host = tdm-broker-be.crs4.it

mqtt_remote_port = 1883

logging_level = 0


[FEINSTAUB_publisher]

mqtt_host = mosquitto

mqtt_port = 1883

logging_level = 0

influxdb_host = influxdb

influxdb_port = 8086


[IOTAWATT_publisher]

mqtt_host = mosquitto
```

```
mqtt_port = 1883

logging_level = 0

influxdb_host = influxdb

influxdb_port = 8086


[HTU21D_publisher]

mqtt_host = mosquitto

mqtt_port = 1883

interval=10

logging_level = 0

EOF
```

7.3.2 MODIFICA DEL FILE DI CONFIGURAZIONE

Il file di configurazione del software TDM può essere modificato successivamente alla creazione col comando '*nano*':

```
nano /opt/tdm-edge/configs/tdm/tdm.conf
```

Una volta terminate le modifiche digitare:

- **CTRL+X,** successivamente **Y** e **INVIO** per salvare il file ed uscire
- **CTRL+X**, successivamente **N** e **INVIO** per uscire senza salvare il file.

7.4 AVVIO DEI SERVIZI

Una volta ultimato l'editing dei file di configurazione si è pronti per l'avvio di tutti i servizi.

Per poterli avviare in background digitare i seguenti comandi:

```
cd /opt/tdm-edge/
docker-compose up -d
```

Una volta avviati i servizi è possibile verificare che siano tutti online digitando il comando

```
cd /opt/tdm-edge/

docker-compose ps
```

Se non ci sono stati problemi si avrà un output simile al seguente:

```
[alarm@tdm-edge-cfa703f4 ~]$ cd /opt/tdm_edge/compose/standard/
[alarm@tdm-edge-cfa703f4 standard]$ docker-compose ps
        Name                      Command               State           Ports
-------------------------------------------------------------------------------------
tdm_dispatcher          src/entrypoint.sh -c /opt/ ...   Up
tdm_feinstaub           src/entrypoint.sh -c /opt/ ...   Up      0.0.0.0:8089->5000/tcp
tdm_grafana             /run.sh                          Up      0.0.0.0:80->3000/tcp
tdm_htu21d_publisher    src/entrypoint.sh -c /opt/ ...   Up
tdm_influxdb            /usr/bin/entry.sh /run.sh        Up      8086/tcp
tdm_iotawatt            src/entrypoint.sh -c /opt/ ...   Up      0.0.0.0:8088->5000/tcp
tdm_mosquitto           usr/sbin/mosquitto -c /mos ...   Up
[alarm@tdm-edge-cfa703f4 standard]$
```

con tutti i servizi nello stato Up.

7.4.1 ACCESSO ALLA DASHBOARD LOCALE

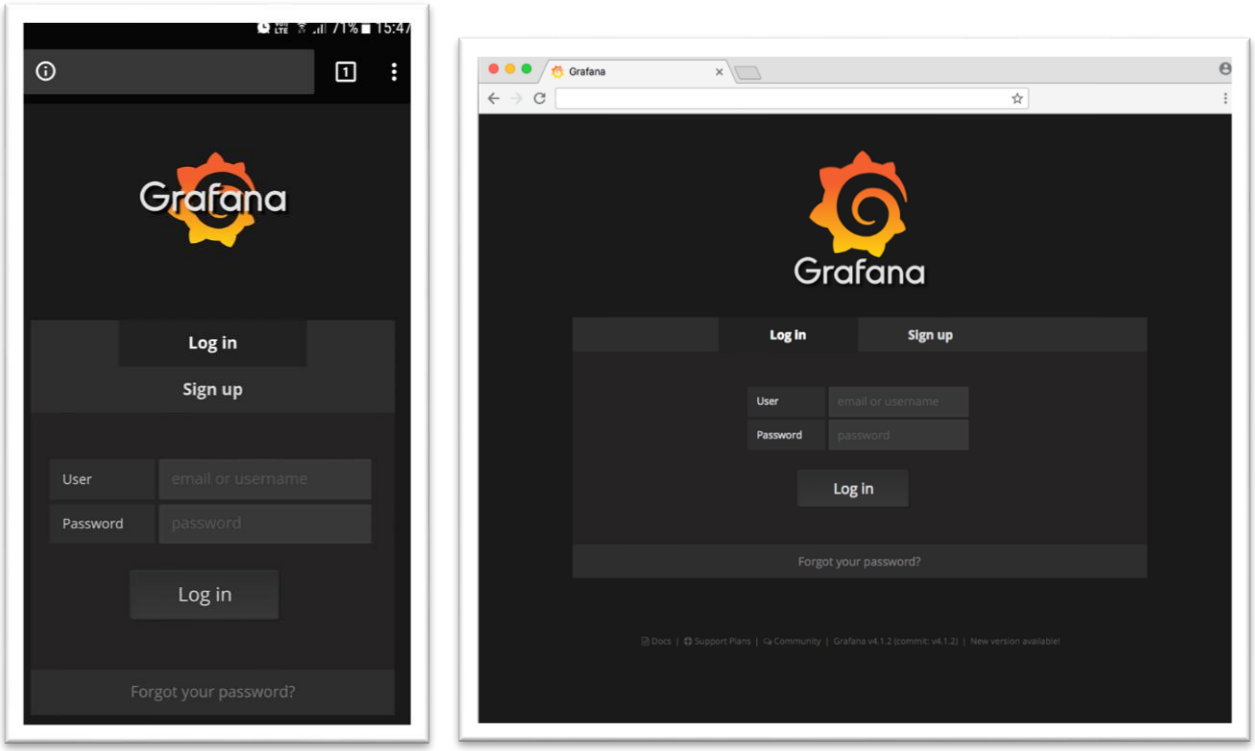
Da questo momento, se tutti i passaggi sono stati effettuati correttamente ci si può collegare alla dashboard locale 'Grafana'.

1. In un web browser, da un PC collegato alla stessa rete dell'Edge Gateway, aprire la pagina all'indirizzo IP dell'Edge Gateway ricavato nel punto 'Configurazione WiFi domestica sull'Edge Gateway'.
2. Alla richiesta di nome utente e password, inserire:

```
User:       admin
Password:   admin
```

3. Da questo punto è possibile configurare Grafana per accedere al database locale InfluxDb e configurare la dashboard.

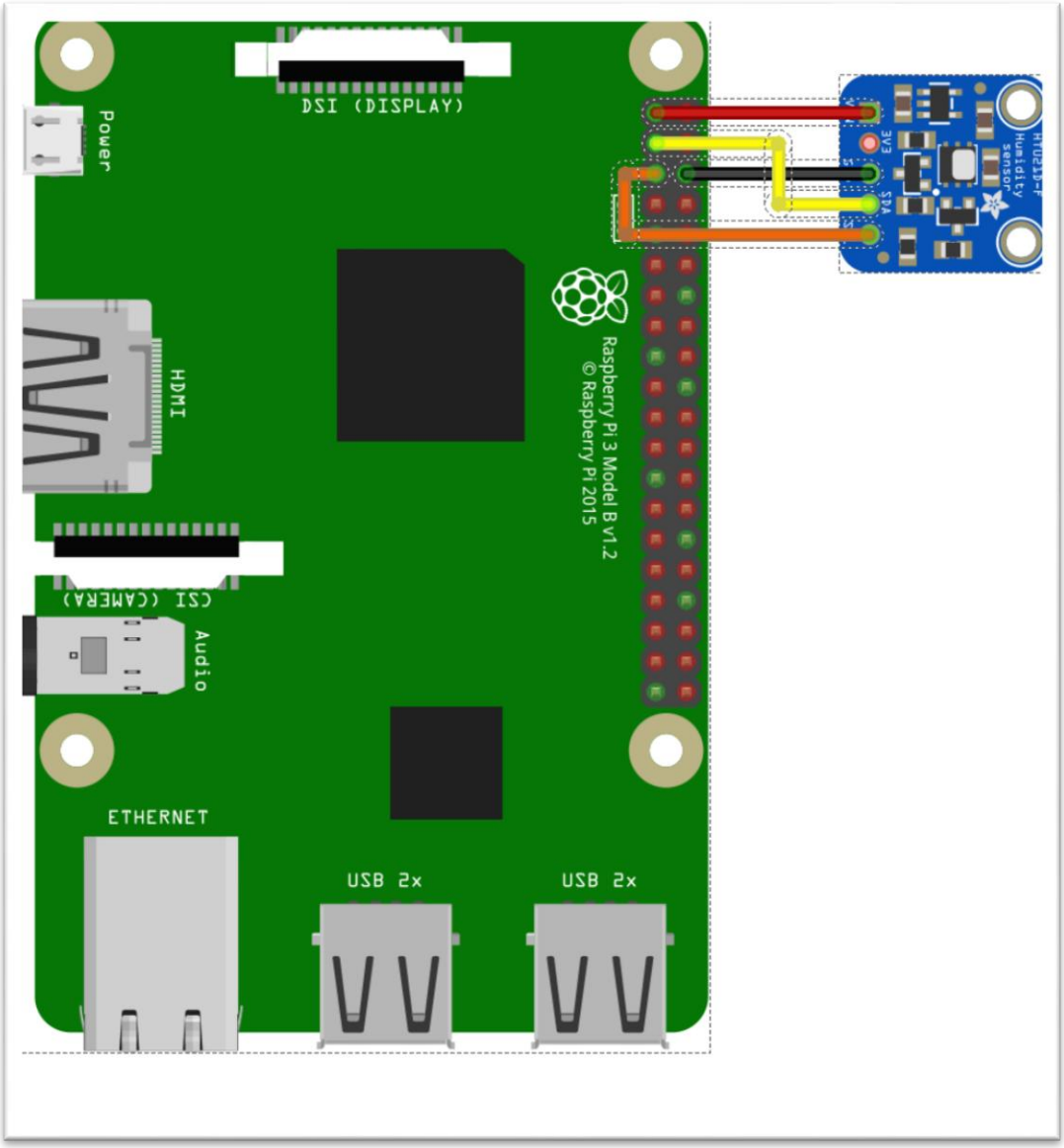
Documentazione ufficiale di Grafana con istruzioni per la configurazione (in lingua inglese): http://docs.grafana.org/.



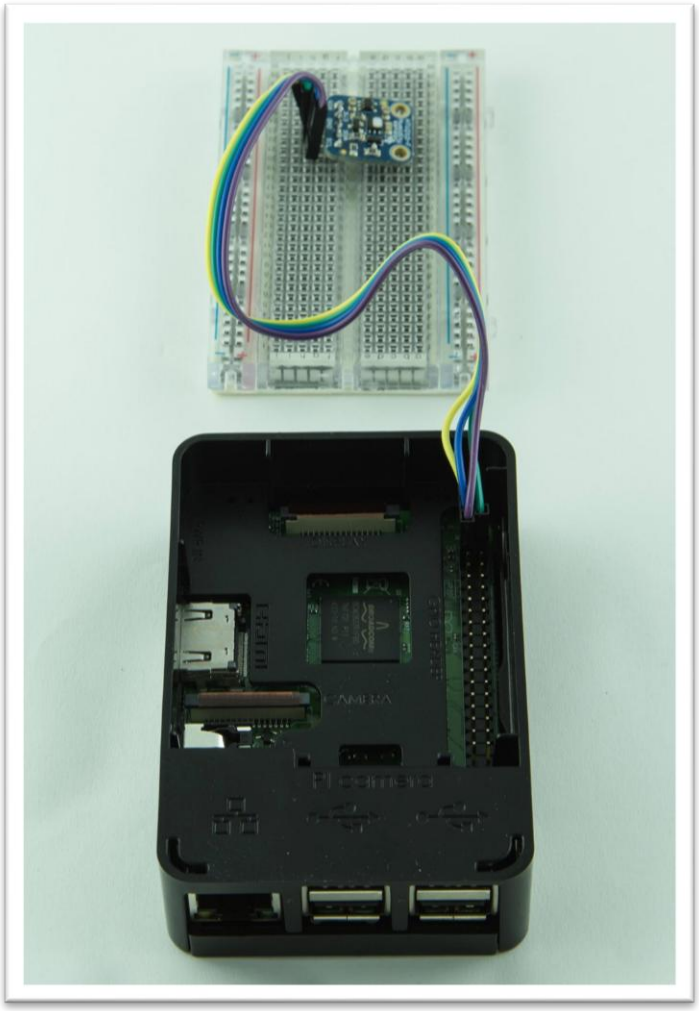
Pagina di accesso di Grafana: Smartphone (sinistra) e PC (destra)

7.5 CABLAGGIO DI ESEMPIO CON SENSORE INTEGRATO

Il modulo raspberry pi 3 può essere corredato da uno o più sensori. A titolo d'esempio si consideri il collegamento di un sensore di temperatura e umidità HTU21D mediante bus I2C. Nella figura seguente si può osservare il dettaglio del collegamento:



Le connessioni gialle e arancio si riferiscono rispettivamente ai segnali SDA e SCK.

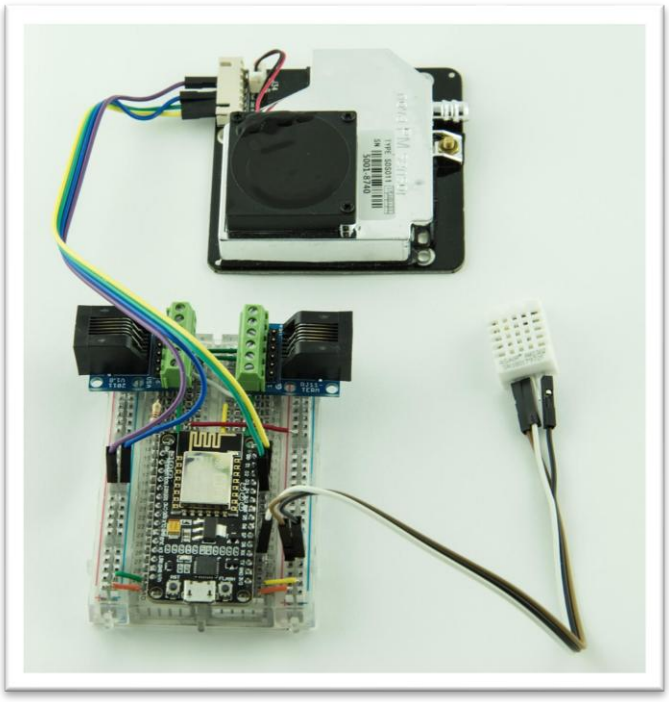


Edge Gateway (Raspberry Pi 3) in test con sensore HTU21D

7.6 CONFIGURAZIONE DELLE STAZIONI DI MISURA

Per lo sviluppo del prototipo si sono utilizzate due stazioni di misura remote, collegate all'Edge Gateway attraverso rete WiFi dell'Edge stesso o rete WiFi domestica disponibile sul sito di installazione. Di seguito verranno fornite le specifiche per il collegamento, per ulteriori dettagli sulle stazioni e loro installazione e configurazione si rimanda alla documentazione propria della stazione.

7.6.1 STAZIONE METEO ‘STUTTGART FINE DUST SENSOR’



Stazione SFDS con sensore temperatura/umidità e particolato/polveri sottili

La stazione meteo SFDS si collega con l’Edge Gateway attraverso la rete WiFi propria dell’Edge Gateway o quella domestica, usando il protocollo del database InfluxDB. I parametri da impostare nella schermata di configurazione sono:

```
Server InfluxDB:    <Indirizzo_IP_domestico_dell'Edge>
Porta InfluxDB:     8089
Username:           utente    impostato    per    il    DB    (attualmente    'root')
Password:           password impostata per il DB (attualmente 'root')
```

L’indirizzo IP domestico dell’Edge è quello ricavato nella sezione ‘Configurazione WiFi domestica sull’Edge Gateway’. Se si vuole utilizzare la rete WiFi usando l’Edge Gateway come Access Point:

```
Server InfluxDB:    192.168.2.1
```

Manuale di costruzione e configurazione della stazione di misura meteo (in lingua inglese): https://luftdaten.info/en/construction-manual/.

7.6.2 STAZIONE DI MONITORAGGIO ENERGETICO ‘IOTAWATT’

ATTENZIONE! L'elettricità è pericolosa.

Questo sezione descrive un dispositivo che utilizza sensori in prossimità di circuiti elettrici in tensione. L'installazione dei sensori all'interno dei quadri elettrici di distribuzione deve essere eseguita esclusivamente da un elettricista qualificato. Gli altri lo fanno a proprio rischio e pericolo. Le avvertenze contenute in questa guida non devono essere considerate un sostituto dell'esperienza e della formazione. Si prega di fare attenzione agli errori e di rivolgersi a un tecnico qualificato per tutte le procedure relative ai collegamenti in tensione di rete. Per la documentazione tecnica relativa alla stazione ‘IotaWatt’ si faccia riferimento alla documentazione ufficiale.

Stazione IotaWatt con sensori collegati

Di seguito sono riportate le configurazioni da inserire per il collegamento all’Edge Gateway.

```
Server InfluxDB:    <Indirizzo_IP_domestico_dell'Edge>
Porta InfluxDB:     8088
Username:           utente    impostato    per    il    DB    (attualmente    'root')
Password:           password impostata per il DB (attualmente 'root')
```

L’indirizzo IP domestico dell’Edge è quello ricavato nella sezione ‘Configurazione WiFi domestica sull’Edge Gateway’. Se si vuole utilizzare la rete WiFi usando l’Edge Gateway come Access Point:

```
Server InfluxDB:    192.168.2.1
```

Documentazione ufficiale della stazione Iotawatt con istruzioni per l’installazione, configurazione e collegamento agli schemi e sorgenti(in lingua inglese): https://guide.openenergymonitor.org/setup/iotawatt/ .

7.7 CONCLUSIONI E RIFERIMENTI

Il presente manuale rappresenta il Refence Design e la sua implementazione prototipale al momento della presentazione del Deliverable. Trattandosi di un sistema in fase di continuo sviluppo e integrazione, le informazioni qui presentate potrebbero essere soggette a variazioni in futuro. Successivi aggiornamenti alla documentazione saranno messi a disposizione attraverso il portale del progetto http://www.tdm-project.it/ e il repository GitHub https://github.com/tdm-project/tdm-edge.