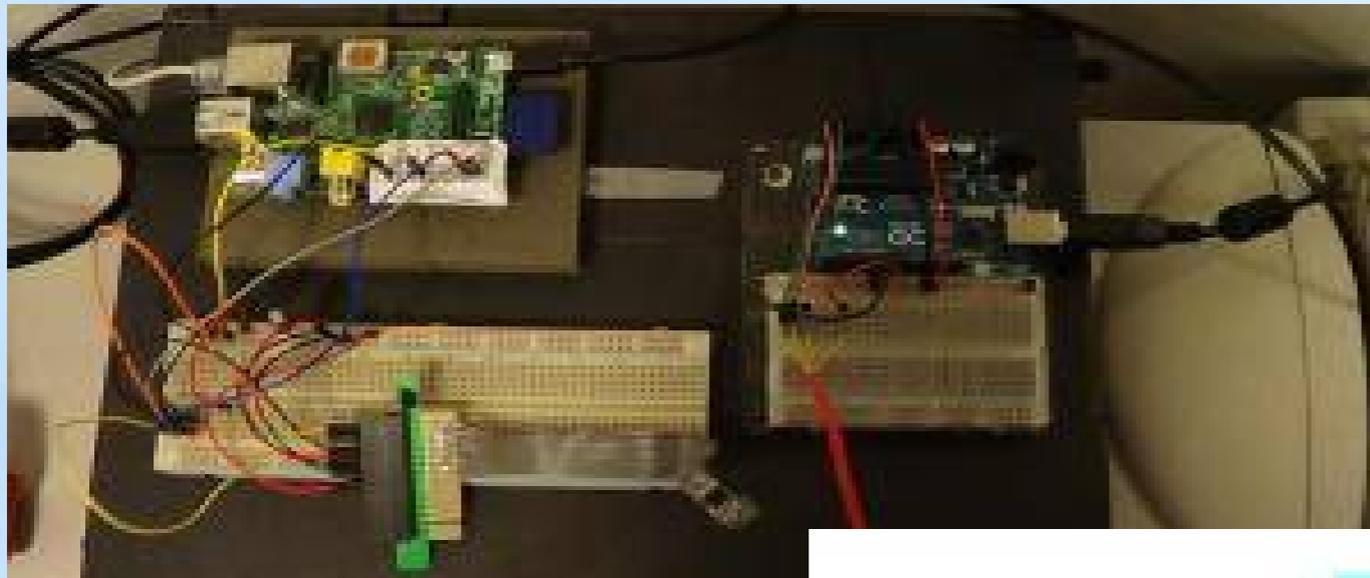




Che Stima per chi Stima!

rete monitoraggio ambientale partecipativo



Reggio Emilia, 2015-10-24

Raspibo per progetto RMAP - [HTTP://rmap.cc](http://rmap.cc)

Patruno, *Che Stima per chi Stima ! Rete Monitoraggio Ambientale Partecipativo*





Smart city, smart citizen e citizen science

- Smart city: territorio urbano che permette di soddisfare le esigenze dei cittadini, delle imprese e delle istituzioni, mediante l'ausilio di strumenti innovativi e partecipazione attiva
- Ad esempio, reti di sensori per il controllo di parametri ambientali.
- Per la Comunità Europea, il grado di intelligenza di una città dovrebbe essere valutato secondo economia, mobilità, ambiente, persone, tenore di vita e governo.
- All'interno di questo ambiente, il cittadino può (deve) avere un ruolo attivo (smart citizen)
- Un esempio di partecipazione attiva può essere ritrovata nella citizen science, una modalità di ricerca scientifica condotta totalmente o in parte da scienziati non professionisti





Inquinamento

- la legislazione europea dice che in una città come Bologna bastano due o tre stazioni per la misura della qualità dell'aria
- sono sufficienti per monitorare gli aspetti generali dell'inquinamento urbano, poi però esistono punti di accumulo locali (sotto i portici, nelle strade strette), differenze tra il primo e l'ultimo piano, emissioni locali (impianti), anche episodiche (cantieri, ingorghi, caminetti accesi), inquinamento dentro le case...
- non basterebbero centinaia di centraline per monitorare tutta questa varietà di situazioni, e i costi delle reti di misura sono già adesso difficili da sostenere
- l'Agenzia Ambientale Europea promuove l'integrazione di strumenti diversi: satelliti, modelli, poche stazioni con strumenti avanzati e reti di microsensori a basso costo gestite da volontari (scuole, associazioni, cittadini)





per un problema multi-scala, monitoraggio multi-scala

- perciò sì, qualche grande pennello e poi pennelli piccoli per definire i dettagli





Obiettivi R-map

- Raccogliere e distribuire dati ambientali rilevati dai cittadini
- Rendere disponibili questi dati ai servizi meteorologici, alle agenzie di prevenzione ambientale, alla protezione civile e istituti di ricerca
- Fornire feedback ai fornitori di dati in modo che essi abbiano servizi per testare e migliorare la qualità dei dati
- Divulgazione scientifica e sensibilizzazione ai temi ambientali
- Coinvolgimento di scuole e università a scopi formativi
- Creare un circolo virtuoso tra Enti Formativi, Pubbliche Amministrazioni, Aziende private e cittadini.





Soggetti coinvolti

- **ARPA Emilia Romagna SIMC**

- Predisporre un disciplinare tecnico rispetto ai metodi di misura, elaborazione dei dati dei sensori e loro collocazione
- Definisce i protocolli e i formati per la comunicazione dei dati
- Eseguire un eventuale controllo di qualità e comunica i risultati

- **ARPA Regione Veneto**

- Supporta la diffusione degli standard/fornisce infrastruttura hardware
- Contribuisce allo sviluppo e sperimenta stazioni commerciali con supporto allo standard Rmap
- Sperimenta nuova sensoristica a basso/medio costo

- **Cineca Consorzio Interuniversitario per il supercalcolo e l'innovazione tecnologica**

- Sperimenta e sviluppa le tecnologie Stima
- Supporto BigData
- Fornisce servizi a valore aggiunto





Soggetti coinvolti

- **Dipartimento informatica Università di Bologna**
 - Esprime pareri sul progetto e prototipo hardware e software
 - Eventuali tesi/tirocini sul progetto software
- **RaspiBO**: gruppo informale di appassionati di elettronica ed informatica libera della zona di Bologna
 - Realizzano un prototipo hardware e software
 - Sperimentano il prototipo
 - Documentano la realizzazione del prototipo come openhardware e distribuiscono il software con licenza libera
- **Soggetto privato / startup**
 - Progetto commerciale per la vendita, installazione e manutenzione delle stazioni
- **Scuole**
- **Soggetti già attivi sul territorio (Meteonetwork)**





Cosa è Rmap

- Un insieme di specifiche:
- **Protocollo di rilevamento dati**
 - Collocazione sensori
 - Accuratezza dei sensori
 - Elaborazioni

http://www.raspibo.org/wiki/index.php/Gruppo_Meteo/DisciplinareStazione

- **Sistema per lo scambio dati**
 - Protocolli di trasmissione
 - Formati dati
 - Metadati

http://www.raspibo.org/wiki/index.php/Gruppo_Meteo/RFC-rmap





Implementazioni hardware e software

- Specifiche realizzazioni che aderiscono allo standard Rmap
- Possibilmente open hardware e open software

STIMA

Acronet





Specifiche Rmap





Campionamento di variabili meteorologiche

- Campionamento è il processo per ottenere una discretizzata sequenza di misure di una quantità.
- Campione è una singola misura, tipicamente una di una serie di letture “spot” di un sistema di sensoristica.
- Una osservazione (misurazione) è il risultato del processo di campionamento. Nel contesto di analisi di serie, un'osservazione è derivato da un numero di campioni.
- Variabili atmosferiche come la velocità del vento, temperatura, pressione e umidità sono funzioni di quattro dimensioni - due orizzontali, una verticale e una temporale. Esse variano irregolarmente in tutte e quattro, e lo scopo dello studio del campionamento è quello di definire le procedure di misura pratiche per ottenere osservazioni rappresentative con incertezze accettabili nelle stime delle medie e variabilità.





Data Level

- Dati **Level I** , sono le letture dirette degli strumenti espresse in appropriate unità fisiche e georeferenziate
- Dati **Level II**, dati riconosciuti come variabili meteorologiche; possono essere ottenuti direttamente da strumenti o derivati dai dati Level I
- Dati **Level III** sono quelli contenuti in dataset internamente consistenti, generalmente su grigliato.
- I dati scambiati a livello internazionale sono livello II o livello III





Disciplinare per il rilevamento di dati

Per ora una proposta per:

- **Schermi dalla radiazione**
- **Temperatura**
- **umidità**

Prima prebozza disponibile a:

- http://www.raspibo.org/wiki/index.php/Gruppo_Meteo/DisciplinareStazione





Protocolli per R-map

- **MQTT** (Message Queue Telemetry Transport) è un protocollo publish/subscribe particolarmente leggero, adatto per la comunicazione M2M tra dispositivi con poca memoria o potenza di calcolo e server o message broker.
- **AMQP** (Advanced Message Queuing Protocol) è protocollo per comunicazioni attraverso code di messaggi. Sono garantite l'interoperabilità, la sicurezza, l'affidabilità, la persistenza. Nella sua implementazione Rabbitmq esporta un broker MQTT e fornisce delle api web
- Json è il formato per il payload

E' fondamentale:

- Integrazione con le funzioni e le specifiche richieste dalle applicazioni per la **domotica**
- Integrazione con applicazioni per la **telefonia mobile** per la rilevazione dello spessore neve e altri parametri





Conceptual data model

These models, sometimes called domain models, are typically used to explore domain concepts with project stakeholders

DB-All.e Conceptual data model

- Il modello è orientato all'applicazione (bisogna capire cosa sono i dati, normalizzarli e ricondurli a metadati standard in fase di accoglienza), quindi si lavora pre e non post
- I dati sono legati ai metadati in modo univoco
- Una osservazione è univoca nello spazio dei suoi metadati
- L'unica possibilità di far coesistere due osservazioni dello stesso parametro nello stesso punto è attraverso il metadato "network" associabile alla classe dello strumento
- La tracciabilità di un sensore, una stazione, un osservatore nello spazio, nel tempo etc. Avviene attraverso il metadato "ident"
- Alcuni metadati sono table driven (level, timerange, network)





- Ogni dato può essere associato a un certo numero di attributi
- Nessuna dimensione è vincolata (intervalli temporali tra dati, numero attributi....)
- E' contemplata la gestione di previsioni; il datetime è sempre quello di verifica
- Misure e metadati hanno troncamenti sulle cifre significative dettati dalla loro possibilità reale di misura e stabiliti a priori
- Esistono due categorie di dato: una che varia tutti i metadati (osservazioni classiche) e l'altra che non prevede l'uso di alcuni metadati e che quindi sono da considerarsi come ulteriori metadati di quella singola stazione (constant station data: es. Nome stazione)
- Nessuno vieta di espandere i metadati estendendo esternamente questo data model





Logical data model (LDM)

LDMs are used to explore the domain concepts, and their relationships, of your problem domain

DB-All.e LDM

- **METADATI**

- **Datetime:** tempo di fine misurazione
- **Ana:** **Longitudine, latitudine** ed un **identificativo**
- **network:** definisce stazioni con caratteristiche omogenee (classe degli strumenti)
- **Time range:** Tr,P1,P2 indica osservazione o tempo previsione ed eventuale elaborazione "statistica"
- **Level:** TL1,L1,TL2,L2 le coordinate verticali (eventualmente strato)
- **Variable:** Btable parametro fisico

- **DATI**

- Valori rappresentabili come interi, reali, doppia precisione, stringhe
 - **Attributi** (alla stregua di dati)





Metadati su MQTT

- Ogni topic corrisponde ai metadati univoci, mentre il payload è composto dal valore e dall'istante temporale
- **/IDENT/COORDS/NETWORK/TRANGE/LEVEL/VAR**
 - **IDENT**: identificativo per stazioni mobili, “-” per stazioni fisse
 - **COORDS**: nella forma *lon,lat*. Le coordinate sono espresse nell forma *int(valore*10⁵)* con eventuale segno negativo
 - **NETWORK**: massimo 16 caratteri
 - **TRANGE**: nella forma *indicator,p1,p2*
 - *Indicator* e *p2* interi senza segno, *p1* intero con eventuale segno negativo. “-” per valori non significativi
 - **LEVEL**: nella forma *type1,l1,type2,l2*
 - *Type1*, *type2* interi con eventuale segno negativo, *l1* e *l2* interi con eventuale segno negativo. “-” per valori non significativi
 - **VAR**: nella forma *BXXYYY*
- Il payload è in formato JSON: `{ “v”: VALUE, “t”: TIME, “a”: { “BXXYYY”: VALUE, ... } }`
 - *VALUE*: valore in formato CREX
 - *TIME*: formato YYYY-mm-ddTHH:MM:SS.MSC (secondi e millisecondi opzionali)
 - Gli attributi (“a”) sono opzionali





RMAP web services

Composizione degli URL per un HTTP GET request

La "base" della richiesta è quella standard:

`/version/ident/coords/network/timerange/level/bcode/`

Ad esempio:

`http://rmap.cc/v0.1/-/1207738,4460016/locali/
254,0,0/103,2000,-,-/B12101`





Serie temporale

Serie temporale mensile, giornaliera e annuale:

/ident/coords/network/timerange/level/bcode/timeseries/year

/ident/coords/network/timerange/level/bcode/timeseries/year/month

/ident/coords/network/timerange/level/bcode/timeseries/year/month/day

Ad esempio:

/-/1207738,4460016/locali/254,0,0/103,2000,-,-/B12101/timeseries/2011

/-/1207738,4460016/locali/254,0,0/103,2000,-,-/B12101/timeseries/2011/01

/-/1207738,4460016/locali/254,0,0/103,2000,-,-/B12101/timeseries/2011/01/13





Serie spaziale e sommario

Serie spaziale di una rete, con granularità oraria (± 30 minuti):

`/**/NETWORK/TIMERANGE/LEVEL/BCODE/spatialseries/YEAR/MONTH/DAY/HOUR`

Riassuntivo

`/**/**/**/**/**/summaries`

`/**/**/**/**/**/summaries`

- `/**/NETWORK/**/**/summaries`
- `/-/COORDINATES/NETWORK/**/**/summaries`
- `/IDENT*/NETWORK/**/**/summaries`
- `/**/NETWORK/TIMERANGE/LEVEL/BCODE/summaries/YEAR/MONTH`
- `/**/NETWORK/TIMERANGE/LEVEL/BCODE/summaries/YEAR/MONTH/DAY`





Bufr

- The Binary Universal Form for the Representation of meteorological data (BUFR) is a binary data format maintained by the World Meteorological Organization (WMO).
- BUFR was designed to be portable, compact, and universal. Any kind of data can be represented, along with its specific spatial/temporal context and any other associated metadata. In the WMO terminology, BUFR belongs to the category of table-driven code forms, where the meaning of data elements is determined by referring to a set of tables that are kept and maintained separately from the message itself.
- Descriptors: all element descriptors will be found in BUFR specification section known as "Table B". The Table B definition of an element descriptor includes its number, short text definition, decoding parameters (bit width, scale factor, and bias), and type (numerical, character string, code table, etc.).





Software di decodifica Bufr

- **WREPORT**: a featureful C++ library for BUFR and CREX encoding and decoding
<http://sourceforge.net/p/wreport/home/Home/>
- **GRIBAPI ecCodes**
<https://software.ecmwf.int/wiki/display/ECC/ecCodes+Home>





Formato GeoJSON per punti sparsi

- <http://geojson.org/>
- E' un formato molto verboso (soprattutto per le serie temporali) ma
- Rispetta il data model che ci siamo dati (anche se in modo non ottimale)
- E' uno dei formati vettoriali interpretati da **GDAL/OGR**
 - OpenLayers
 - QGIS
 - ...
- Fornito mediante postprocessatore **json** di Arkimet, ma disponibile anche come tool da riga di comando





Formato JSON

- <http://www.json.org/json-it.html>
- JSON (JavaScript Object Notation) è un semplice formato per lo scambio di dati. Per le persone è facile da leggere e scrivere, mentre per le macchine risulta facile da generare e analizzarne la sintassi.
- Rispetta il data model che ci siamo dati
- Ogni elemento è il report con i dati di una certa stazione per un certo istante di riferimento.
- Fornito mediante postprocessatore **json** di Arkimet, ma disponibile anche come tool da riga di comando





Il sistema di monitoraggio sviluppato da RaspiBO

Stima





Vista d'insieme

- **Open hardware e free software**
- **Utilizzo delle piattaforme più diffuse e board di prototipizzazione**
- **Disegni hardware con Kicad (by Daniele Castellari)**
- **5 moduli hardware che soddisfano differenti esigenze**
 - **Consumi:** alimentazione tramite rete, batterie con pannello solare, batterie
 - **Collegamento:** all'interno della casa, in esterno con un cavo ethernet e PoE, in esterno in postazione fissa ravvicinata, in esterno in postazione mobile
 - **Domotica:** sono collegabili attuatori
- **Un modulo con funzioni server**
 - Database
 - Web server
 - NTP server
 - Sviluppo





E' anche un “framework” per makers

- E' possibile utilizzare il modulo base per lo sviluppo del firmware
- Sono installate le librerie Arduino con i file di specifiche per i microcontrollori utilizzati
- Come build system si può utilizzare INO in una versione con alcune patch
- Tutte le librerie personalizzate sono già installate
- L'aggiornamento avviene tramite git / pip
- E' possibile “scriptare” il build, upload e configurazione delle board





Modulo base: vista d'insieme

circa 75€

- Sviluppato su **Raspberry**
- Distribuzione Pidora 2014 (fedora 20)
- Software completamente pacchettizzato RPM
- Repository software pubblico
<http://rmapv.rmap.cc/repo/rmap/fedora/20/RPMS/arm/repoview/>
- Per ora una immagine SD da 8G scaricabile e pronta all'uso
- Il modulo gestisce direttamente la sensoristica su I2C
- Testata connessione alla LAN con Ethernet WIFI e GPS
- Gestione di una propria LAN con dhcp server, dns server e nat



Board con microcontroller: vista d'insieme

circa 50€ /100€



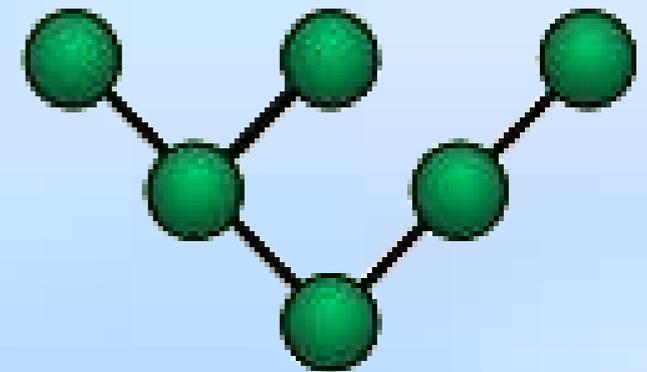
- Modulo master
 - Microduino core+ 1284p / arduino mega 2560
 - Breadboard/circuito stampato
 - scheda RTC
 - scheda mini ethernet ENC60 /microduino ENC / scheda SIM900
 - scheda radio RF24
 - cavo power over ethernet
 - stabilizzatore dc/dc switched
 - sensori
- Modulo satellite Modulo Bluetooth
 - microduino core+ 644p @5V 16MHz
 - microduino RF24 con antenna / scheda Bluetooth
 - microduino seriale USB FT232R
 - sensori





Trasporti

- Trasporto **Bluetooth** (HC-05)
- Trasporto **Seriale**
 - Principalmente per configurazione e debug
 - Piccole distanze via cavo
- Trasporto **TCP/IP**
 - Collegamenti tramite cavo ethernet a breve e media distanza
- Trasporto **RF24Network**
 - OSI Network Layer using nRF24L01(+) radios 2.4GHz ISM
 - 50/150m in aria libera
 - Host Addressing. Each node has a logical address on the local network.
 - Message Forwarding. Messages can be sent from one node to any other, and this layer will get them there no matter how many hops it takes.
 - Ad-hoc Joining. A node can join a network without any changes to any existing nodes.



Tree





RF24Network Addressing and Topology

Each node must be assigned an 15-bit address by the administrator. This address exactly describes the position of the node within the tree. The address is an octal number. Each digit in the address represents a position in the tree further from the base.

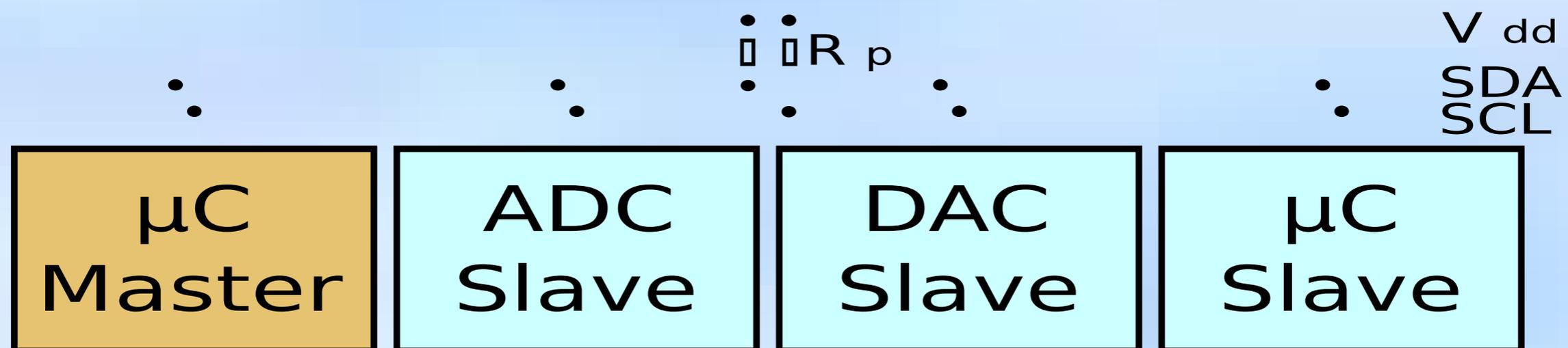
- Node 00 is the base node.
 - Nodes 01-05 are nodes whose parent is the base.
 - Node 021 is the second child of node 01.
 - Node 0321 is the third child of node 021, and so on.
 - The largest node address is 05555, so 3,125 nodes are allowed on a single channel.
- Alla libreria distribuita è stata aggiunta la crittografia e frammentazione e ricomposizione del payload





BUS I2C

- Il protocollo i2c prevede l'utilizzo di un bus formato da due linee bidirezionali. Le due linee, chiamate "scl" e "sda" rispettivamente, trasportano la tempistica di sincronizzazione (chiamata anche "clock") e i dati.
- Abbiamo scelto il bus i2c in quanto:
 - È diventato lo standard di fatto per una serie di integrati tra cui i sensori
 - Si possono collegare fino a 127 dispositivi
 - La comunicazione è bidirezionale (read e write) con velocità assolutamente sufficienti per i nostri scopi





Per ora 4 tipi di trasporto

I moduli possono essere così caratterizzati dal trasporto supportato, se eseguono RPC, se richiedono RPC, se pubblicano su MQTT.

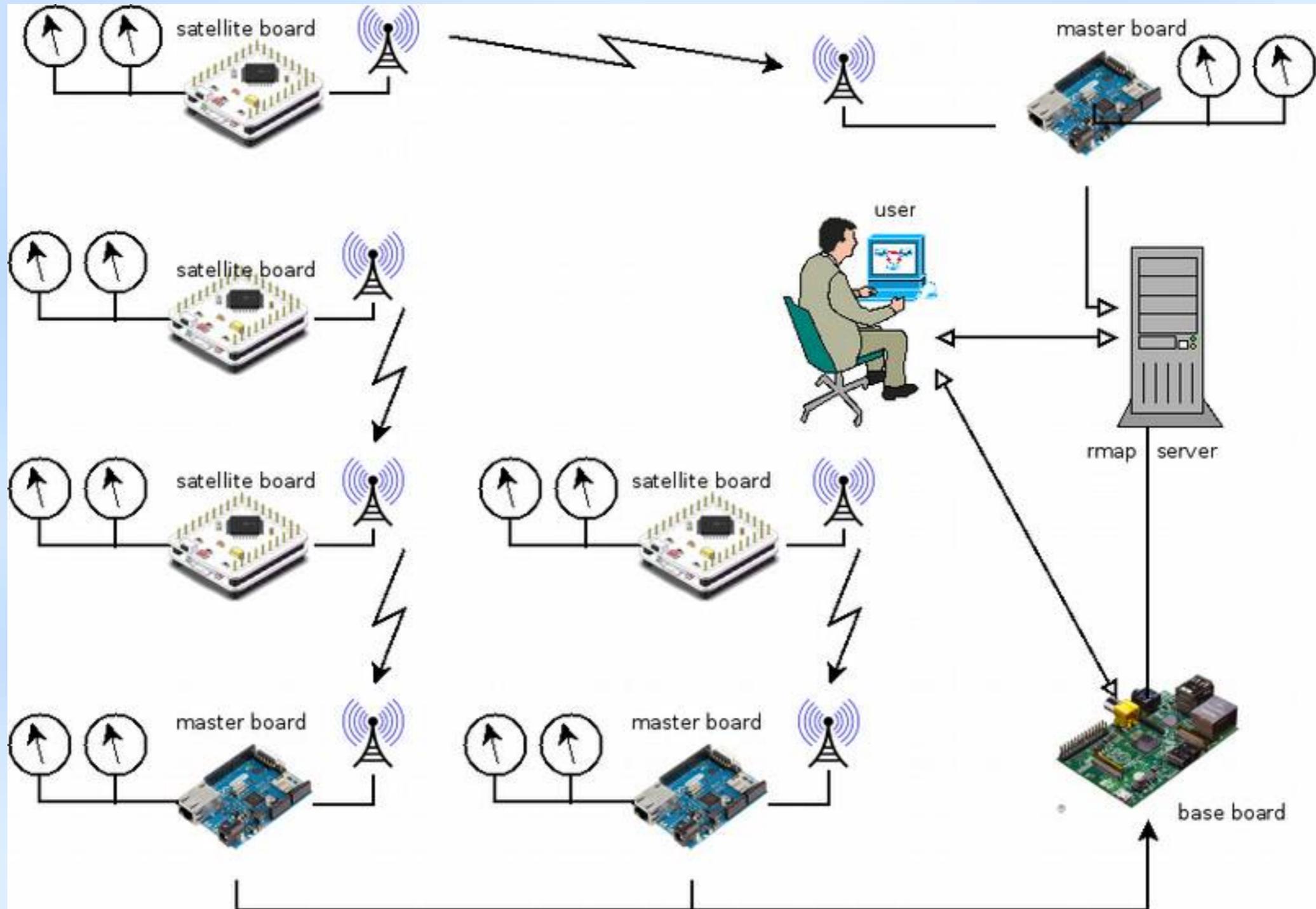
I moduli prototipati in r-map sono così denominati e caratterizzati:

	Attivo / Passivo	Trasp: Seriale	Trasp: TCP/IP	Trasp: radio RF24 Network	publish on MQTT	funzioni server	Alimentazione
Modulo Base	Attivo	Si	Si	Da sviluppare	Si	Si	Rete Batterie con pannello solare
Modulo Bluetooth	Passivo	Si	No	No	No	No	Batterie
Modulo master	Attivo / Passivo	Si	Si	Si	Si	No	Ethernet PoE
Modulo satellite	Passivo	Si	No	Si	No	No	Batterie (con pannello solare)
Modulo GPS	Attivo	Si	(Si)	Si	Si	No	Batterie con pannello solare





Schema comunicazione hardware





Json-rpc

- Json
 - JavaScript Object Notation ed è un formato adatto ad immagazzinare varie tipologie di informazioni, e quindi a scambiare queste informazioni tra applicazioni client/server.
 - JSON possiede una struttura semplicissima
- JSON-RPC is lightweight remote procedure call protocol similar to XML-RPC. It's designed to be simple!

Esempi:

```
--> {"jsonrpc": "2.0", "method": "subtract", "params": {"subtrahend": 23, "minuend": 42}, "id": 3}
```

```
<-- {"jsonrpc": "2.0", "result": 19, "id": 3}
```

```
--> {"jsonrpc": "2.0", "method": "subtract", "params": {"minuend": 42, "subtrahend": 23}, "id": 4}
```

```
<-- {"jsonrpc": "2.0", "result": 19, "id": 4}
```





Json-rpc un modo per fare tutto...

Questo un esempio di interrogazione e risposta di un sensore di temperatura

- SEND: {"jsonrpc":"2.0", "method":"getjson", "params":{"node":1, "type":"TMP", "driver":"I2C", "address":72},"id": 0}
- RECEIVE: {"jsonrpc":"2.0","result":{"B12101":30633},"id":0}

B12101 indica che il numero che segue è una temperatura in centesimi di gradi Kelvin, quindi 33.18 C.

Le remote procedure disponibili sono documentate sul wiki del progetto:
http://www.raspibo.org/wiki/index.php/Gruppo_Meteo/RemoteProcedures





Caratteristiche moduli master e satellite

- Firmware unico configurabile al tempo della compilazione
- Parametri di configurazione run time salvati su EEPROM
- Debug on serial insieme a json-rpc
- WatchDog con timeout di 8 sec
- Modalità sleep con interrupt sulla radio per modulo satellite
- RTC sincronizzato tramite NTP / GPS / http





Gestione dei sensori

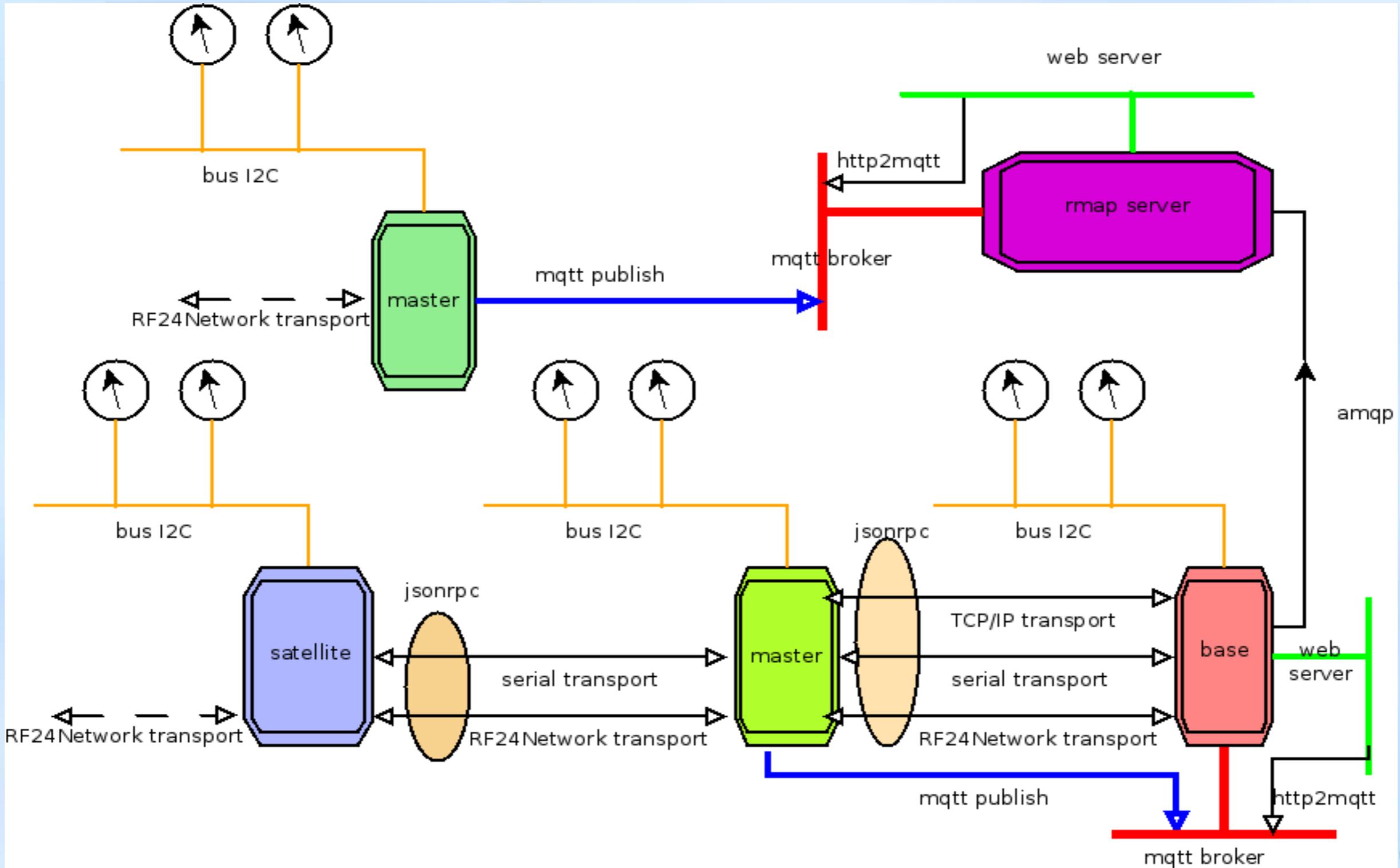
Libreria di "driver" per sensori

- Esistono attualmente due versioni, una in C++ e una in python
- Porta la gestione della sensoristica ad un livello di astrazione più alto. Aggiungere un nuovo tipo di sensore consiste nell'estendere una classe con quattro metodi per effettuare la lettura di quello specifico sensore:
 - int **setup**(int address);effettua eventuali settaggi necessari al funzionamento del sensore; esempio per temperatura: numero di bit di risoluzione, operazione di misura one-shot
 - int **prepare**(unsigned long* waittime); impartisce al sensore il comando per effettuare una singola misurazione torna il tempo in millisecondi di attesa necessario
 - int **get**(int* value); torna i valori della misurazione
 - jsonObject* **getJSON**() = 0; torna i valori in formato json





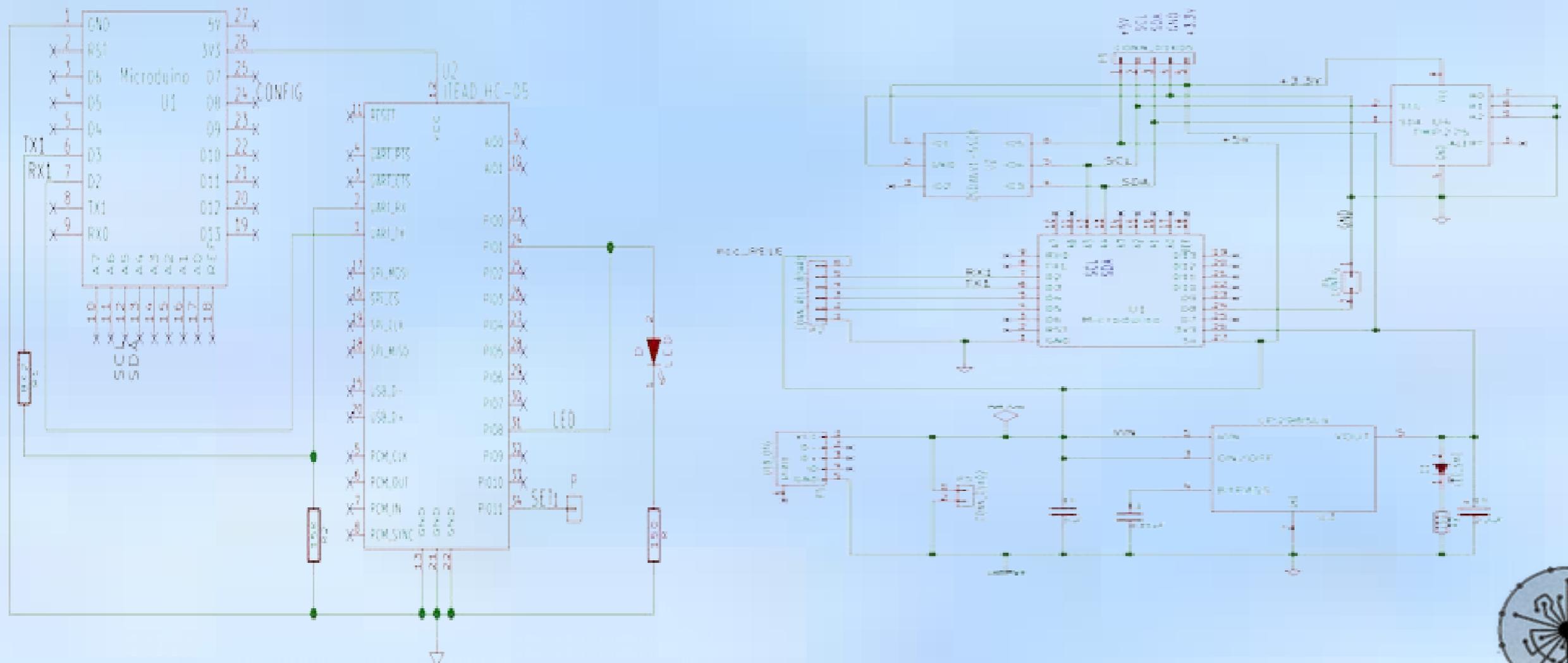
Schema comunicazione tra moduli

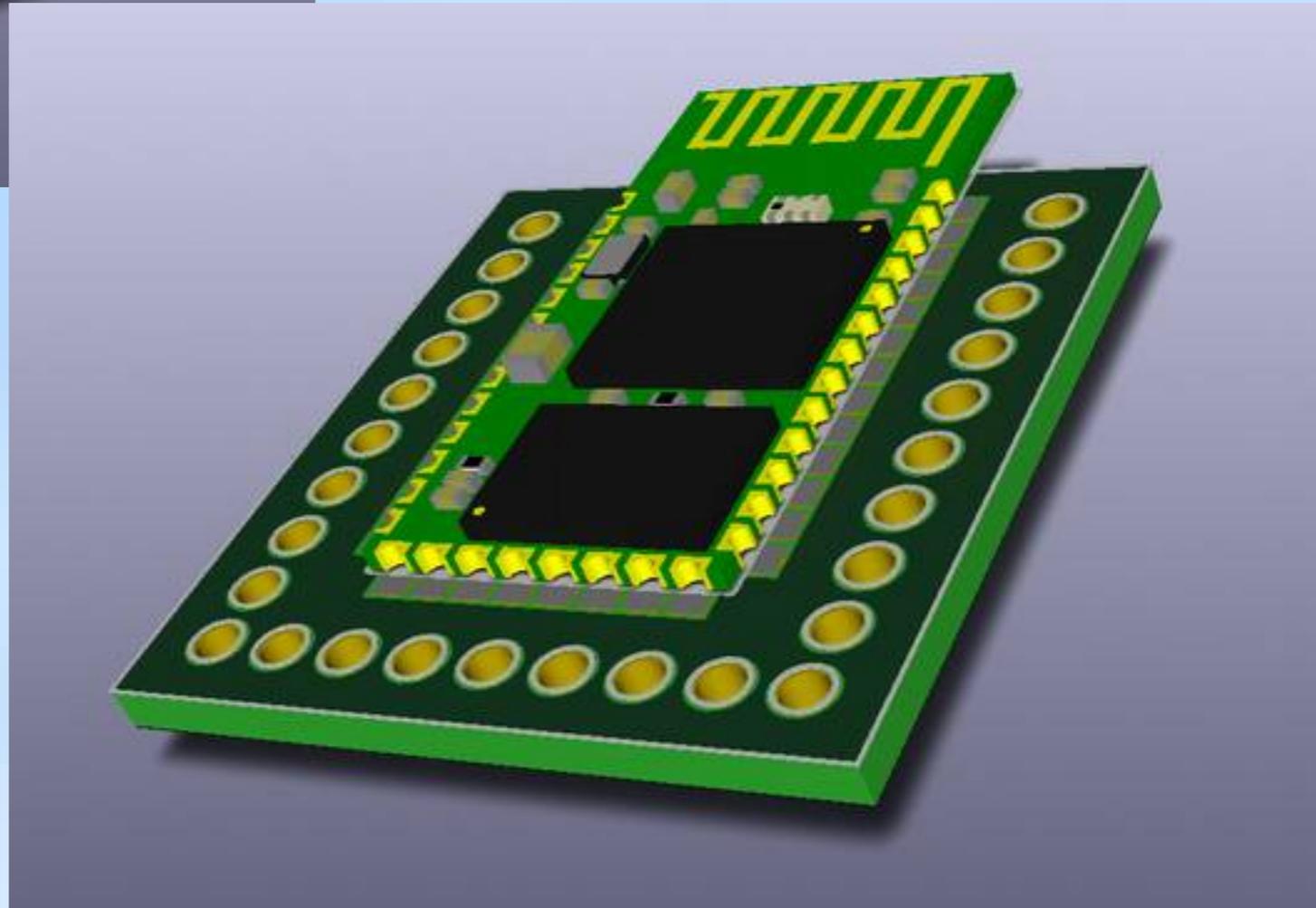
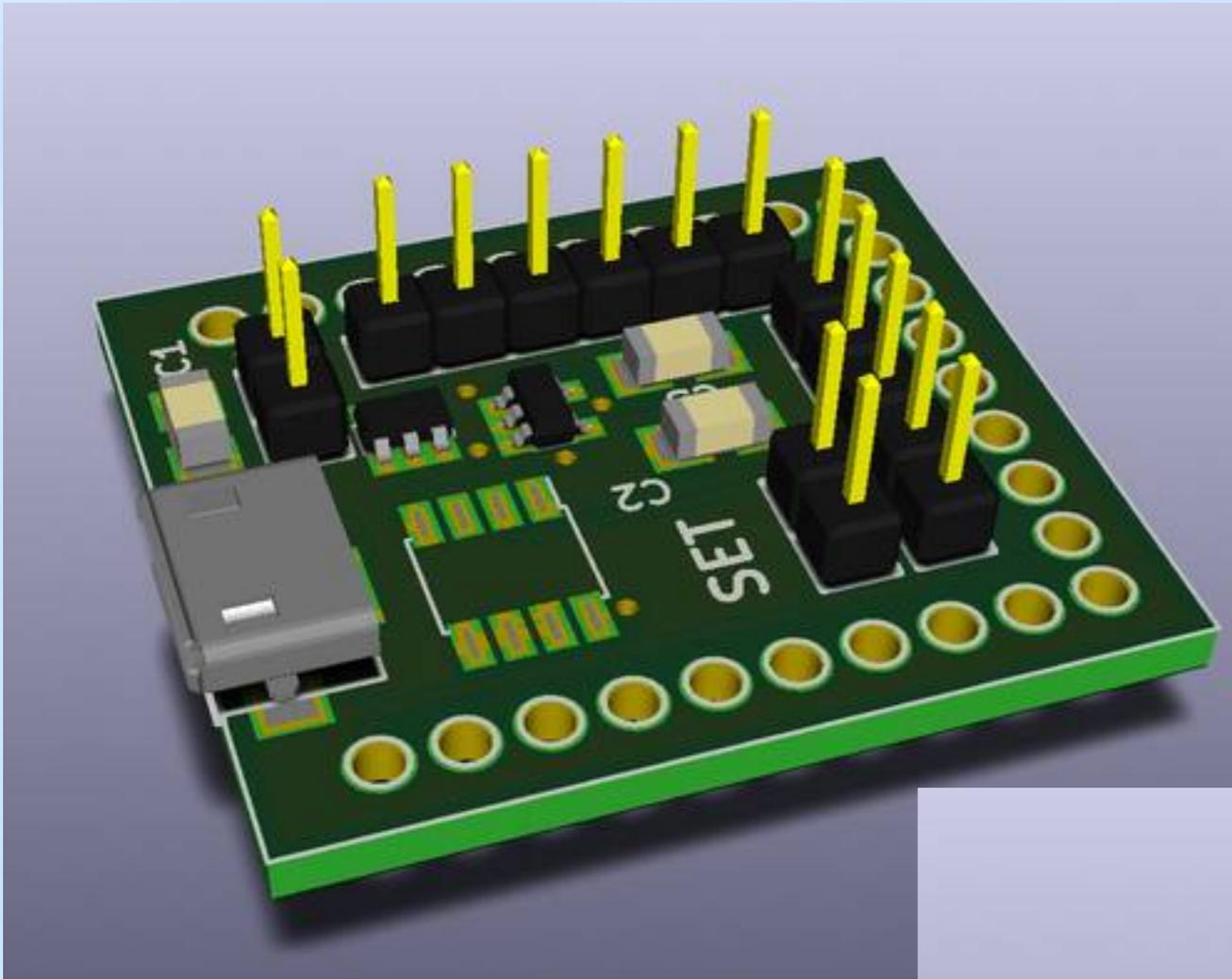




KiCad

- KiCad è una suite open source di software Electronic Design Automation (EDA) per il disegno di schemi elettrici e circuiti stampati (PCB). Ha un ambiente di sviluppo integrato (IDE) con editor di schematici, generazione della distinta base, sbroglio circuitale del PCB e visualizzatore di file Gerber.



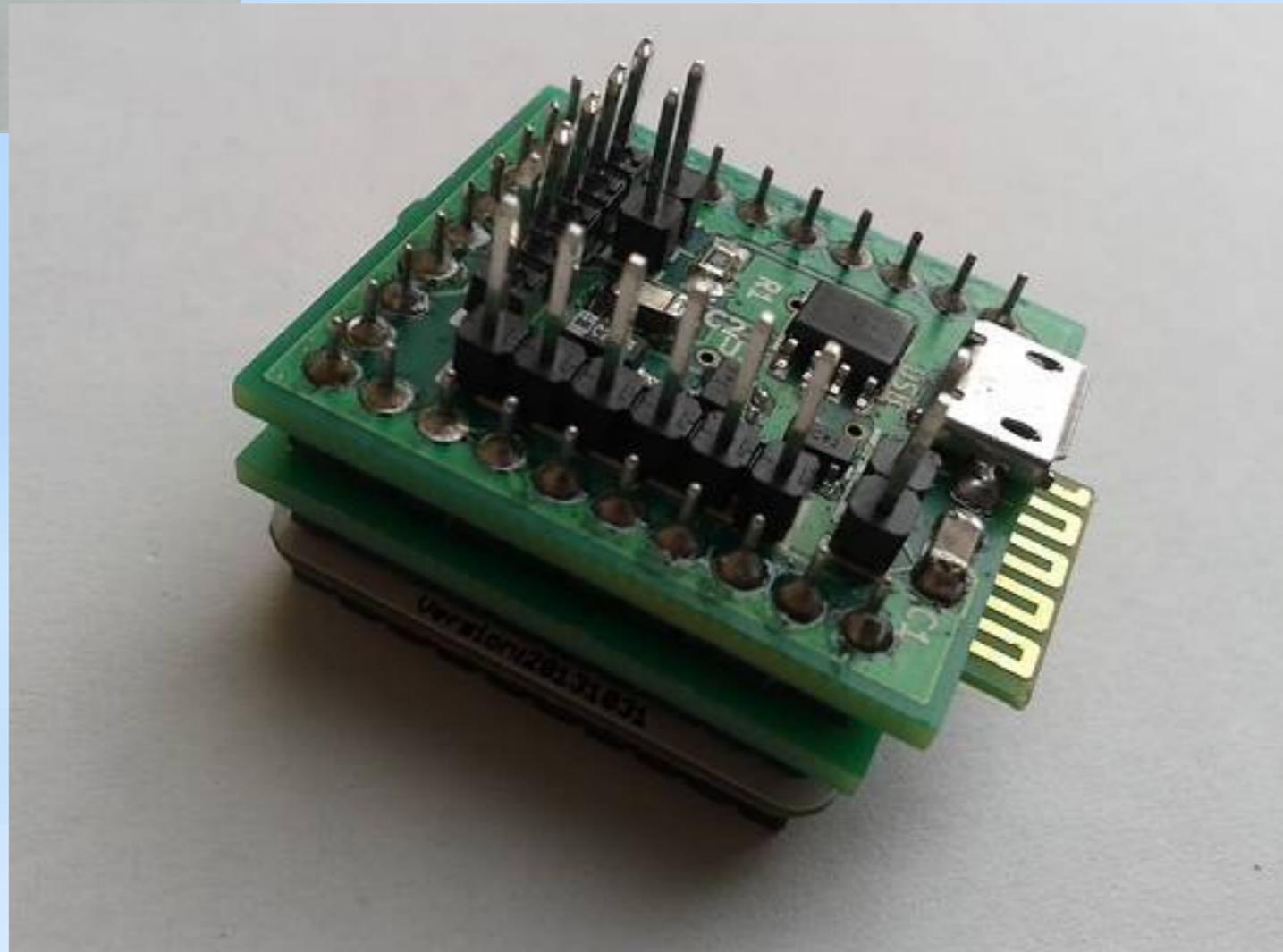


Reggio Emilia, 2015-10-24

Patruno, *Che Stima per chi Stima !* Rete Monitoraggio Ambientale Partecipativo

Raspibo per progetto RMAP - [HTTP://rmap.cc](http://rmap.cc)



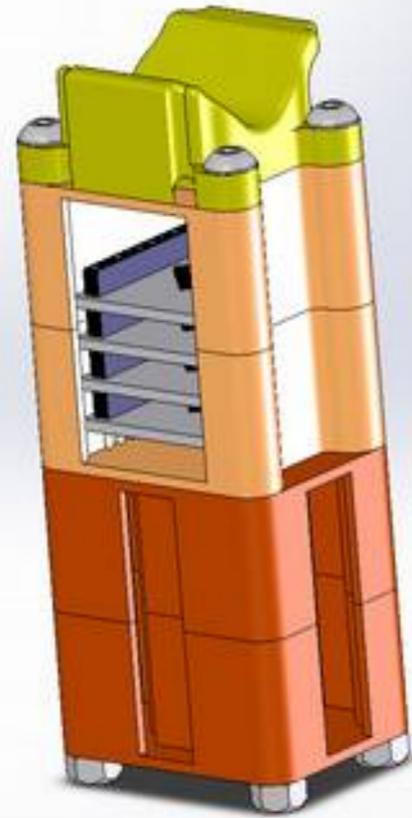


Reggio Emilia, 2015-10-24

Raspibo per progetto RMAP - [HTTP://rmap.cc](http://rmap.cc)

Patruno, Che Stima per chi Stima ! Rete Monitoraggio Ambientale Partecipativo





Reggio Emilia, 2015-10-24

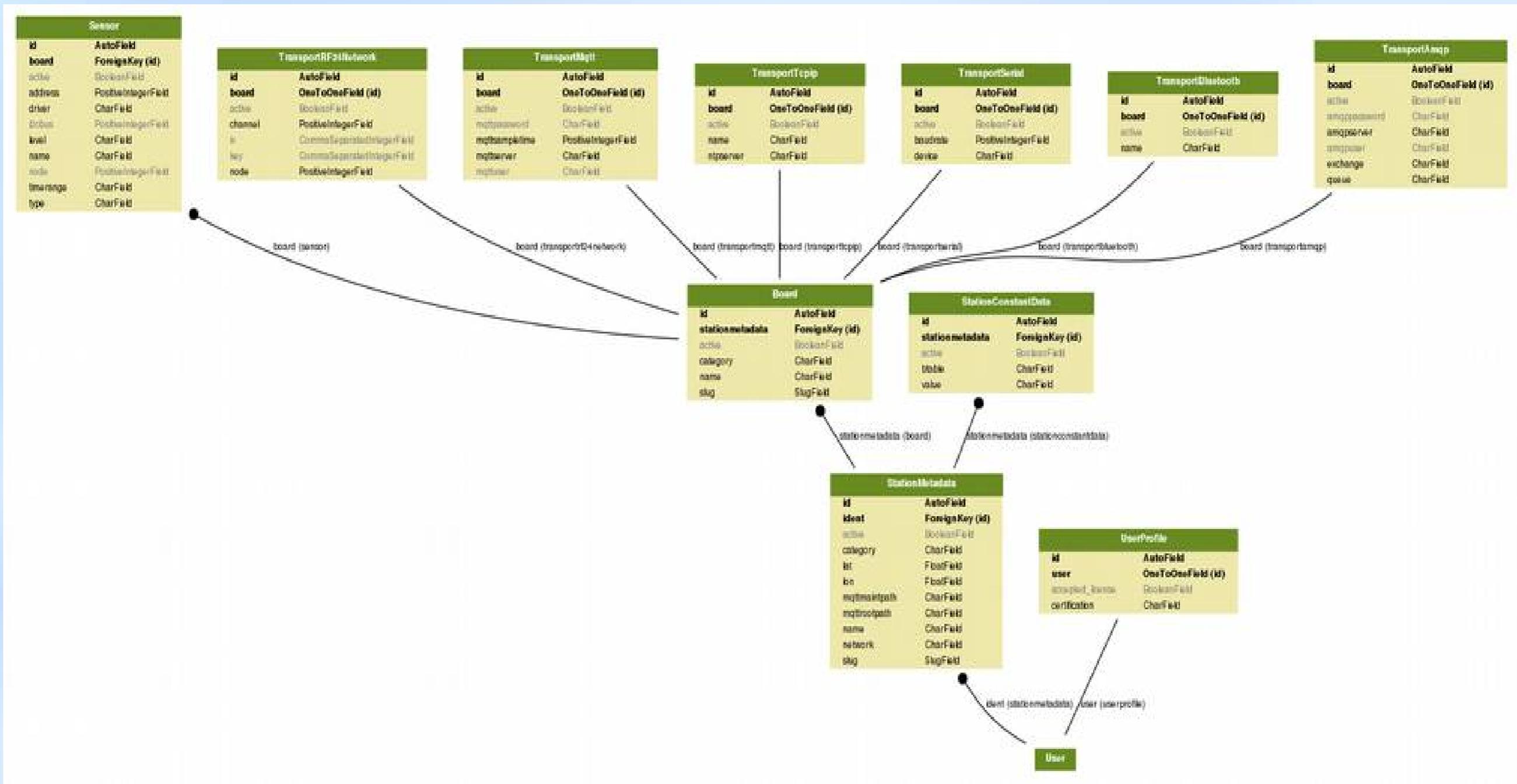
Patruno, Che Stima per chi Stima ! Rete Monitoraggio Ambientale Partecipativo

RaspiBo per progetto RMAP - [HTTP://rmap.cc](http://rmap.cc)





Tabelle e relazioni dei metadati di stazione





L'applicazione RMAP

- Questo programma interagisce con l'utente tramite interfaccia grafica e permette la pubblicazione di dati ambientali sul server rmap.cc. I dati possono essere rilevati manualmente e a vista oppure con i moduli STIMA

L'APP guida attraverso una sequenza di fasi:

- identificazione della propria posizione
- inserimento dei dati manuali
- connessione e attivazione dei dispositivi Stima per il rilevamento automatico dei dati
- connessione e invio dei dati al server

L'APP ha due modalità di funzionamento: una interattiva e una in background.





Tecnologie utilizzate nell'applicazione RMAP

- **Django**

Django è un web framework per lo sviluppo di applicazioni web, scritto in linguaggio Python. Fornisce funzionalità che facilitano lo sviluppo rapido di applicazioni per la gestione di contenuti.

Principali funzionalità:

- Astrazione del **database** relazionale ad oggetti
- Sistema di **template** basato su tag con ereditarietà dei template
- Supporto per **localizzazione**
- Sistema di gestione degli **utenti** e loro **autenticazione** nell'applicazione Web
- Sistema per la creazione e la validazione di **form** HTML





Kivy

Kivy è una libreria Python per lo sviluppo di applicazioni su device mobili e/o multi-touch con una interfaccia utente naturale (NUI). Funziona su Android, iOS, Linux, OS X, e Windows. Distribuito con la licenza MIT, Kivy è software libero.

Cellulari e tablet hanno portato con sé un cambiamento drammatico nell'uso delle applicazioni. La compatibilità è diventata essenziale e ha aumentato il tipo di interazione che gli utenti si aspettano : gesti , multi-touch , animazioni e penne magiche. Kivy è una soluzione open source Python che copre queste esigenze di mercato con un approccio di sviluppo facile da imparare e rapida . Kivy sta crescendo rapidamente e guadagnando attenzione come alternativa alle piattaforme di sviluppo standard.

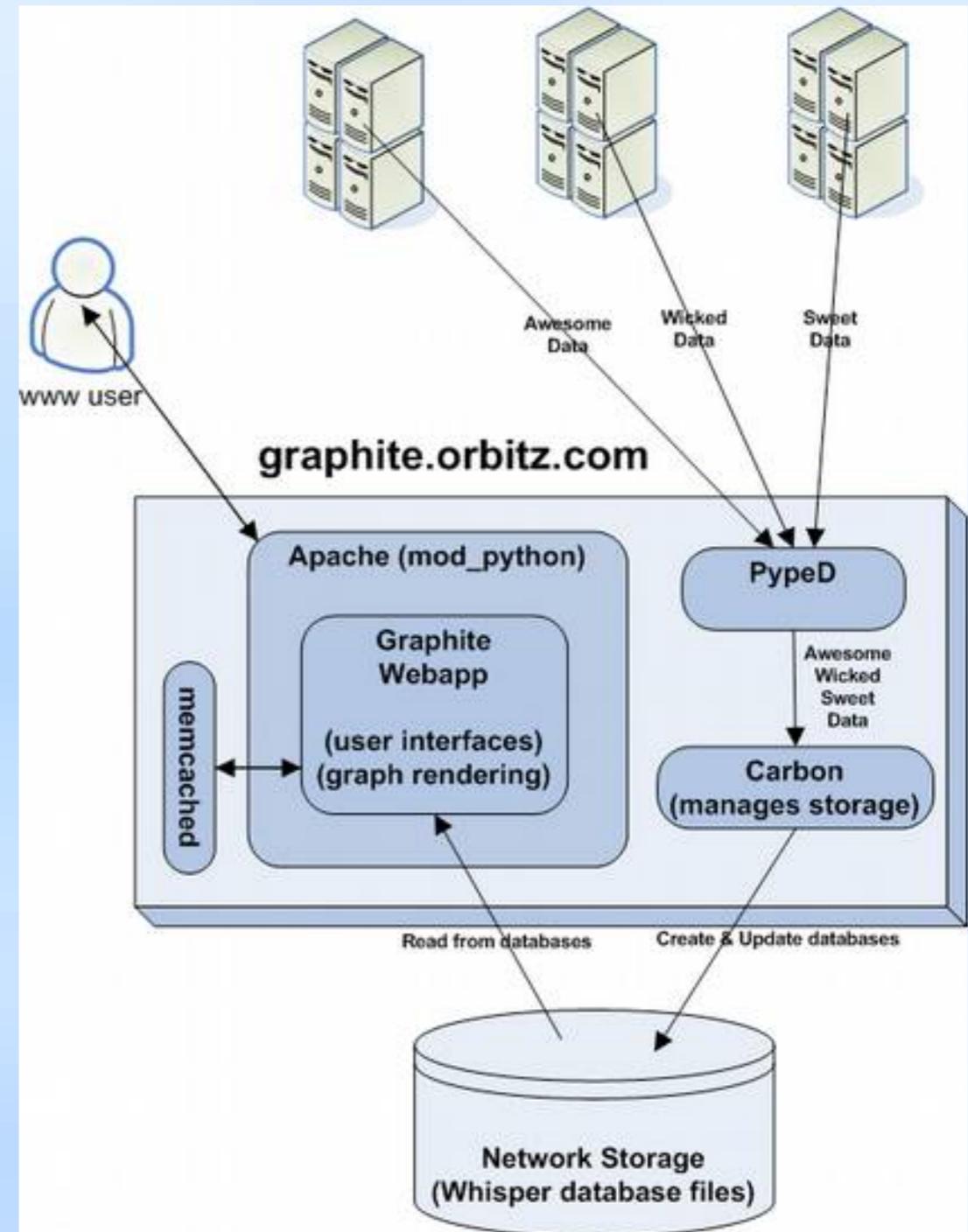


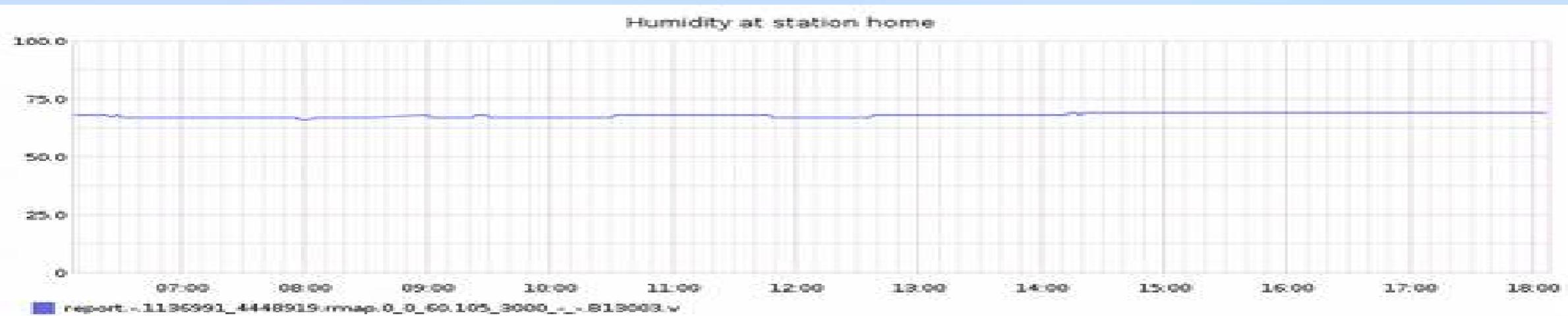
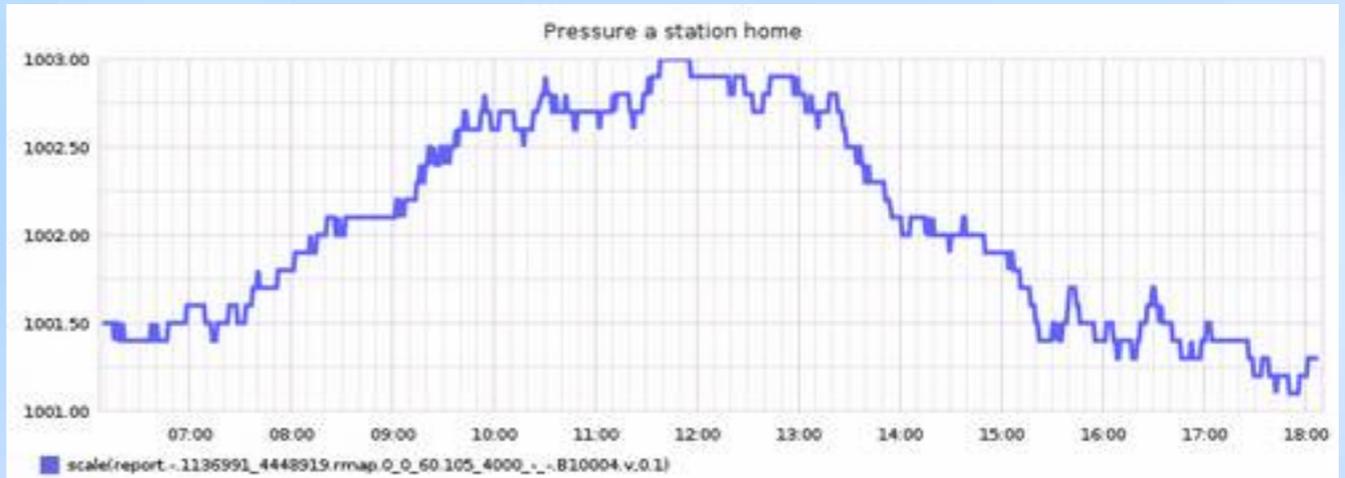
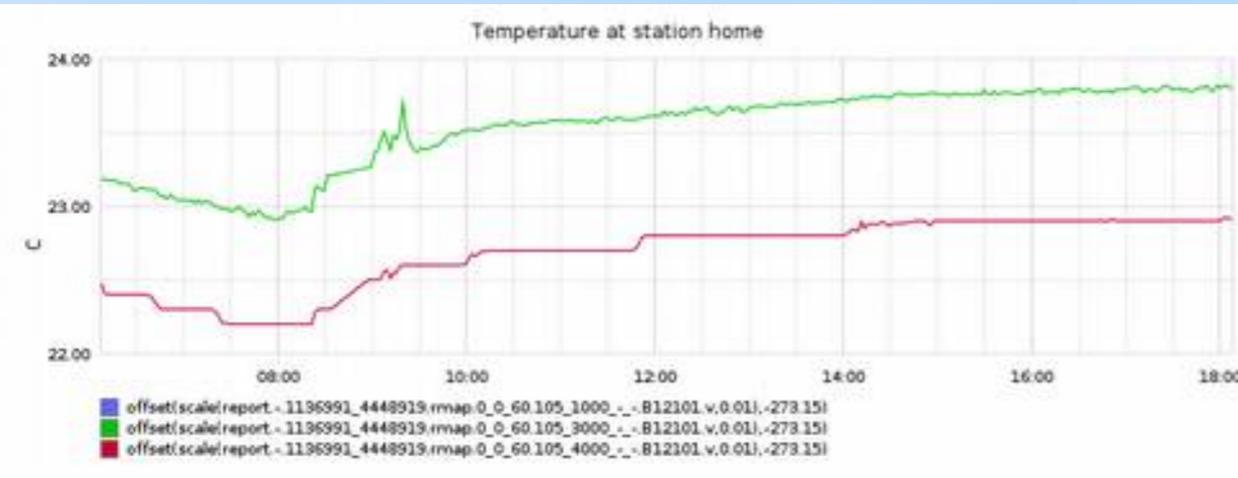
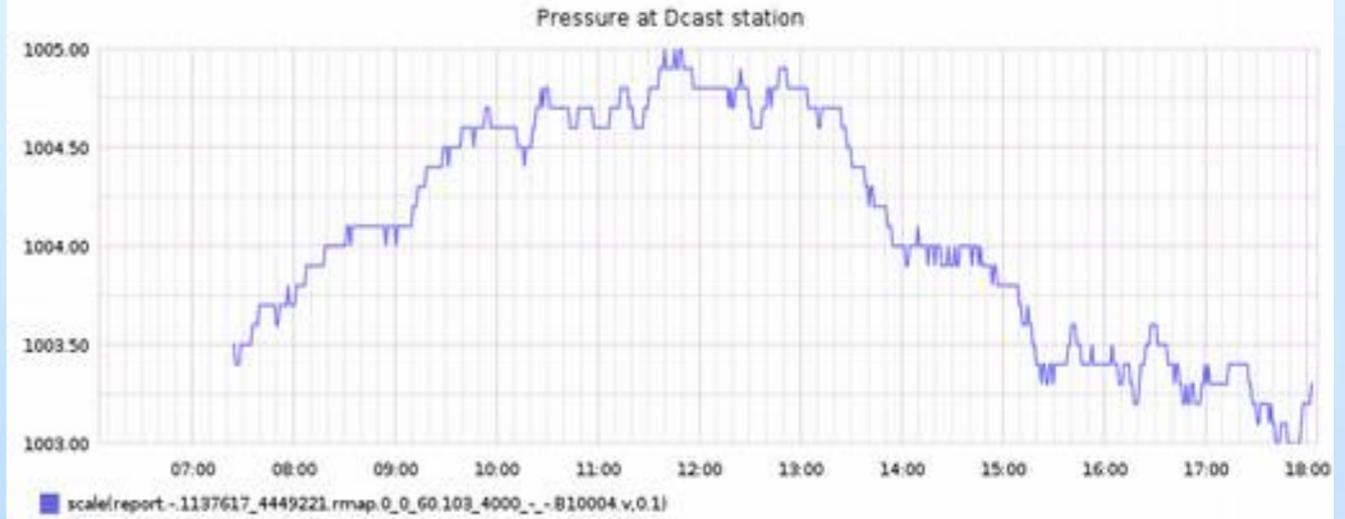
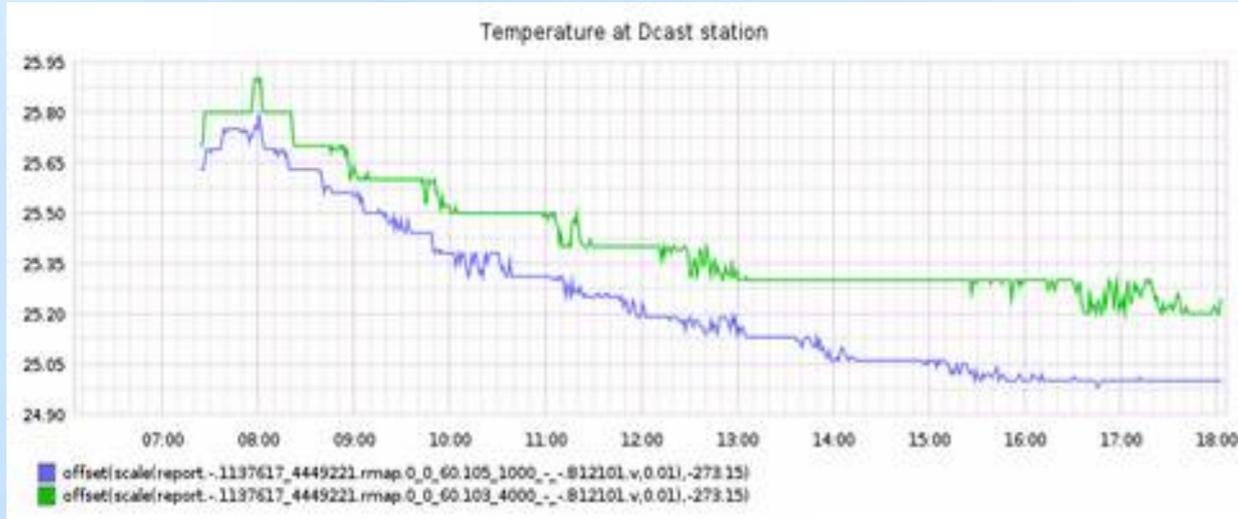


Graphite - Scalable Realtime Graphing

What is Graphite?

Graphite is a highly scalable real-time graphing system. As a user, you write an application that collects numeric time-series data that you are interested in graphing, and send it to Graphite's processing backend, carbon, which stores the data in Graphite's specialized database. The data can then be visualized through graphite's web interfaces.



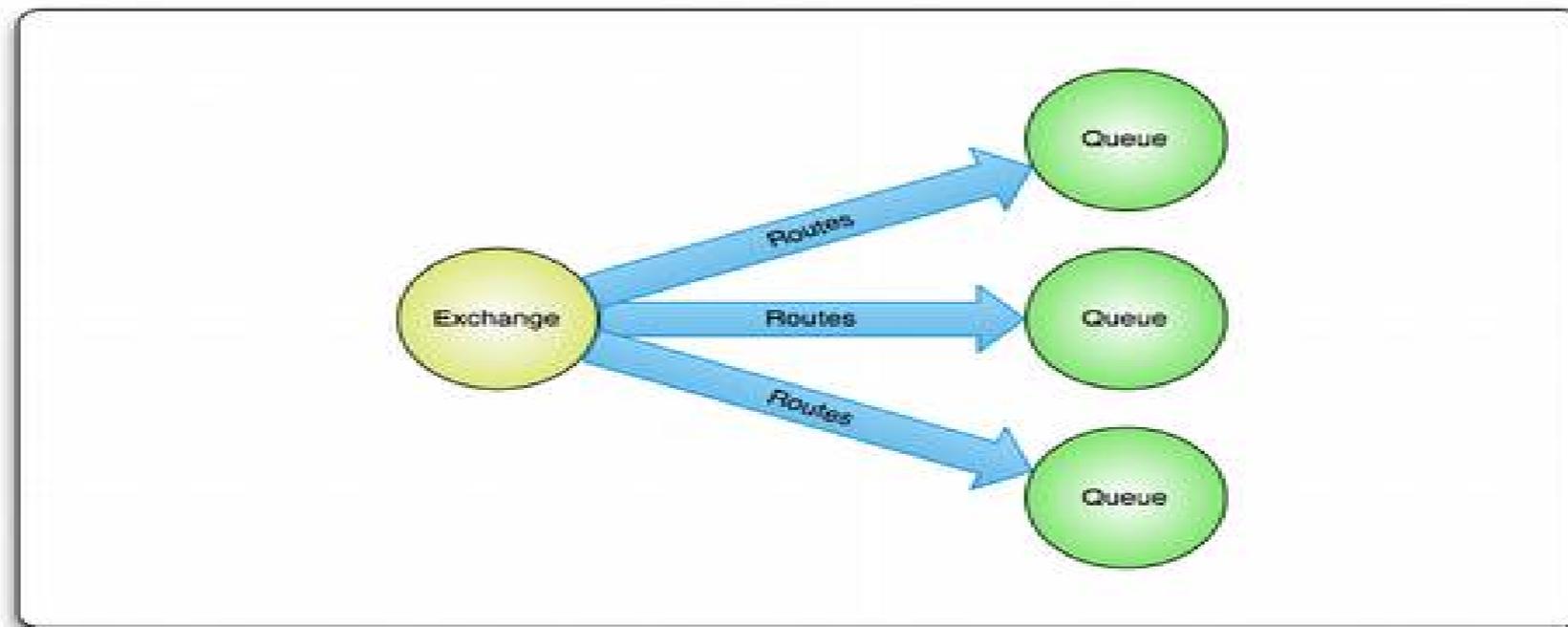




Rabbitmq

- message broker software that implements the Advanced Message Queuing Protocol (**AMQP**).
- **Exchanges** are AMQP entities where messages are sent.
- **Queue** store messages that are consumed by applications
- A **fanout exchange** routes messages to all of the queues that are bound to it

Fanout exchange routing





Upload messages

- **Shovel plugin**

- The high level goal of a shovel is to reliably and continually take messages from a queue (a source) in one broker and publish them to exchanges in another broker (a destination).
- The primary advantages of a shovel are:
 - Loose coupling
 - A shovel can move messages between brokers (or clusters) in different administrative domains:
 - they may have different users and virtual hosts;
 - they may run on different versions of RabbitMQ and Erlang.
 - WAN-friendly
 - The Shovel plugin uses AMQP to communicate between brokers, and is designed to tolerate intermittent connectivity without message loss.
 - Highly tailorable
 - When a shovel connects (either to the source or the destination) it can be configured to perform any number of explicit methods. For example, the source queue need not exist initially, and can be declared on connect.





DB-All.e

- tool per gestione dati puntuali meteorologici basato sulla loro rappresentazione fisica
- sviluppato dal SIMC, utilizzato per verifica modelli, applicazione operativa filtro di Kalman, casi studio
- Corredato di Provami, sofisticato programma interattivo per la:
 - Visualizzazione
 - Ricerca
 - Modifica
 - Esportazione





DB-All.e

DB-All.e is a fast on-disk database where meteorological observed and forecast data can be stored, searched, retrieved and updated. Many modern meteorological applications, like data assimilation, quality control and verification, need complex procedures for input of meteorological data, such as decoding and standardisation, and their organisation in memory. The need to manage a high number of measurement points and the need of long spans of space and time dimensions lead to a disproportionate use of RAM and increase the complexity of programming.

This framework allows to manage large amounts of data using its simple Application Program Interface (API), and provides tools to visualise, import and export in the standard formats BUFR and CREX.





DB-all.e

- **Fortran (77,2003), C, C++** and **Python** API are provided.
- Developed using **ODBC** programming layer
- To make computation easier, data is stored as **physical** quantities, that is, as values of a variable in a specific point of space and time, rather than as a sequence of encoded reports.
- Representation is in 7 dimensions: **observation network, x, y, z, datetime, timerange, variable**, where x,y are geographic coordinate, z table driven vertical coordinate, datetime the reference time, timerange table driven observation and forecast specification, variable table driven unique definition.
- Any data may have **attribute**, containing more information linked to the data.
- Unlimited station information data are possible.
- **Real, integer** and **character** data type are supported.
- It is **fast** for both read and write access.





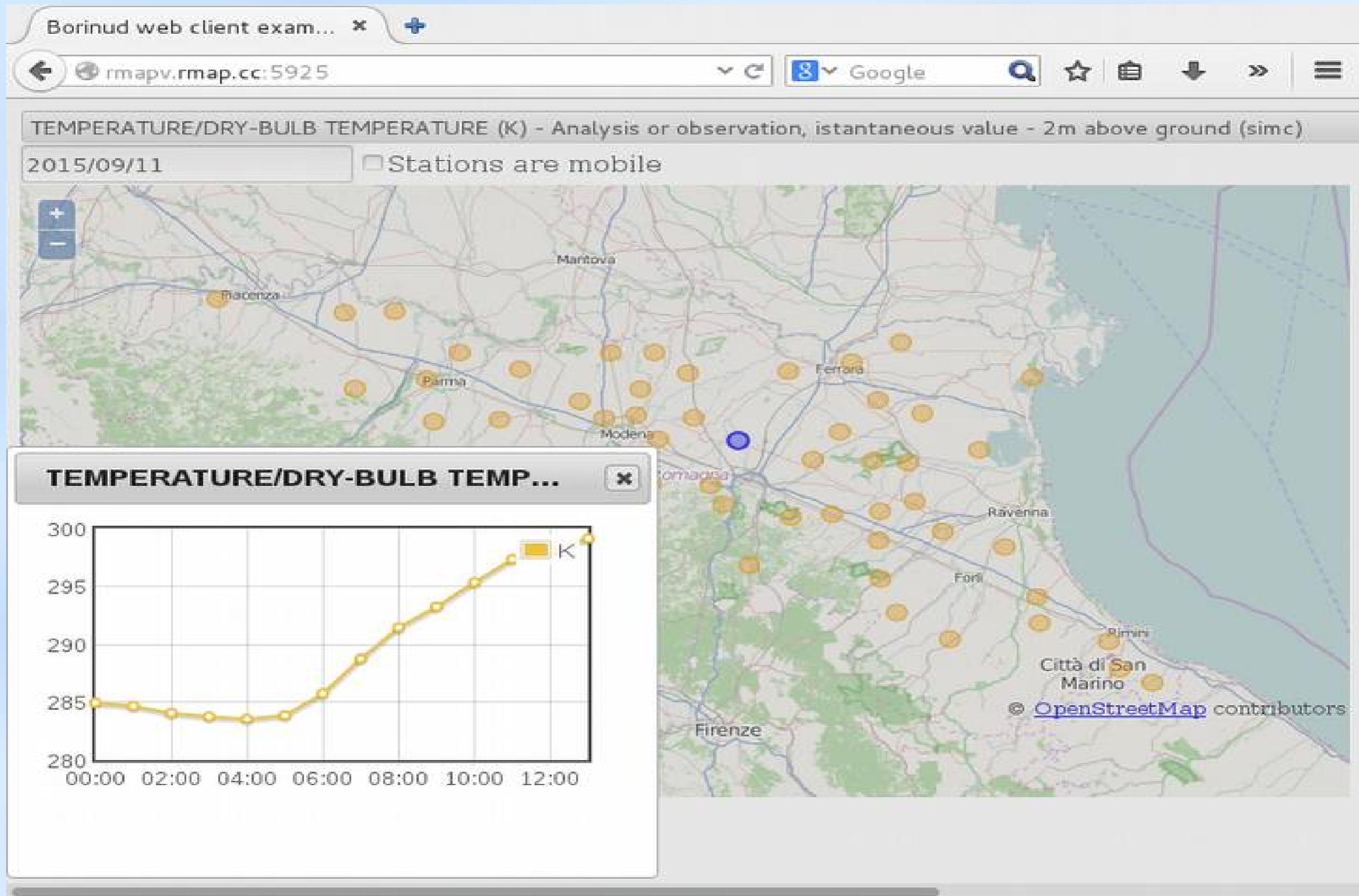
DB-all.e

- It is based on physical principles, that is, the data it contains are defined in terms of **homogeneous** and **consistent** physical data. For example, it is impossible for two incompatible values to exist in the same point in space and time.
- It can manage **fixed stations** and **moving stations** such as **airplanes** or **ships**.
- It can manage both **observational** and **forecast** data.
- It can manage data along **all three dimensions in space**, such as data from soundings and airplanes.
- Report information is preserved. It can work based on physical parameters or on report types.





Borinud: DB-All.e web and web service





The Arkimet archiving system

- Set di tool per archiviazione e distribuzione di dati ambientali
- Accesso locale (filesystem), remoto (HTTP) omogeneo tramite CLI
- Integrità dei dati: sono trattati come una stringa binaria opaca, in sola lettura e mai modificata
- E' possibile estendere facilmente i formati supportati (attualmente GRIB, BUFR, ODIMH5)
- Deploy molto semplice e veloce
- Arkiweb: interfaccia web <http://www.smr.arpa.emr.it/arkiweb>
- Sviluppato dal SIMC Licenza GPLv2+
- <http://svn.smr.arpa.emr.it/arkimet/arkimet/trunk>





Arkimet: come lavora

- Data is examined and metadata are extracted
- Data and metadata are acquired into datasets
- Datasets are self-contained collections of homogeneous information
- Datasets store data, metadata, and also summaries of the data
- A summary can be used to explore the contents of a dataset, or the output of a query, without extracting the data





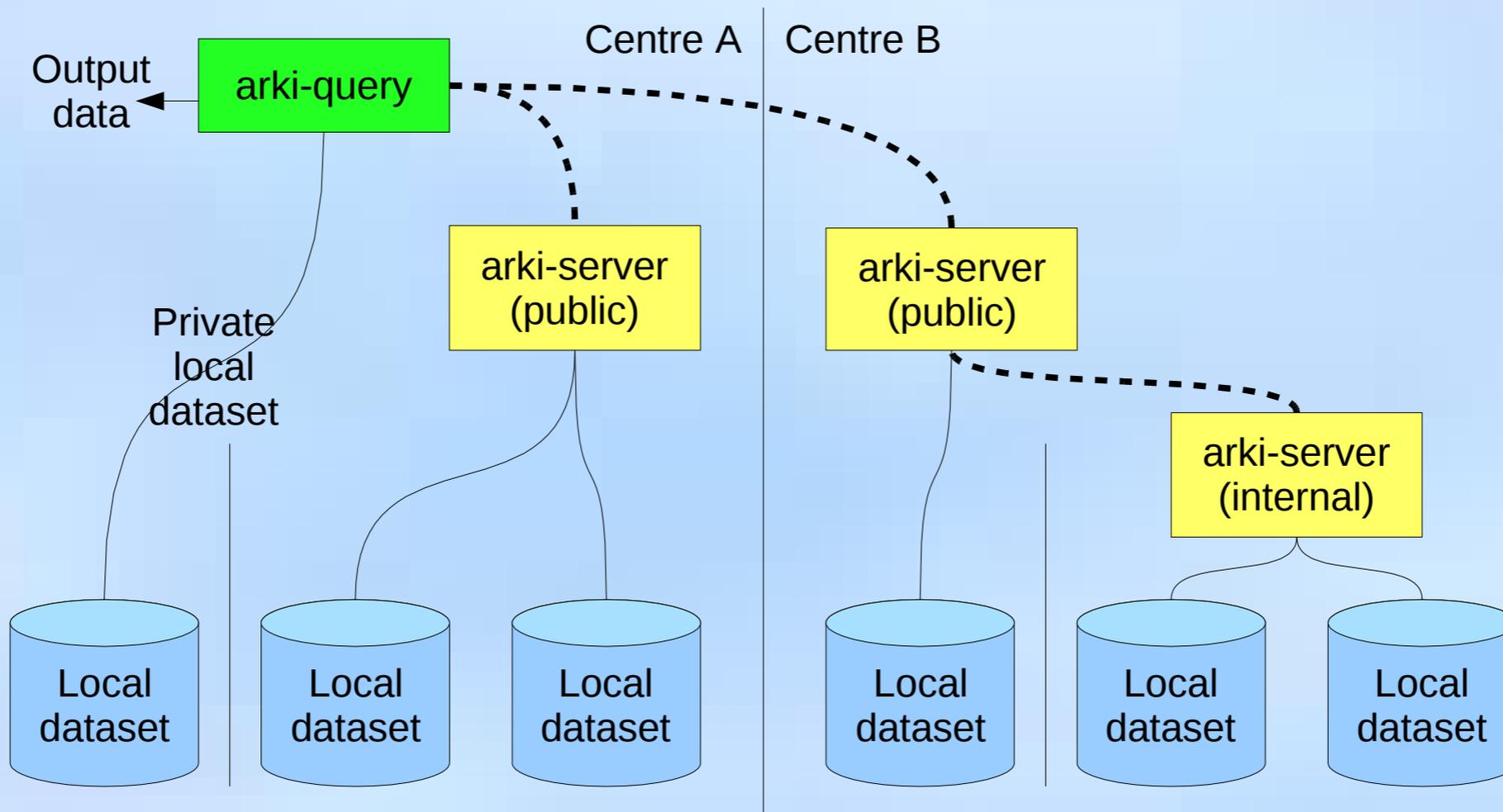
Arkimet: distributed

- Data is accessible locally and remotely, in the same way
- Remote access uses a client-server model, over standard HTTP
- Any centre / unit can deploy their own
- Server can share local and remote datasets
 - It is possible to create a public front-end server that aggregates several internal servers





Arkimet: distributed





Arkiweb

http://rmapv....cc/arkiweb/

rmapv.rmap.cc/arkiweb/

Più visitati Downloads Software bfsf Autolesionista Fortran Wiki comodino.org

back to datasets selection clear selection load summary

count: 30840 size: 4479042 from: 1970-01-01 00:04:00 until: 2014-10-14 16:51:00

query:

- reftime
- area
- origin

Copyright © 2011 ARPA-SIMC Emilia-Romagna - released under GNU General Public License version 0.15





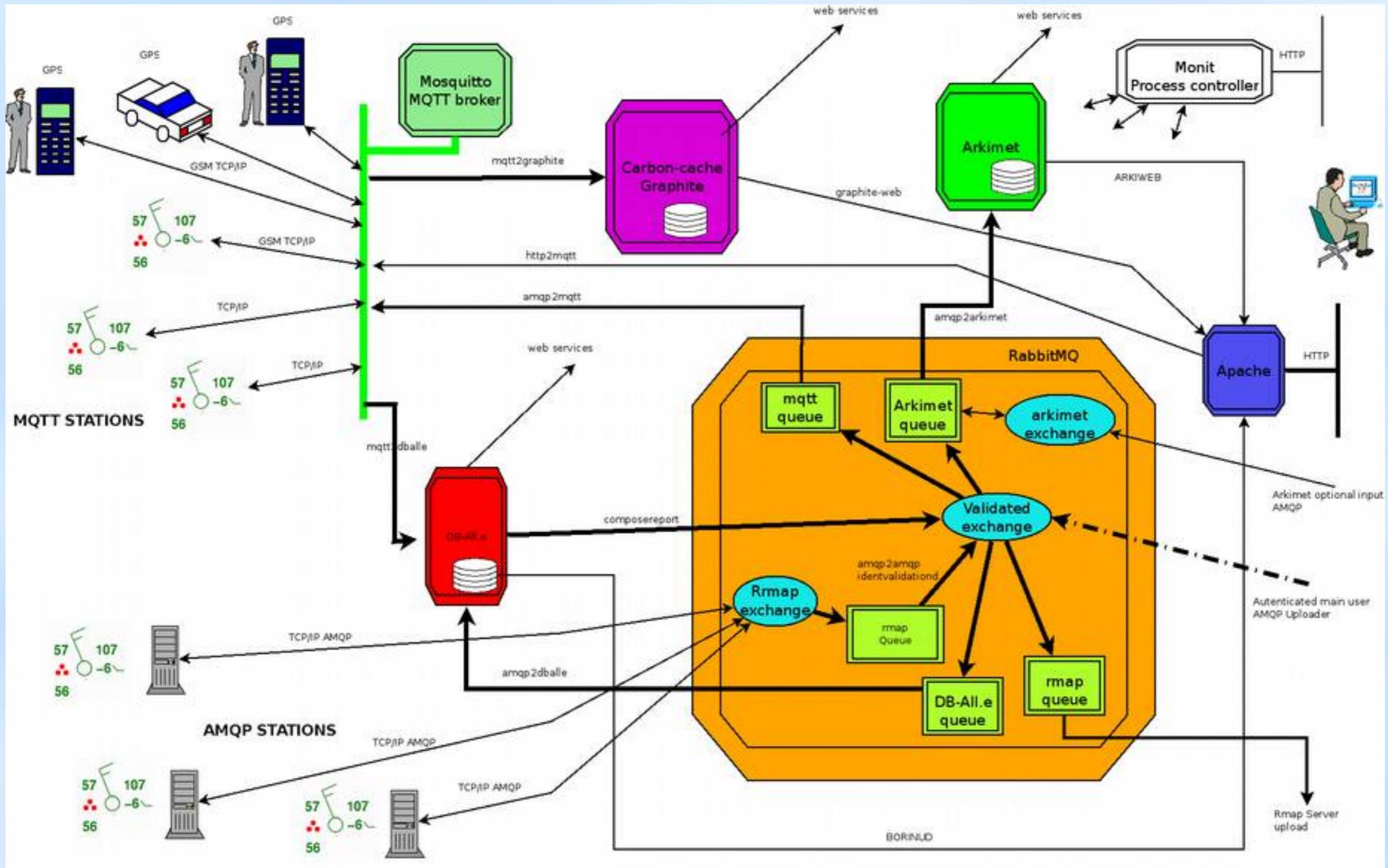
Libsim

- Libsim comprende quattro gruppi di moduli di utilità in Fortran 90:
 - libsim_base definisce moduli e classi di uso generale in applicazioni scientifiche, come la gestione di errori in esecuzione, la gestione di dati georeferenziati, di coordinate temporali, ecc.
 - libsim_grib definisce una serie di classi ad alto livello stratificate sopra la libreria ECMWF grib_api per gestire l'I/O di file in formato grib.
 - libsim_vol7d definisce una serie di classi per facilitare l'elaborazione di dati osservativi idro-meteo, includendo metodi per la loro importazione da database tipo DbAll-e
 - libsim_volgrid6d definisce una serie di classi per facilitare l'elaborazione di dati idro-meteo su grigliati georeferenziati, compresa la trasformazione in griglie di tipo diverso e in oggetti di tipo vol7d.





Flusso dati e processi





Monit

Monit is running on localhost with *uptime, 21d 21h 0m* and monitoring:

System	Status	Load	CPU	Memory	Swap
system_localhost	Running	[0.00] [0.07] [0.08]	0.2%us, 0.7%sy, 0.0%wa	44.5% [458040 kB]	6.2% [133552 kB]

Process	Status	Uptime	CPU Total	Memory Total
amqp2arkimemd	Running	21d 20h 59m	0.0%	0.2% [2828 kB]
amqp2mqtt	Running	21d 20h 59m	0.0%	0.2% [2840 kB]
mqtt2graphited	Running	21d 20h 59m	0.0%	0.2% [2804 kB]
amqp2dballed	Running	21d 20h 59m	0.0%	0.2% [2788 kB]
borinudd	Running	14d 6h 35m	0.0%	1.2% [12656 kB]
mqtt2dballed	Running	18d 4h 43m	0.0%	0.2% [2936 kB]
composereportd	Running	19d 2h 26m	0.0%	0.1% [1532 kB]

Copyright © 2000-2011 [Tikdeslash](#). All rights reserved. [Monit web site](#) | [Monit Wiki](#) | [M/Monit](#)





STIMA software

- <http://rmap.cc>
- <http://liste.raspibo.org/wws/subscribe/meteo>
- <https://github.com/r-map>

