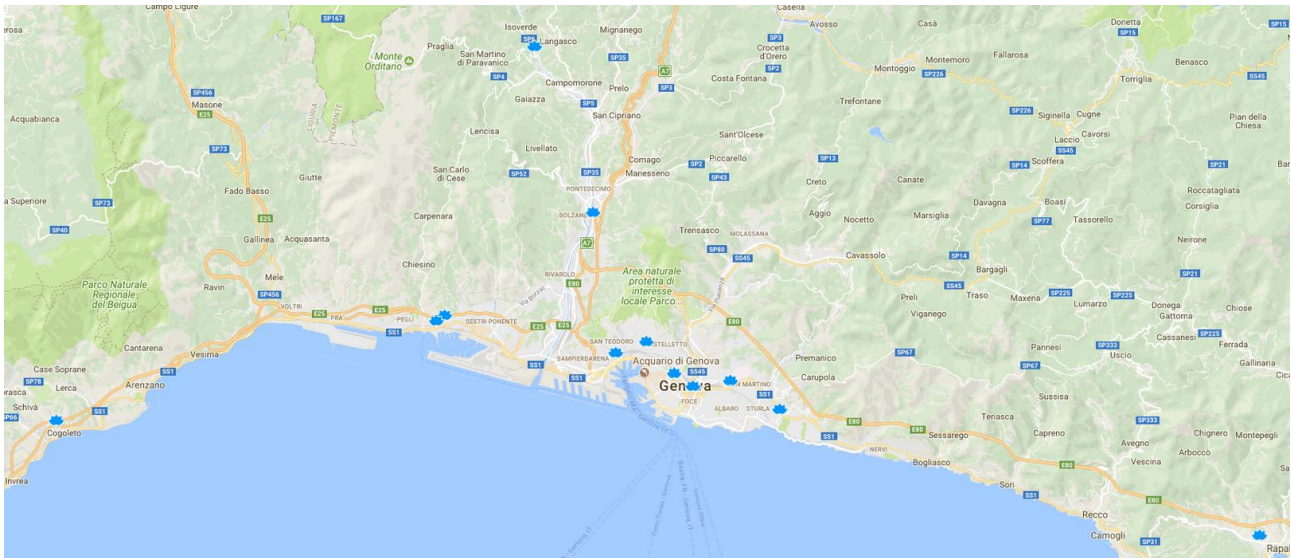


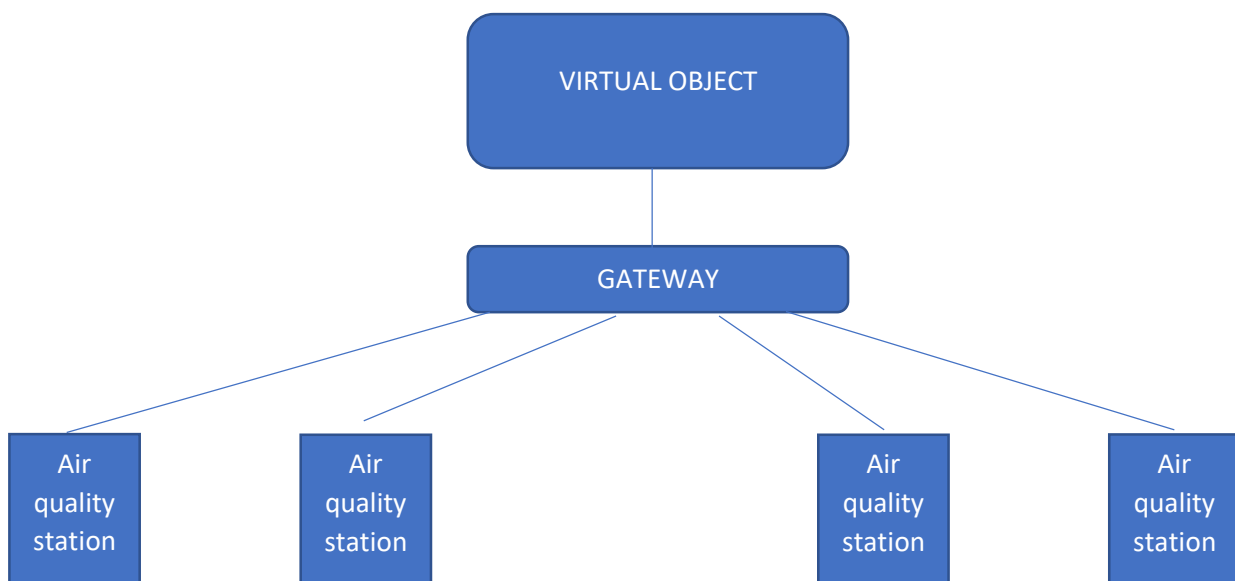
Air quality stations VO – Specifiche

Il virtual object “Air quality stations vo” salva ed elabora i dati provenienti dalle air quality stations installate sul territorio per rilevare la qualità dell’aria in macroaree della città.

Oggetti reali



Le air quality stations che invieranno i dati al virtual object saranno quelle installate nel territorio del comune di Genova. Le air quality stations registrano nuovi dati ogni ora. Tutte i dati delle air quality stations sono accessibili direttamente tramite un servizio REST (GET) al link: http://www.iononrischiocloud.comune.genova.it/back_end/airsensors.php. Tutte le air quality stations sono interrogate da un gateway che si occuperà di inviare i dati al virtual object tramite una POST REST.



Gateway

Il gateway è un programma Java esterno con un loop sempre attivo che si occupa ogni ora di:

- Interrogare il servizio http://www.iononrischioclout.comune.genova.it/back_end/airsensors.php per ricevere tutti i dati di tutte le air quality stations. Riceverà un JSON di questo tipo:

```
[
  {
    "id": "Quarto",
    "name": "Quarto Se.Di.",
    "description": "Background - urbana, parchi, imp. Sportivi",
    "latitude": "44.395095",
    "longitude": "8.991577",
    "attributes": [
      Vari attributi.... Sceglierne solo ozono e NO2.
    ],
    "last_update": "1 febbraio 2018"
  },
  continua ....
]
```

- Inviare al VO il json con una POST al servizio new data del VO inserendo tutti i dati (insieme ad attributi NO2 e ozono) in un JSON. Il formato del JSON si può recuperare effettuando una GET al servizio getInfo del virtual object.

Virtual object

Inizializzazione.

Caricare nel servlet context le properties presenti nel file di properties. In particolare URL VORegister e URL Orchestratore.

Inizializzare il logger su un rolling file appender.

Inizializzare il VO inviando al VORegister:

- Nome
- Indirizzo IP/URL su cui è in ascolto la invoke action
- Hashmap azioni invocabili-descrizione
- Hashmap eventi-descrizione
- Proprietà

Elenco azioni invocabili su VO

Nessuna.

Elenco eventi inviabili da VO

CRITICAL_OZONE_LEVEL

CRITICAL_NO2_LEVEL

Modulo comunicazione real object – virtual object.

CLASSE VORealObjectCommunication, classe annotata JAX-RS che contiene:

- FUNZIONE NEW DATA: Espone funzione annotata JAX-RS con metodo POST 'newData' chiamabile dalle oggetti reali.

esempio

POST:

http://localhost:8080/airqualitystationsVO/rest/VORealObjectCommunication/newData

- FUNZIONE GET INFO: Espone funzione annotata JAX-RS con metodo GET 'getInfo' chiamabile dalle oggetti reali per ricavare informazioni sull'oggetto virtuale (formato dati inviabile su new data)

Modulo comunicazione Spreadsheet Space.

Sviluppare lo Spreadsheet Space connection module nel virtual object..

- Alla ricezione di un nuovo messaggio 'newData' il virtual object istanzia il thread SSSConnection che chiama il metodo updateSSSDData(String payload). Dove il payload è il JSON ricevuto dal client.
- updateSSSDData legge il payload inserendo i dati in una struttura dati locale e legge l'id della air quality station.
- Se è la prima volta che riceve i dati dell'air quality station crea una nuova view chiamata (air_quality_station_data_air_quality_station_id) dove air_quality_station_id è l'ID dell'air quality station contenente un range di tipo 'table' (a dimensione variabile). La view deve essere esposta al proprio account e all'account 'giancarlo.camera@spreadsheetspace.net'.
- Se è già stata creata una view per quell'air quality station bisogna aggiornare la tabella contenente i dati di quell'air quality station.
- I dati da inserire in ogni riga sono tutti i dati ricevuti con il Json eccetto id, latitude e longitude.
- Il metodo ritorna un messaggio di successo in caso di buona riuscita dell'operazione o di errore in caso di problemi. Il messaggio deve essere loggato sul file di log impostato.

Modulo core.

CLASSE onReceiveNewData

Estende classe **thread*** e contiene:

- FUNZIONE per il processing del livello d'ozono ricevuto. Se supera il valore indicato come valore massimo all'interno del JSON invoca la notify event passando l'evento CRITICAL_OZONE_LEVEL.
- FUNZIONE per il processing del livello di NO2 ricevuto. Se supera il valore indicato come valore massimo all'interno del JSON invoca la notify event passando l'evento CRITICAL_NO2_LEVEL.
- FUNZIONE notifyEvent () : Client REST che chiama metodo POST notify event dell'orchestratore per notificare una particolare condizione verificatasi sul virtual object (CRITICAL_OZONE_LEVEL / CRITICAL_NO2_LEVEL)

Modulo comunicazione virtual object – orchestratore.

Nessuno