Le OGC API per promuovere gli Obiettivi di Sviluppo Sostenibile 2030*

Nunziata Vincenzo[1]

[1] Università degli Studi di Salerno, v.nunziata16@studenti.unisa.it

Abstract. Le tecnologie maggiormente utilizzate ancora oggi per la distribuzione su Internet dei dati territoriali fanno uso di standard e protocolli che, sebbene fossero lo stato dell'arte quando tali tecnologie sono state sviluppate, mostrano i loro limiti nel venire incontro alle sfide tecnologiche del futuro. Il presente articolo descrive l'utilizzo dei nuovi standard proposti dall'Open Geospatial Consortium per realizzare un approccio moderno per la pubblicazione online dei dati spaziali.

Parole chiave: SDI, OGC, API, dataset territoriali, web

1 Introduzione

Sin dalla fine degli anni 90 è nata l'esigenza di definire modelli e protocolli per la pubblicazione, l'accesso e l'interoperabilità dei dati territoriali. Durante il corso degli anni, enti governativi e amministrazioni locali hanno raccolto grandi quantità di informazioni geografiche, avendo poi la necessità di dover organizzare e distribuire pubblicamente tali dati.

Per rispondere a questa esigenza sono nate le *Spatial Data Infrastructures* (SDI), ovvero un insieme di politiche, tecnologie, standard e risorse umane necessarie per acquisire, elaborare, archiviare, distribuire dati geospaziali. Le SDI offrono una piattaforma unificata per l'accesso ai dati territoriali, incentivandone il riuso e risolvendo il problema della ridondanza ("publish once, reuse everywhere"). Inoltre, incrementano la trasparenza degli enti governativi e stimolano la crescita di business che non sarebbero possibili altrimenti [1].

Anche a livello europeo, la direttiva INSPIRE (*Infrastructure for Spatial Information in the European Community*), pubblicata nel 2007, ha fissato le linee guida per la realizzazione di SDI per facilitare la condivisione dei dati territoriali e l'interoperabilità dei servizi tra gli Stati della Comunità Europea. Tale direttiva stabilisce politiche, protocolli, standard e tecnologie per la pubblicazione e l'integrazione a livello europeo dei dati geospaziali provenienti da enti governativi e amministrazioni locali.

Un problema legato alle SDI è che le politiche e gli standard adottati devono restare costantemente al passo con il rapido sviluppo delle tecnologie informatiche. Ad

* Supported by MIUR PRIN 2017 grant number 2017JMHK4F 004

esempio, gli enormi progressi che recentemente si sono avuti in ambiti come IoT, Intelligenza Artificiale e Big Data, hanno fatto emergere nuove necessità per la raccolta e la gestione di (Spatial) Big Data in vari domini, primi tra cui quelli legati alla Agenda 2030 dell'ONU con i suoi 17 obiettivi di Sviluppo Sostenibile. Ancora oggi, molte SDI sono implementate, data la loro età, seguendo protocolli e standard che, sebbene ancora largamente utilizzati nell'ambito GIS, non risultano adatti per venire incontro alle nuove sfide tecnologiche presenti e future.

Questo lavoro, in collaborazione con ISPRA (Istituto Superiore per la Protezione e la Ricerca Ambientale), propone quindi un nuovo approccio per la pubblicazione di dati spaziali online che, prendendo spunto dalle moderne best practice nell'ambito dello sviluppo web, permette un accesso rapido, semplice e standardizzato a grandi quantità di dati territoriali strutturati e non strutturati. Tale approccio agevola la collaborazione tra settori diversi per una migliore comprensione di scenari ambientali a sostegno dello Sviluppo Sostenibile.

2 Limiti dei Web Services attuali

Nel corso degli anni i dati sono diventati sempre più un elemento centrale nell'ambito tecnologico. Settori come l'Intelligenza Artificiale richiedono l'analisi di enormi quantità di dati per funzionare con accuratezza; sensori e altri dispositivi appartenenti al mondo dell'IoT producono in streaming milioni di dati al secondo. D'altro canto, motori di ricerca e di aggregazione permettono l'accesso ai dati a una vasta platea di "non addetti ai lavori". Ovviamente, anche i dati territoriali hanno subito la stessa sorte, basti pensare, ad esempio, a dati metereologici e ai dati di sensori ambientali.

Sin dalla fine degli anni 90, servizi come *Web Feature Service* (WFS) e *Web Coverage Service* (WCS) permettono di distribuire dati territoriali online mediante Web Services. In Figura 1 è presentato un esempio di dataset di features vettoriali (riguardante nodi ferroviari) pubblicato da un servizio WFS del Geoportale Nazionale [2].

La figura permette di evidenziale le principali problematiche legate al servizio WFS, analoghi problemi si riscontrano anche in servizi simili che utilizzano lo stesso protocollo (ad esempio, WCS).

WFS è uno standard ad-hoc, uno sviluppatore interessato a implementare una applicazione che utilizzi servizi WFS deve necessariamente conoscere tutta la specifica per poter comprendere come interrogare il servizio.

La verbosità del formato di codifica (GML) rende difficile la lettura dei dati e, di conseguenza, il loro utilizzo. Inoltre, ne appesantisce il trasferimento e l'elaborazione, rendendoli inutilizzabili in applicazioni che processano grandi quantità di informazioni. Per tanto, un servizio WFS non può essere utilizzato, ad esempio, da algoritmi di Machine Learning e da dispositivi IoT che generano enormi volumi di dati ogni secondo.

Infine, un servizio WFS non è indicizzabile dai motori di ricerca. Il suo riferimento deve essere pubblicato online (ad esempio, in un catalogo) per poter essere individuato dagli utenti. Per tanto, una persona che necessita di particolari dati non può "scoprire"

l'esistenza del servizio attraverso una ricerca sul web, ma deve conoscere i cataloghi dove effettuare la ricerca.

Queste problematiche rendono difficile l'utilizzo di tali servizi da parte di sviluppatori e utenti non esperti, frenandone la diffusione. Ancora oggi è raro trovare applicazioni di uso generale che scaricano dati spaziali interrogando servizi WFS o WCS, i quali restano limitati all'utilizzo in software e visualizzatori GIS specializzati.

```
<wfs:FeatureCollection xsi:schemaLocation="…">
    <gml:boundedBy>
        <gml:Envelope srsName="EPSG:4326">
            <gml:lowerCorner>36.731882 6.681638</gml:lowerCorner>
            <gml:upperCorner>47.006998 18.481041</gml:upperCorner>
        </gml:Envelope>
    </gml:boundedBy>
    <gml:featureMember>
        <ms:RT.RETEFERROVIARIA.NODOFERROVIARIO gml:id="…">
            <gml:boundedBy>
                <gml:Envelope srsName="EPSG:4326">
                    <gml:lowerCorner>42.893576 13.900488</gml:lowerCorner>
                    <gml:upperCorner>42.893576 13.900488</gml:upperCorner>
                </gml:Envelope>
            </gml:boundedBy>
            <ms:boundary>
                <gml:Point srsName="EPSG:4326">
                    <gml:pos>42.893576 13.900488</gml:pos>
                </gml:Point>
            </ms:boundary>
            <ms:reg>13</ms:reg>
            <ms:dec_tipo>Intersezione con confine regionale (competenza)</ms:dec_tipo>
            <ms:nome> </ms:nome>
        </ms:RT.RETEFERROVIARIA.NODOFERROVIARIO>
    </gml:featureMember>
```

Figura 1 Estratto di un dataset vettoriale pubblicato da un servizio WFS (in formato GML)

3 OGC API

Per offrire una soluzione a tali problemi, negli ultimi anni, l'Open Geospatial Consortium (OGC) ha pubblicato le *OGC API*, una nuova famiglia di standard che semplificano l'accesso online ai dati territoriali, definendo standard moderni per la distribuzione dei dataset spaziali che seguano le Web Best Practice per i dati sul web stabilite dal *World Wide Web Consortium* (W3C) [3].

Seguendo il principio dell'architettura Client-Server classica del Web, è possibile dividere l'architettura di una SDI che implementa le specifiche OGC API tra chi produce i dati (il Server) e chi li consuma (il Client).

Il Server raccoglie i dati spaziali dalle fonti nei formati e nei sistemi di riferimento più disparati (JSON, CSV, GeoPackage, geoTIFF, servizi WFS, ecc.) e ne permette l'accesso online in maniera uniforme e strutturata, attraverso API web ed endpoint RESTful. A ogni risorsa (dataset, metadati, ecc.) è assegnato un URI univoco che la identifica e che ne costituisce l'endpoint da contattare per l'accesso. Le specifiche

raccomandano che i dati siano pubblicati per lo meno in formato html e JSON (o, in questo caso, GeoJSON), e con sistema di riferimento WGS84 (sebbene permettano l'utilizzo di altri formati e sistemi di riferimento, opportunamente indicati nelle *Conformance Classes* alle quali il servizio aderisce). Inoltre, in aggiunta ai dataset, il Server deve pubblicare una *landing page* descrittiva, l'elenco delle *Conformance Classes* a cui il servizio aderisce (ovvero, le specifiche che il Server implementa) e una descrizione dettagliata delle API in formato *OpenAPI* [4].

Il Client accede ai dati pubblicati dal Server tramite gli endpoint RESTful, utilizzando i comuni protocolli di comunicazione HTTP. Ciò rende possibile l'accesso ai dati anche tramite browser Web, oltre che programmaticamente con i più diffusi linguaggi di programmazione.

La Figura 2 mostra schematicamente l'architettura Client-Server descritta nei paragrafi precedenti.

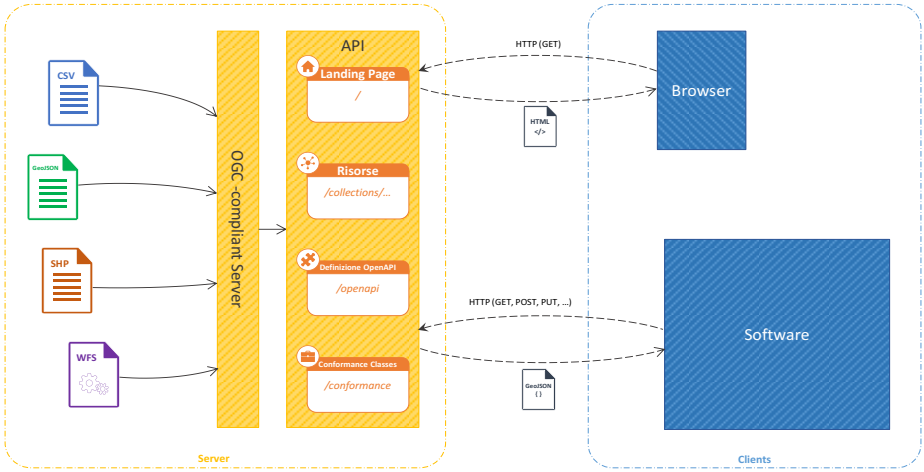

Figura 2 Architettura Client-Server, estesa alle OGC API

L'utilizzo di formati sia machine-readable che human-readable (come JSON e OpenAPI) rende fruibile il servizio sia da parte di sistemi automatizzati che da esseri umani. Inoltre, un utente qualsiasi, sebbene debba conoscere nozioni base nell'ambito GIS, può utilizzare il servizio senza dover necessariamente conoscere l'intera specifica OGC, in quanto è il Server stesso a pubblicare tutte le informazioni necessarie per il suo utilizzo. Infine, l'uso di standard tipici del Web, come HTTP e HTML, rendono gli endpoint del servizio indicizzabili dai motori di ricerca. In tal modo, un qualsiasi utente può "scoprire" il servizio semplicemente effettuando una ricerca sul web.

La famiglia di standard OGC API copre tutte le tipologie di dati spaziali, con la pubblicazione delle specifiche per ciascuna di esse, di seguito sono elencate alcune delle principali:

- **OGC API – Features** per la pubblicazione di dataset vettoriali
- **OGC API – Coverages** per la pubblicazione di dataset raster

- **OGC API – Records** per la pubblicazione di record di metadati
- **OGC API - Environmental Data Retrieval** per la pubblicazione di dati ambientali

Sebbene le specifiche differiscano per alcuni dettagli implementativi, tutte si basano sugli stessi principi elencati precedentemente. Per tale motivo, per questo lavoro è stata presa in considerazione una sola specifica, in particolare OGC API – Features. Resta inteso che gran parte dei concetti possono essere estesi a tutte le specifiche.

3.1 OCG API – Features

Nota anche come *WFS 3.0*, la specifica *OGC API – Features* descrive i dettagli per l'implementazione di server per la creazione, la modifica e l'interrogazione di dataset vettoriali. La specifica è composta da più parti, di cui la prima (*Core*) descrive le capacità minime che deve supportare un servizio che vi aderisce ed è limitata alle operazioni di "discovery" e di accesso in lettura ai dati vettoriali.

Lo standard propone una interfaccia standardizzata per la pubblicazione di dataset vettoriali (definiti *collections*). L'elemento atomico che compone un dataset, ovvero il dato che ha componenti spaziali, è denominato *Feature*.

L'accesso ai dati avviene mediante richieste HTTP di tipo GET verso URI univoci assegnati a ciascuna risorsa (una collection, una feature, ecc.). La specifica OGC API – Features stabilisce quattro path per l'accesso ai dati spaziali pubblicati dal server, riassunti nella Tabella 1.

Tabella 1. Schema degli URI identificativi delle risorse

Risorsa	Path	Scopo
Feature collections	*/collections*	Ottenere l'elenco dei dataset pubblicati dal server
Feature collection	*/collections/{collectionId}*	Ottenere i metadati relativi a una collection
Features	*/collections/{collectionId} /items*	Ottenere l'elenco delle features presenti nella collection
Feature	*/collections/{collectionId} /items/{featureId}*	Ottenere le proprietà di una feature

Ogni richiesta a tali endpoint può contenere anche parametri per il filtraggio spaziale, temporale o sui valori degli attributi dei risultati. In tal modo, anche se il dataset contiene numerose features, l'utilizzatore può ottenere solo il sottoinsieme cui è interessato.

La Figura 3 presenta lo stesso dataset della Figura 1, questa volta pubblicato da un server che aderisce alle specifiche OGC API – Features. La figura evidenzia come il formato GeoJSON adottato dalla specifica renda la notazione più compatta e leggibile: anche un utente non esperto del dominio può comprenderne le informazioni con una rapida lettura. Inoltre, la Figura 4 mostra lo stesso dataset pubblicato dal server anche

in formato HTML e visualizzato in un browser web. Si noti come, a differenza dei servizi WFS, dove l'utente aveva necessità di utilizzare un software separato per la visualizzazione dei dati, un server OGC API è esso stesso un semplice visualizzatore, permettendo la consultazione dei dati anche a utenti completamente agnostici ai concetti GIS.


```
{
    "type": "FeatureCollection",
    "features": [
        {
            "type": "Feature",
            "geometry": {
                "type": "Point",
                "coordinates": [
                    13.900488,
                    42.893576
                ]
            },
            "properties": {
                "reg": "13",
                "dec_tipo": "Intersezione con confine regionale (competenza)",
                "nome": ""
            },
            "id": "RT.RETEFERROVIARIA.NODOFERROVIARIO.1"
        }
    ],
    "links": [
        ...
    ],
    "timeStamp": "2022-05-20T12:40:48.607824Z"
}
```


Figura 3 Dataset vettoriale di Figura 1, pubblicato con un servizio OGG API – Feature

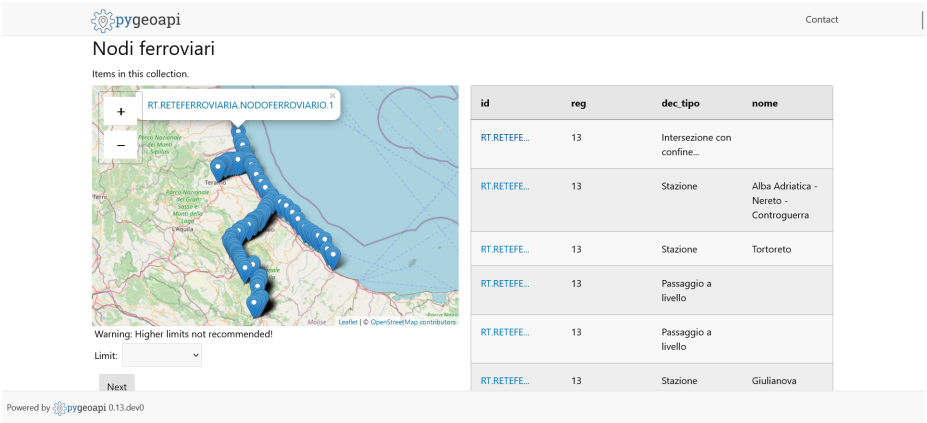

Figura 4 Dataset vettoriale in formato HTML. La visualizzazione grafica permette una consultazione semplice per qualsiasi tipo di utente.

4 Pubblicare dataset tramite OGC API

Come dimostrazione dei concetti presentati in questo lavoro, è stata realizzata una piattaforma che coprisse sia la pubblicazione di dataset territoriali mediante OGC API – Features, che l'utilizzo delle API per la realizzazione di una applicazione web.

La piattaforma è stata realizzata nel contesto della filiera della "*Nocciola di Giffoni IGP*", dove, da studi precedenti, erano state realizzate diverse carte tematiche relative al territorio di coltivazione (DEM, precipitazioni, limiti territoriali, ecc.). Scopo della piattaforma era, dunque, offrire un sistema di supporto alle decisioni attraverso la consultazione e l'analisi di tali tematismi sul web.

I dataset sono serviti da un server OGC API, per la cui implementazione è stato utilizzato il software open source *pygeoapi* [5]. È da sottolineare che il codice client utilizzato resta immutato con qualsiasi altra soluzione tra quelle certificate OGC API.

Con una richiesta HTTP di tipo GET al path */collections* è possibile ottenere l'elenco dei dataset pubblicati dal Server. In Figura 5 è mostrato il risultato della richiesta, visualizzata in formato HTML tramite un browser web. Ovviamente, richiedendo i dati in formato GeoJSON è possibile integrarli in altre applicazioni client.

Collections in this service

Name	Type	Description
Altitudine		Altitudine
Esposizione		Esposizione
Pendenza		Pendenza
Precipitazioni		Precipitazioni
Temperatura massima		Temperatura massima periodo Luglio-Agosto
Temperatura media		Temperatura media periodo 15 Maggio - 15 Giugno
Comuni IGP	feature	Limiti comuni appartenenti al IGP
Sistemi terre	feature	Sistemi terre
Aziende	feature	Aziende
Terreni Aziende	feature	Terreni appartenenti alle aziende del IGP, con informazioni relative alla coltura
Attitudine climatica	feature	Mappa di attitudine climatica areale di produzione "Nocciola di Giffoni IGP"

Figura 5 Elenco dei dataset offerti dal server

La piattaforma client, realizzata con il framework *Django* [6], permette la selezione di tematismi multipli, che saranno poi visualizzati come layer sovrapposti all'interno della mappa. In Figura 6 è mostrato un esempio della piattaforma.

Selezionato il tematismo da visualizzare, è necessario ottenere dal server tutte le features contenute nella collection. Anche in questo caso, è necessario effettuare una richiesta HTTP al path */collections/{collectionId}/items*. La Figura 7 mostra come sia semplice effettuare programmaticamente tale richiesta utilizzando il linguaggio di programmazione Python.

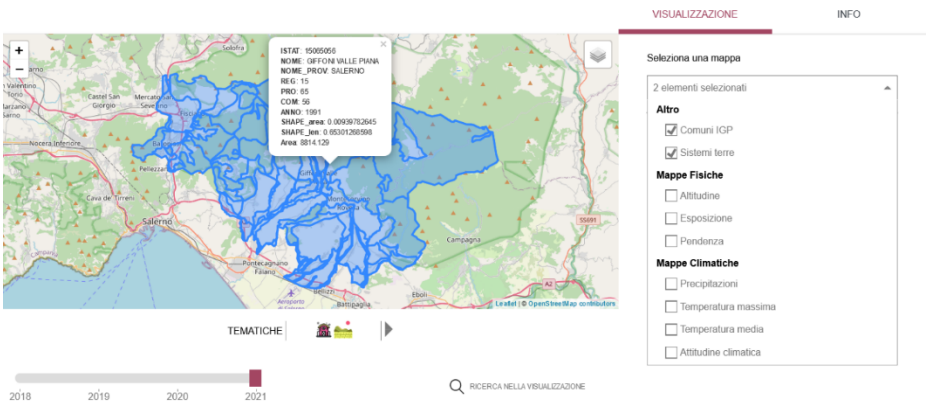

Figura 6 Esempio di visualizzazione. Sulla mappa sono visualizzate le due mappe selezionate dalla lista sulla destra

```
def get_map_data():
    server_url = settings.PYGEOAPI_URL
    collection_url = f"{server_url}/collections/{collection_id}/ "

    map_metadata = requests.get(collection_url).json()
    if "itemType" in map_metadata and map_metadata["itemType"] == "feature":
        # features collection
        data = requests.get(f"{collection_url}/items", params={"f": "json",
                            "limit": 20000}).json()

    return data
```

Figura 7 Richiesta programmatica delle features di un dataset pubblicato su server OGC API (codice Python)

La sovrapposizione dei tematismi permette di ottenere nuove informazioni combinando i dati provenienti da dataset differenti.

Sebbene la piattaforma offra solamente la possibilità di visualizzare i dataset, le OGC API rendo accessibili i dati anche a strumenti automatizzati, aprendo, in ottica futura, nuove prospettive di analisi non possibili fin ora. Ad esempio, si potrebbero applicare algoritmi di Machine Learning sui dataset a disposizione. In questo modo, fornendo all'algoritmo informazioni su un nuovo terreno (esposizione, temperatura mdeia, precipitazioni, ecc.) esso potrebbe essere capace di stabilire con un discreto grado di certezza se il terreno sia adatto o meno alla coltivazione. Inoltre, per aumentare la quantità di informazioni a disposizione degli algoritmi, si potrebbe prevedere l'installazione di sensori in loco i cui dati registrati sono distribuiti tramite OGC API. L'analisi di questi nuovi tipi di dati può avere un grande impatto per l'ottimizzazione dei processi produttivi e il miglioramento delle tecniche di coltivazione, rendendole più efficaci e sostenibili.

5 Conclusioni

La famiglia di standard OGC API offre un nuovo approccio per la distribuzione di dati territoriali online, applicando le best practice del W3C per la pubblicazione dei dati online ed ereditandone tutti i vantaggi.

Una interfaccia API uniforme, che utilizza i moderni standard del web, ben conosciuti dagli sviluppatori, rende l'accesso ai dati territoriali semplice e flessibile, agevolando la realizzazione di software e piattaforme.

I dati spaziali sono "a portata di click" per i cittadini, che possono accedervi con una semplice ricerca sul web, e diventano un valore aggiunto per amministrazioni e aziende che possono sfruttarli, grazie a nuovi strumenti di analisi, per costruire città più sostenibili e processi produttivi moderni ed efficienti.

Se sfruttate in modo adeguato sia da chi produce i dati che da chi li consuma, le OGC API possono essere uno strumento importante per abilitare azioni che assicurino sostenibilità over time come richiesto dagli obiettivi dell'Agenda 2030.

Ringraziamenti

L'autore ringrazia il Dott. Carlo Cipolloni di ISPRA il prezioso contributo e il Dott. Pietro Battistoni di UniSa per il valido supporto, durante la realizzazione delle attività descritte nel presente lavoro.

Riferimenti bibliografici

1. Hu, Y. & Li, W. (2017). "Spatial Data Infrastructures", The Geographic Information Science & Technology Body of Knowledge, John P. Wilson (ed.).
2. Geoportale Nazionale, Servizio di scaricamento WFS, http://www.pcn.minambiente.it/mattm/servizio-di-scaricamento-wfs.
3. World Wide Web Consortium, "Data on the Web Best Practices", https://www.w3.org/TR/dwbp.
4. Open API Initiative, "OpenAPI Specification", https://spec.openapis.org/oas/latest.html.
5. Pygeoapi, "Pygeoapi Documentation", https://docs.pygeoapi.io/en/stable/.
6. Django Software Foundation, Django homepage, https://www.djangoproject.com/.