ELISE action
Webinar Series

Workshop: Smart Data Loader and Templating for GeoServer

11/02/2021 at 14:00 CET (UTC+1)

European Location Interoperability Solutions for e-Government

Enabling Digital Government through Geospatial and Location Intelligence

ISA²

ISA² Programme & ELISE action

European Interoperability Programme

cross-border *and* ***cross-sector*** *Interoperability solutions*

for ***public administrations, businesses*** *and* ***citizens***

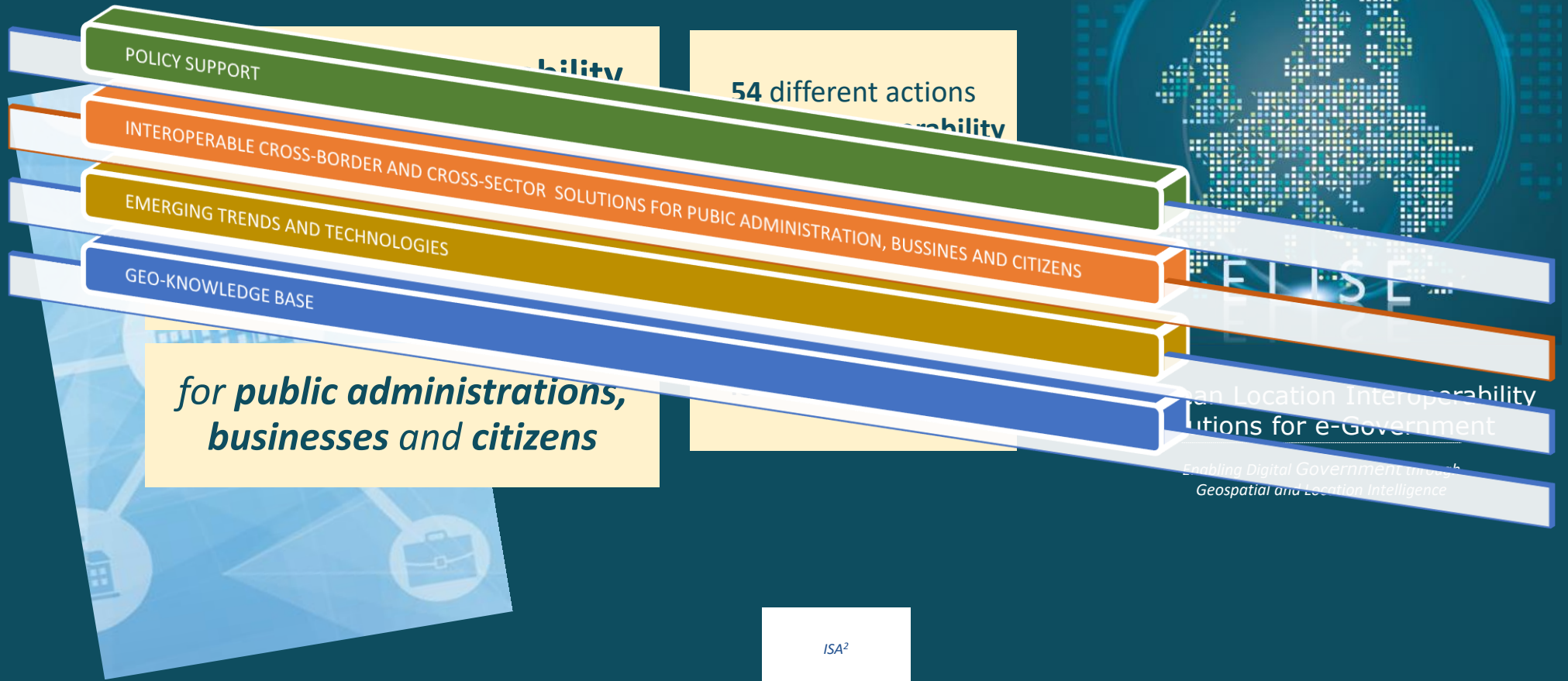
54 different actions tackling **interoperability** from different angles

ELISE action is the **only** action focusing on the **location dimension**

European Location Interoperability Solutions for e-Government

Enabling Digital Government through Geospatial and Location Intelligence

ELISE action



Welcome to the ELISE webinar series

ELISE **Knowledge Transfer** activities

Purpose:

- Engage in **an agile way**
- with **topics** of relevance to the **Digital Transformation**
- **by harnessing the use of spatial data and technology.**
- **Validate** and s**hare** the results **of ELISE activities.**

https://europa.eu/!nP74ph

Our speakers

Kathi Schleidt

kathi@datacove.eu

INSPIRE Expert

Nuno Oliviera

nuno.oliveira@geo-solutions.it

Expert on OGC Services

The views expressed are purely those of the authors and may not in any circumstances be regarded as stating an official position of the European Commission.

What we will cover today

- 14:05-14:45 Overview OGC API – Features: What is the new standard?
- 14:40-14:45 Break
- 14:45-15:15 Smart Data Loader
- 15:15-15:20 Break
- 15:20-15:50 Feature Templating
- 15:50-16:00 Wrap-up

Introduction

- OGC API – Features is the follow-up Standard for WFS2 (was initially titled WFS3)
- Accepted as an INSPIRE Good Practice: https://inspire.ec.europa.eu/good-practice/ogc-api-%E2%80%93-features-inspire-download-service
- Supporting Material on GitHub: https://github.com/INSPIRE-MIF/gp-ogc-api-features https://github.com/INSPIRE-MIF/gp-ogc-api-features/blob/master/spec/oapif-inspire-download.md

DATA COVE

Geoscience for a sustainable Earth brgm

Geo Solutions

Introduction

- OGC API – Features is the follow-up Stanc (was initially titled WFS3)
- Accepted as an INSPIRE Good Practice: https://inspire.ec.europa.eu/good-practic features-inspire-download-service
- Supporting Material on GitHub: https://github.com/INSPIRE-MIF/gp-ogc-a https://github.com/INSPIRE-MIF/gp-ogc-a features/blob/master/spec/oapif-inspire-

OGC API – Features as an INSPIRE download service

Several possible solutions for implementing download services are already endorsed by the INSPIRE Maintenance and Implementation (MIG) group. Technical guidelines documents are available that cover implementations based on ATOM, WFS 2.0, WCS and SOS. While all of these approaches use the Web for providing access to geospatial data, the new family of OGC API standards aim to be more developer friendly by requiring less up-front knowledge of the standard involved. The rapid emergence of Web APIs provide a flexible and easily understandable means for access to data, as recommended by the W3C Data on the Web Best Practices.

This good practice proposes an additional option for the implementation of INSPIRE download services. The draft guideline for setting up an INSPIRE Download service based on the OGC API-Features standard, together with implementation evidences are available on GitHub.

In order to facilitate the use of off-the-shelf software implementing the OGC API - Features standard to meet the requirements in this document, INSPIRE-specific extensions are limited to the absolute minimum. Where several implementation options exist, this document guideline defines a specific way of application of the OAPIF and associated standards to meet the requirements of the INSPIRE Implementing Rules for download services.

Relevance & expected benefits:
The adoption of the OGC API – Features standard as an INSPIRE Download service would provide a modern approach for the exposure of geospatial data that adheres to the recommendations of the W3C Data on the Web Best Practices. Both implementers of INSPIRE and users of the data would benefit from the powerful, yet simple approach for data sharing.

References:
- Mandate of MIWP action 2020.1 on INSPIRE Download Services based on the OGC API – Features standard
- Discussion paper on Mapping the requirements from the INSPIRE IR's for Network Services with the OGC – API Features standard
- The OGC API – Features standard

Outcome:
- Technical guideline for setting up INSPIRE Download services based on the OGC API - Features standard
- Abstract Test Suite
- Implementation evidences

Evidence:
The INSPIRE MIWP Action 2020.1 is currently collecting evidences for the implementation of the OGC API – Features for INSPIRE in a structured manner. Those will be provided when submitting the good practice for endorsement by the MIG.

Limitations:
Sharing of coverage data is out of the scope of this good practices. Executable tests and validation of OGC API – Feature instances are also not covered in this good practice as they will be dealt with within the context of the work on the INSPIRE Reference validator.

IKB Area:
Implement
Use
IKB Component:
Metadata
Download Services

Tags:
validator
Normative Reference:

Introduction

- OGC API – Features is the follc (was initially titled WFS3)
- Accepted as an INSPIRE Good https://inspire.ec.europa.eu/g features-inspire-download-se
- Supporting Material on GitHu https://github.com/INSPIRE-N https://github.com/INSPIRE-N features/blob/master/spec/oa

Setting up an INSPIRE Download service based on the OGC API-Features standard

`Version: 1.0` `Date: 2021-02-05`

Table of Contents

1. Introduction

This document proposes a technical approach for implementing the requirements set out in the INSPIRE Implementing Rules for download services based on the newly adopted OGC API - Features standard.

Several possible solutions for implementing download services are already endorsed by the INSPIRE Maintenance and Implementation (MIG) group. Technical guidelines documents are available that cover implementations based on ATOM, WFS 2.0, WCS and SOS.

While all of these approaches use the Web for providing access to geospatial data, the new family of OGC API standards aim to be more developer friendly by requiring less up-front knowledge of the standard involved. The rapid emergence of Web APIs provide a flexible and

Overview

- OGC API – Common
- OGC API – Features (Core)
- OGC API – Extensions
- Example Use Case – BRGM boreholes
- One Step Back to the Future!

OGC API - Common

- Specification available from: https://github.com/opengeospatial/oapi_common
- Resources:
  - `/` → landing page
  - `/conformance` → conformance classes implemented
  - `/api` → API definition
  - `/collections` → list of collections published
  - `/collection/{collectionId}` → metadata of the single collection
- Encodings:
  - HTML (for people to browse an API)
  - JSON/GeoJSON

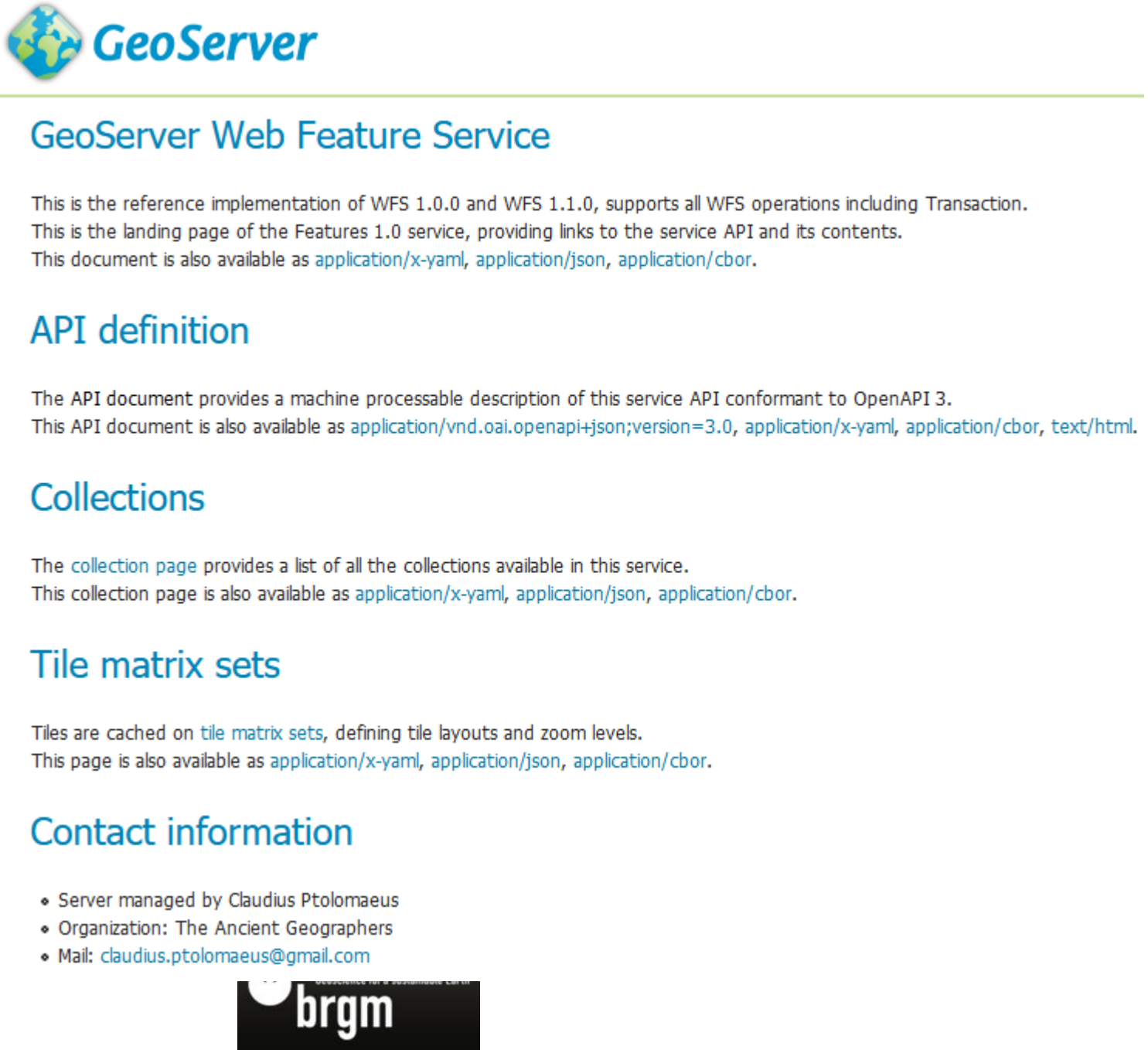
DATA COVE

Geoscience for a sustainable Earth brgm

Geo Solutions

OGC API - Co

- Specification availab
  https://github.com/
- Resources:
  - `/`
  - `/conformance`
  - `/api`
  - `/collections`
  - `/collection/{col`
- Encodings:
  - HTML (for people tc
  - JSON/GeoJSON



DATA COVE

brgm

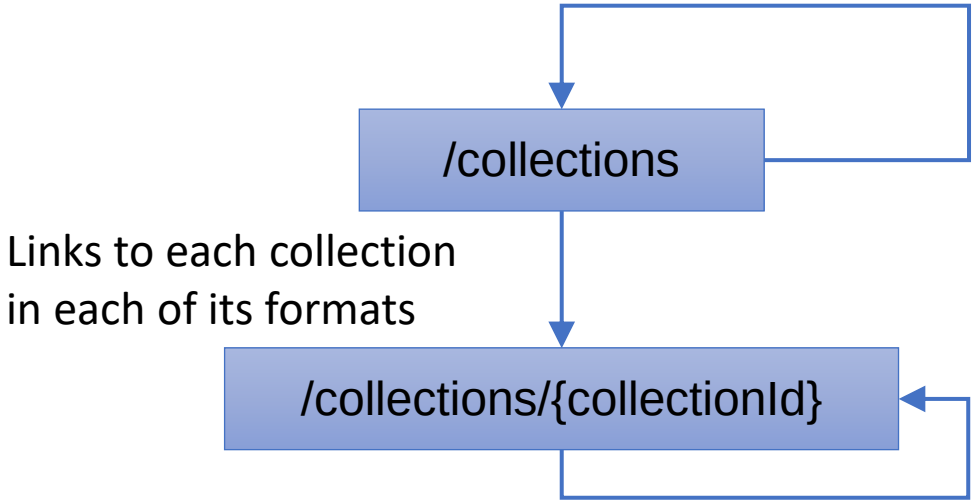
Geo Solutions

OGC API - Co

- Specification availab
  https://github.com/
- Resources:
  - **/**
  - **/conformance**
  - **/api**
  - **/collections**
  - **/collection/{col**
- Encodings:
  - HTML (for people tc
  - JSON/GeoJSON

Links, links everywhere!

All resources are linked to the others, in the various encodings:



Backlinks to self and
alternative representations
(different formats/encodings)

```
{
   "href":"http://data.example.org/collections.json",
   "rel":"self",
   "type":"application/json",
   "title":"this document"
}
```

Very little is mandatory

None of the encodings are mandatory, a server could do XML or protocol buffers and still be compliant:

- The **`/api`** description is not mandatory
- If present, the usage of OpenAPI is suggested but not mandatory!
- A client could work by following links between resources

How does a client work then??
→ By checking the conformance declaration at **`/conformance`**:

```
{
  "conformsTo": [
    "http://www.opengis.net/spec/ogcapi-features-1/1.0/conf/core",
    "http://www.opengis.net/spec/ogcapi-features-1/1.0/conf/oas30",
    "http://www.opengis.net/spec/ogcapi-features-1/1.0/conf/html",
    "http://www.opengis.net/spec/ogcapi-features-1/1.0/conf/geojson"
```

DATA COVE

Geoscience for a sustainable Earth brgm

Geo Solutions

Very little is manc

None of the encodings are mandatory compliant:

- The **`/api`** description is not manda
- If present, the usage of OpenAPI is
- A client could work by following link

How does a client work then??
→ By checking the conformance decla

```
{
  "conformsTo": [
    "http://www.opengis.net/s
    "http://www.opengis.net/s
    "http://www.opengis.net/s
    "http://www.opengis.net/s
```

A.2. Conformance Class Core

Conformance Class	
http://www.opengis.net/spec/ogcapi-features-1/1.0/conf/core	
Target type	Web API

A.2.1. General Tests

A.2.1.1. HTTP

Abstract Test 1	**/ats/core/http**
Test Purpose	Validate that the resource paths advertised through the API conform with HTTP 1.1 and, where approprate, TLS.
Requirement	/req/core/http
Test Method	1. All compliance tests shall be configured to use the HTTP 1.1 protocol exclusively. 2. For APIs which support HTTPS, all compliance tests shall be configured to use HTTP over TLS (RFC 2818) with their HTTP 1.1 protocol.

A.2.1.2. CRS 84

Abstract Test 2	**/ats/core/crs84**
Test Purpose	Validate that all spatial geometries provided through the API are in the CRS84 spatial reference system unless otherwise requested by the client.
Requirement	/req/core/crs84
Test Method	1. Do not specify a coordinate reference system in any request. All spatial data

DATA COVE

Very little is mar

None of the encodings are mandat
compliant:

- The `/api` description is not man
- If present, the usage of OpenAPI
- A client could work by following l

How does a client work then??
→ By checking the conformance de

```
{
  "conformsTo": [
    "http://www.opengis.net
    "http://www.opengis.net
    "http://www.opengis.net
    "http://www.opengis.net
```

A.3. Conformance Class GeoJSON

Conformance Class	
http://www.opengis.net/spec/ogcapi-features-1/1.0/conf/geojson	
Target type	Web API

A.3.1. GeoJSON Definition

Abstract Test 30	**/ats/geojson/definition**
Test Purpose	Verify support for JSON and GeoJSON
Requirement	/req/geojson/definition
Test Method	

1) A resource is requested with response media type of `application/geo+json`

2) All 200-responses SHALL support the following media types:

- `application/geo+json` for resources that include feature content, and

- `application/json` for all other resources.

A.3.2. GeoJSON Content

Abstract Test 31	**/ats/geojson/content**
Test Purpose	Verify the content of a GeoJSON document given an input document and schema.

Overview

- OGC API – Common
- OGC API – Features (Core)
- OGC API – Extensions
- Example Use Case – BRGM boreholes
- One Step Back to the Future!

OGC API – Features (Core)

- In addition to OGC API (OAPI) core:
  - `/collection/{collectionId}/items` → features in collection
  - `/collection/{collectionId}/items/{itemId}` → individual feature
- Only supported CRS are
  - CRS84 (WGS84 lon/lat)
  - CRS84h (WGS84 lon/lat/height)
- No mandated schema, features can be anything:
  - Simple
  - Complex
  - Heterogeneous
- Additional encodings:
  - gmlsf0 (flat)
  - gmlsf2 (non geometry properties can be nested)

`/api`

- The OpenAPI document can be setup in whatever way (assuming it matches the spec)

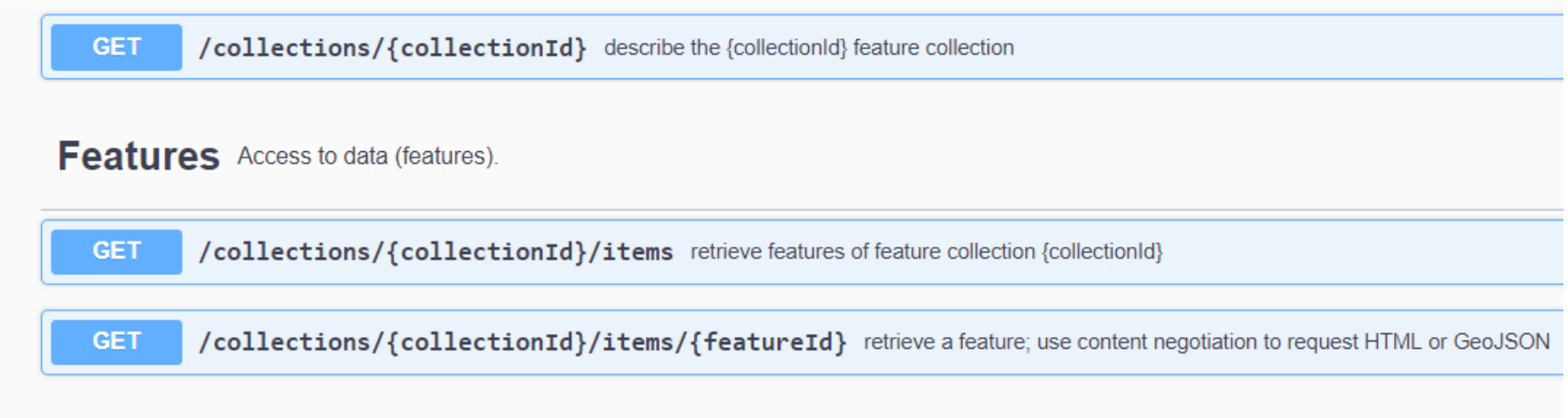
- Two commons approaches:
  - **Uniform collections**, all collections are described by a path as follows: **`/collections/{collectionId}`**

  - **Distinct collections**, each is given a separate path in the OpenAPI document: **`/collections/surfaceWaterQuantityStations`** **`/collections/groundWaterQuantityStations`**

Uniform collections

- Simpler
- Scales to thousands of collections easily
- Limited: cannot say anything unique to each collection

- GeoServer currently implements this approach



Distinct collections

- Verbose
- Suitable for small number of collections
- Can provide unique info about a collection, including
  - The returned JSON schema (if so desired)
  - Specific query parameters used for filtering (later)

/collection/{collectionId}/items

- Lists the contents of a collection
- Can be a GeoJSON document, GML, anything
- Filtering
  - bbox in CRS84 or CRS84h (or specify bbox-crs)
  - datetime (instant or range of times)
  - extra parameters declared in the API document (by equality)

```
…/collections/buildings/items?
      bbox=41,54,42,55
      &datetime=2018-02-12T23%3A20%3A52Z
      &buildingState=good
```

DATA COVE

Geoscience for a sustainable Earth brgm

Geo Solutions

Paging

- **`limit`** query parameter
- A (default) limit is always present (like in WFS 1.x and 2.x) but can be high
- Paging works through links, "**`prev`**" and "**`next`**" **`rel`** values
- Links can be implemented the way you want
- GeoServer uses **`offset`** as an extra paging parameter

```
[
   {
      "href":"http://example.org/collections/buildings/items.json?limit:50",
      "rel":"prev",
      "type":"application/geo+json"
   },
   {
      "href":"http://example.org/collections/buildings/items.json?limit:50&offset:50",
      "rel":"self",
      "type":"application/geo+json"
   },
   {
      "href":"http://example.org/collections/buildings/items.json?limit:50&offset:100",
      "rel":"next",
      "type":"application/geo+json"
   }
]
```

DATA COVE

Geo Solutions

Overview

- OGC API – Common
- OGC API – Features (Core)
- OGC API – Extensions
- Example Use Case – BRGM boreholes
- One Step Back to the Future!

Extensions

- Approach similar to GeoPackage.
- There will be extensions covering more functionality.
- Some extensions are already in the making:

  https://github.com/opengeospatial/ogcapi-features/tree/master/extensions

  - CQL filtering
  - CRS (reprojection and filtering)
  - Transactions

DATA COVE

Geoscience for a sustainable Earth brgm

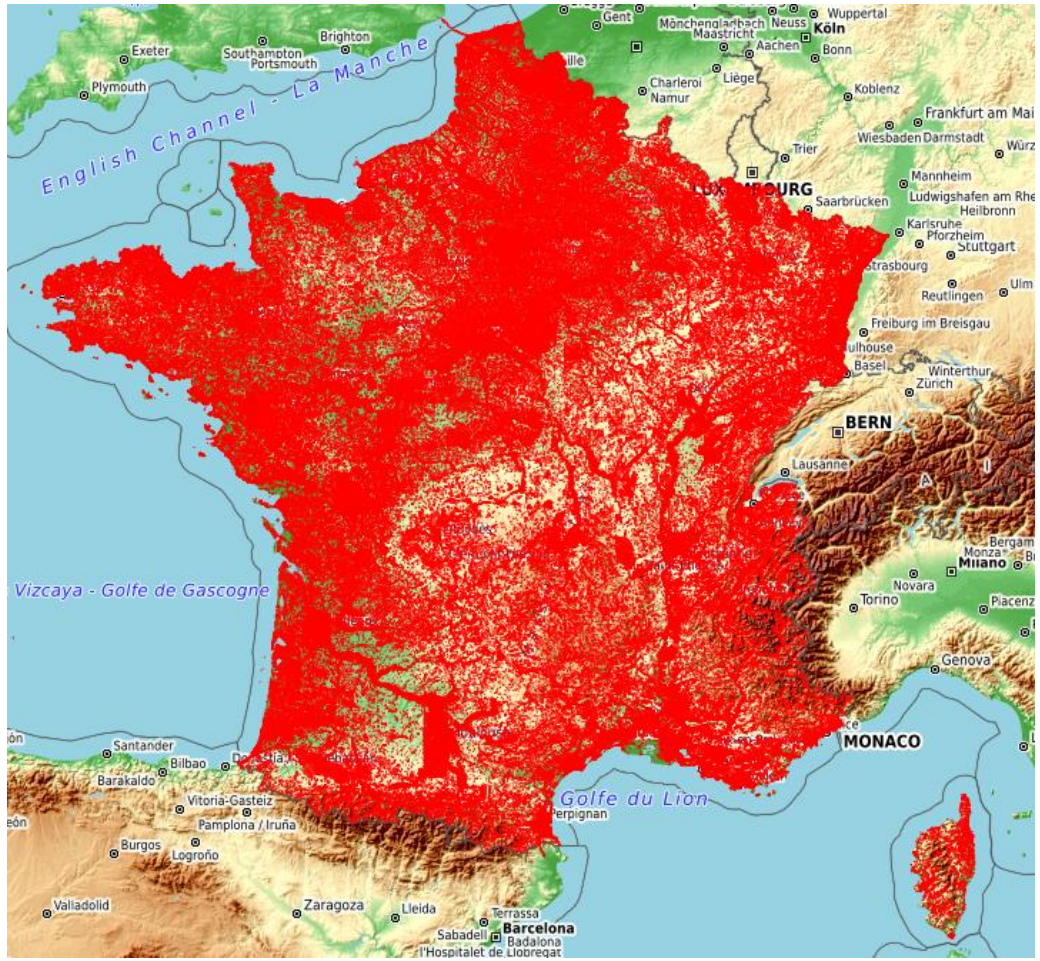
Geo Solutions

Overview

- OGC API – Common
- OGC API – Features (Core)
- OGC API – Extensions
- Example Use Case – BRGM boreholes
- One Step Back to the Future!

BRGM boreholes use case

- Demo with almost **1 million boreholes**:
- Available through:
  - WFS 2.0.0
  - ~~WFS 3.0.0~~
  - **OGC API – Features**
- Simple Features (SF-0) and **Complex Features (SF-1)**!



Features API landing page /

Service Capabilities

FEATURES
- 1.0

STYLES
- 1.0

WCS
- 1.0.0
- 1.1.0
- 1.1.1
- 1.1
- 2.0.1

WFS
- 1.0.0
- 1.1.0
- 2.0.0
- 3.0.0

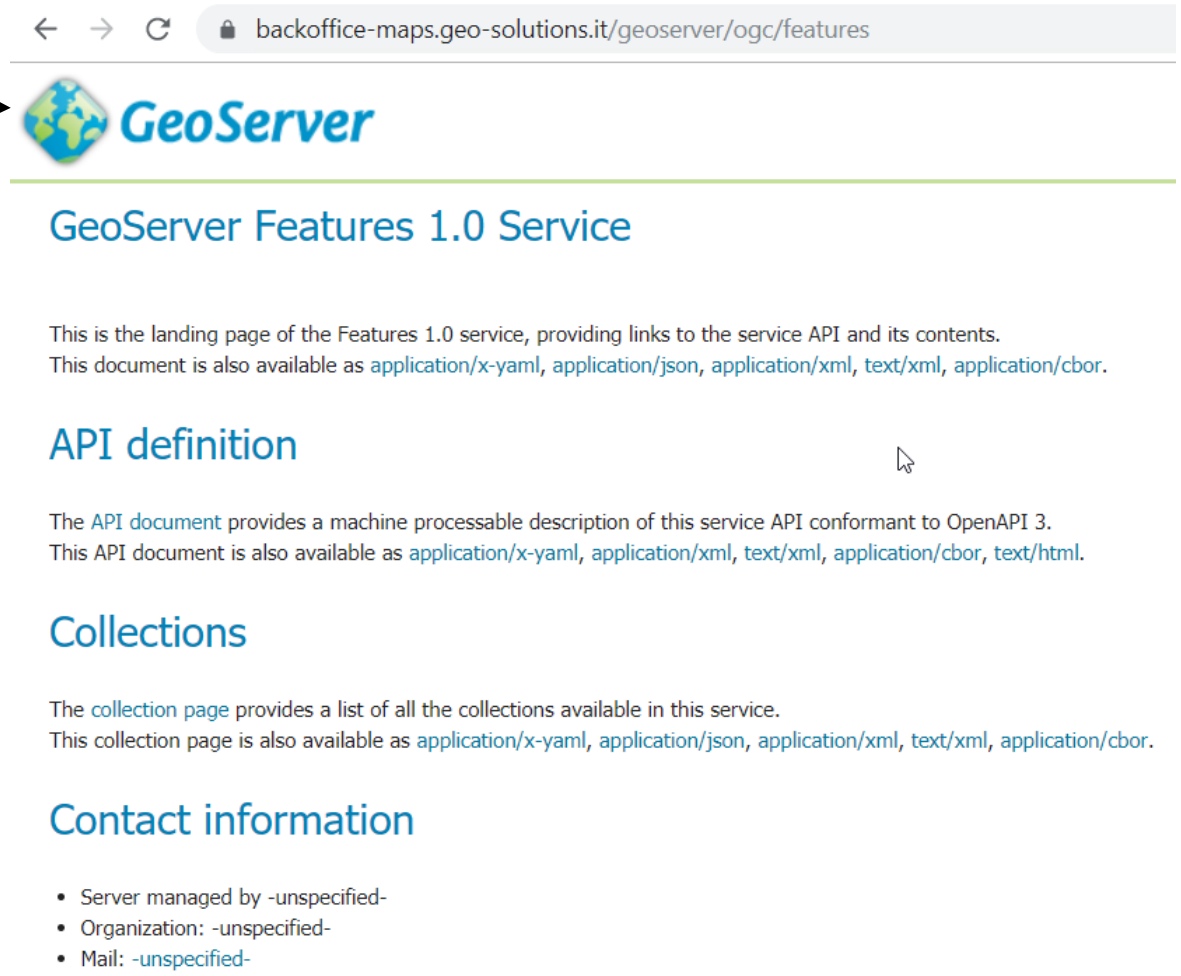
WMS
- 1.1.1
- 1.3.0

WPS
- 1.0.0

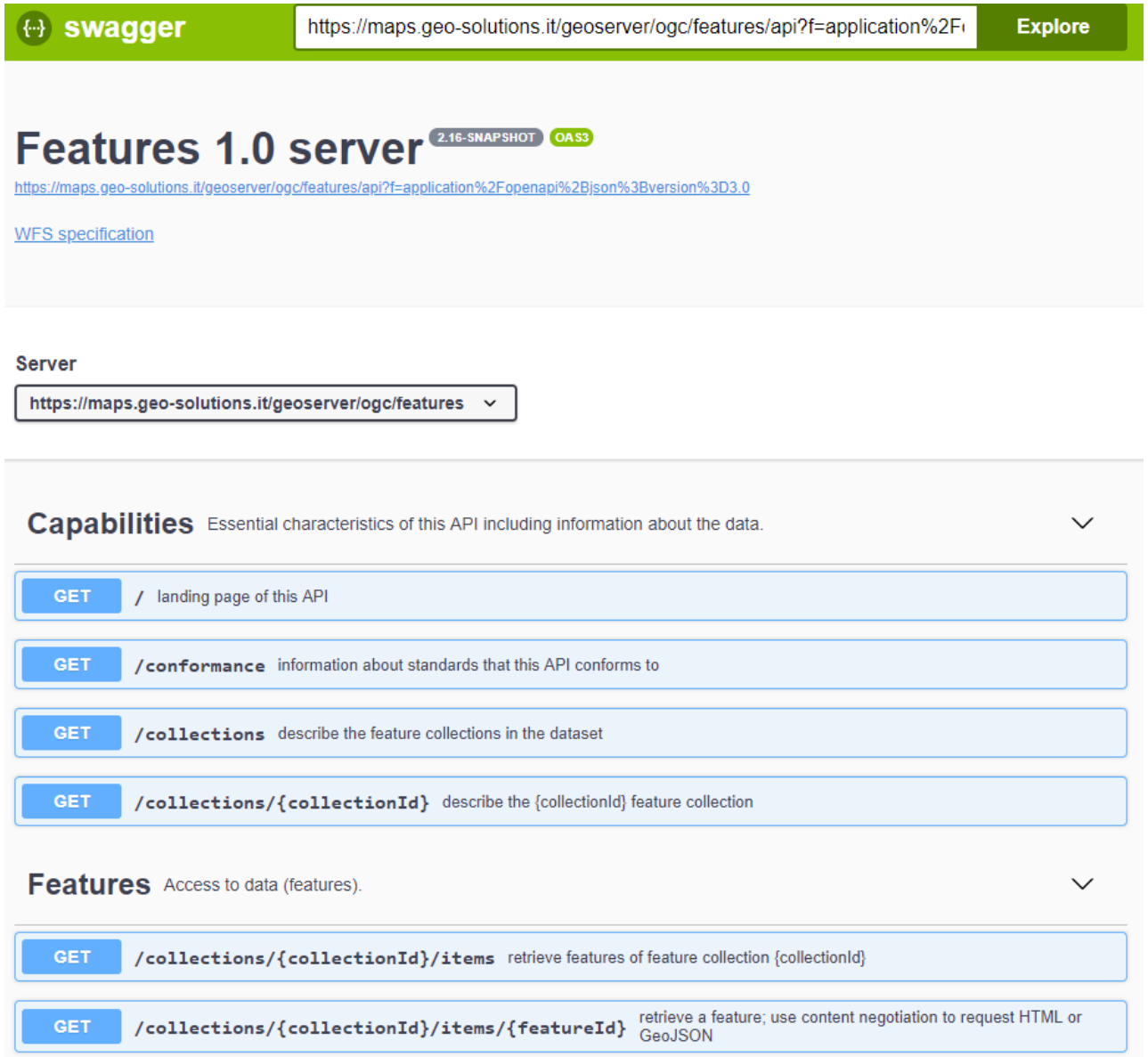
TMS
- 1.0.0

WMS-C
- 1.1.1

WMTS
- 1.0.0



Features API **`/api`**



Collections **`/collections`**

GeoServer Feature Collections

This document lists all the collections available in the Features service.
This document is also available as application/x-yaml, application/json.

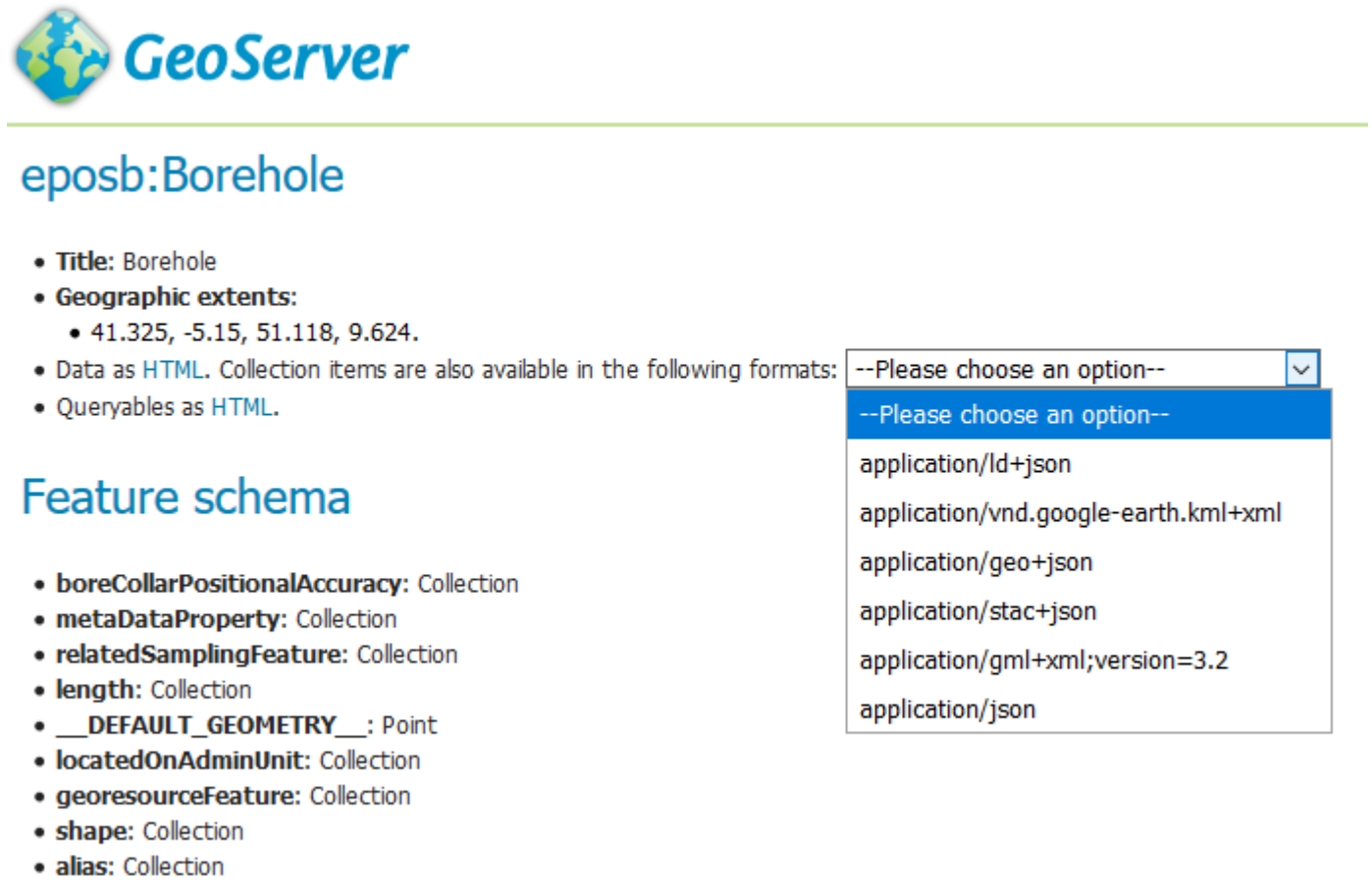
eposb:Borehole

- **Title:** Borehole
- **Geographic extents:**
  - 41.325, -5.15, 51.118, 9.624.
- Data as HTML. Collection items are also available in the following formats: --Please choose an option--
- Queryables as HTML.

gsmlp:BoreholeView

- **Title:** BoreholeView
- **Geographic extents:**
  - -64.656, -21.388, 139.535, 87.933.
- Data as HTML. Collection items are also available in the following formats: --Please choose an option--
- Queryables as HTML.

Collection **`/collections/{collectionId}`**



DATA COVE

Geoscience for a sustainable Earth brgm

Geo Solutions

Items **/collections/{collectionId}/items**

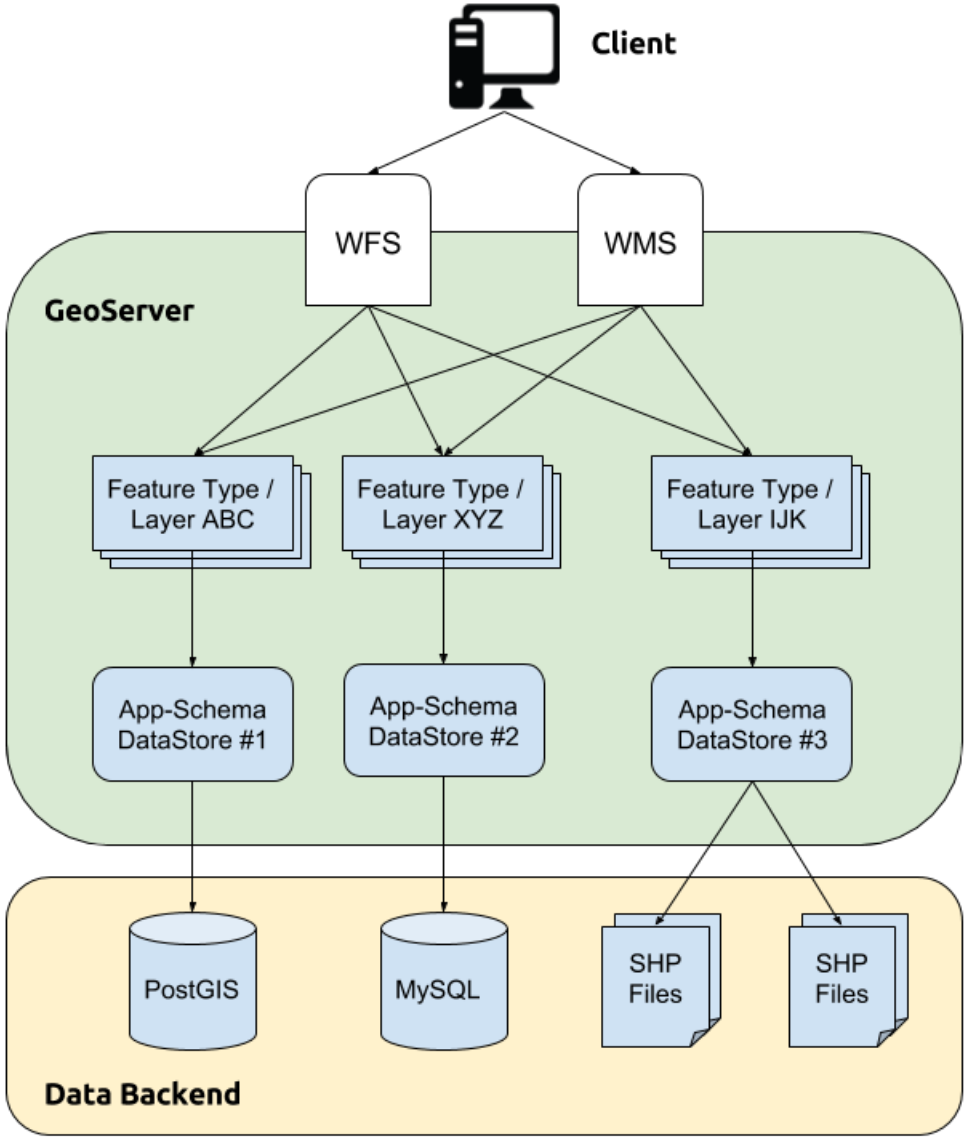

```
{
    type: "FeatureCollection",
  ▾ features: [
    ▾ {
          type: "Feature",
          id: "0001000001",
        ▾ geometry: {
              type: "Point",
            ▾ coordinates: [
                  [redacted],
                  [redacted]
              ]
          },
        ▾ properties: {
              "@featureType": "Borehole",
            ▾ identifier: {
                  value: "BSS000AAAA",
                  "@codeSpace": http://www.ietf.org/rfc/rfc2616
              },
            ▾ type: {
                  "@href": http://vocabulary.odm2.org/samplingfeaturetype/borehole/,
                  "@title": "borehole"
              },
            ▾ sampledFeature: [
                ▾ {
                      "@href": https://sweet.jpl.nasa.gov/2.3/realmEarthReference.owl#EarthLithosphere,
                      "@title": "Lithosphere"
                  }
              ],
            ▾ shape: {
                  "@dataType": "shape",
                ▾ Point: {
                      type: "Point",
                    ▾ coordinates: [
                          [redacted],
                          [redacted]
                      ]
                  }
              },
            ▾ custodian: {
                  "@dataType": "Custodian",
                ▾ custodian: {
                      "@dataType": "CI_ResponsibleParty",
                      organisationName: "BRGM",
                    ▾ role: {
                          "@codeListValue": "custodian",
                          "@codeList": http://www.isotc211.org/2005/resources/Codelist/gmxCodelists.xml
                      }
```

Overview

- OGC API – Common
- OGC API – Features (Core)
- OGC API – Extensions
- Example Use Case – BRGM boreholes
- One Step Back to the Future!

One step back

What to do with the existing mapped and published data sets?

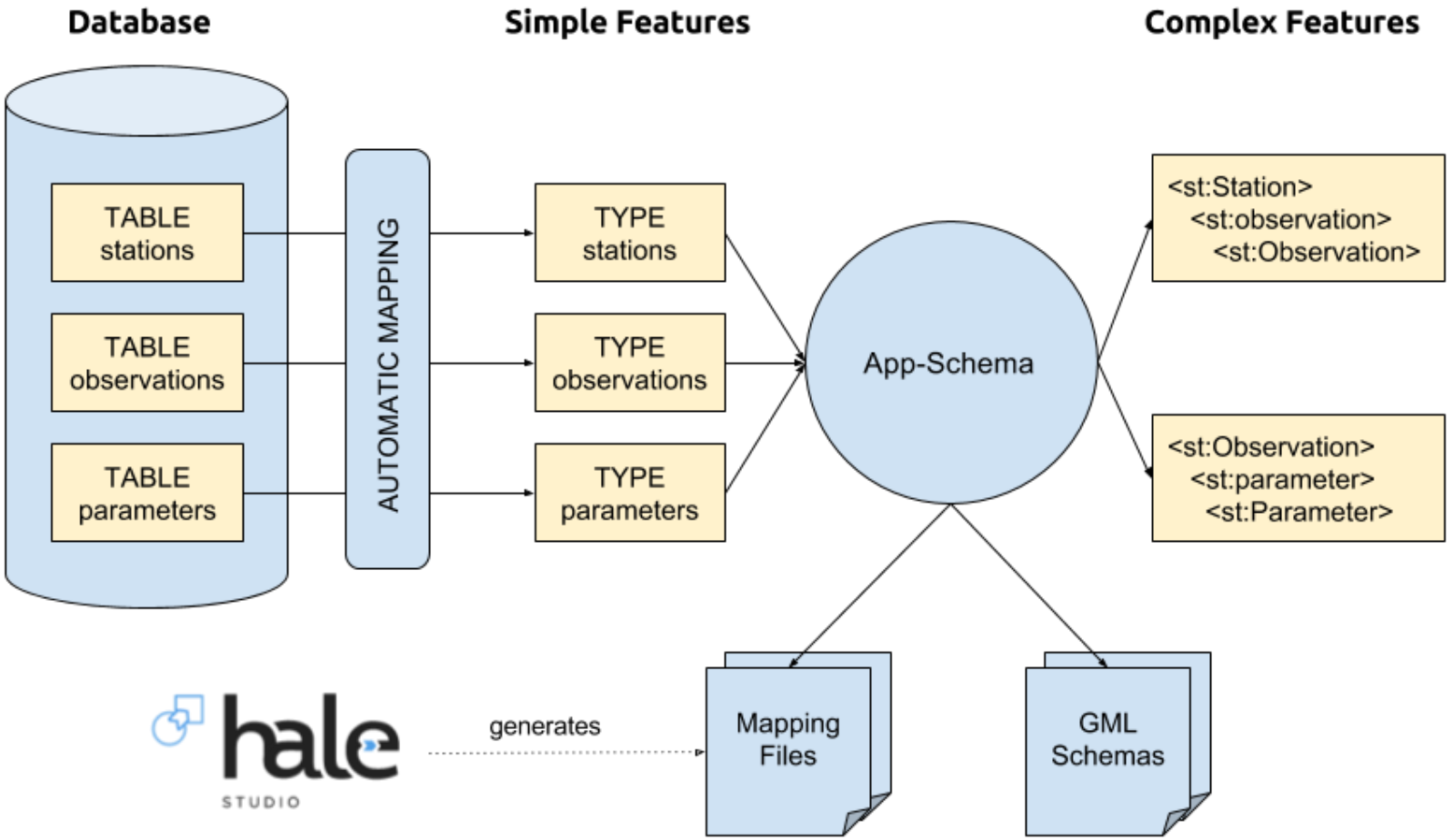


DATA COVE

Geoscience for a sustainable Earth brgm

Geo Solutions

One step back



DATA COVE

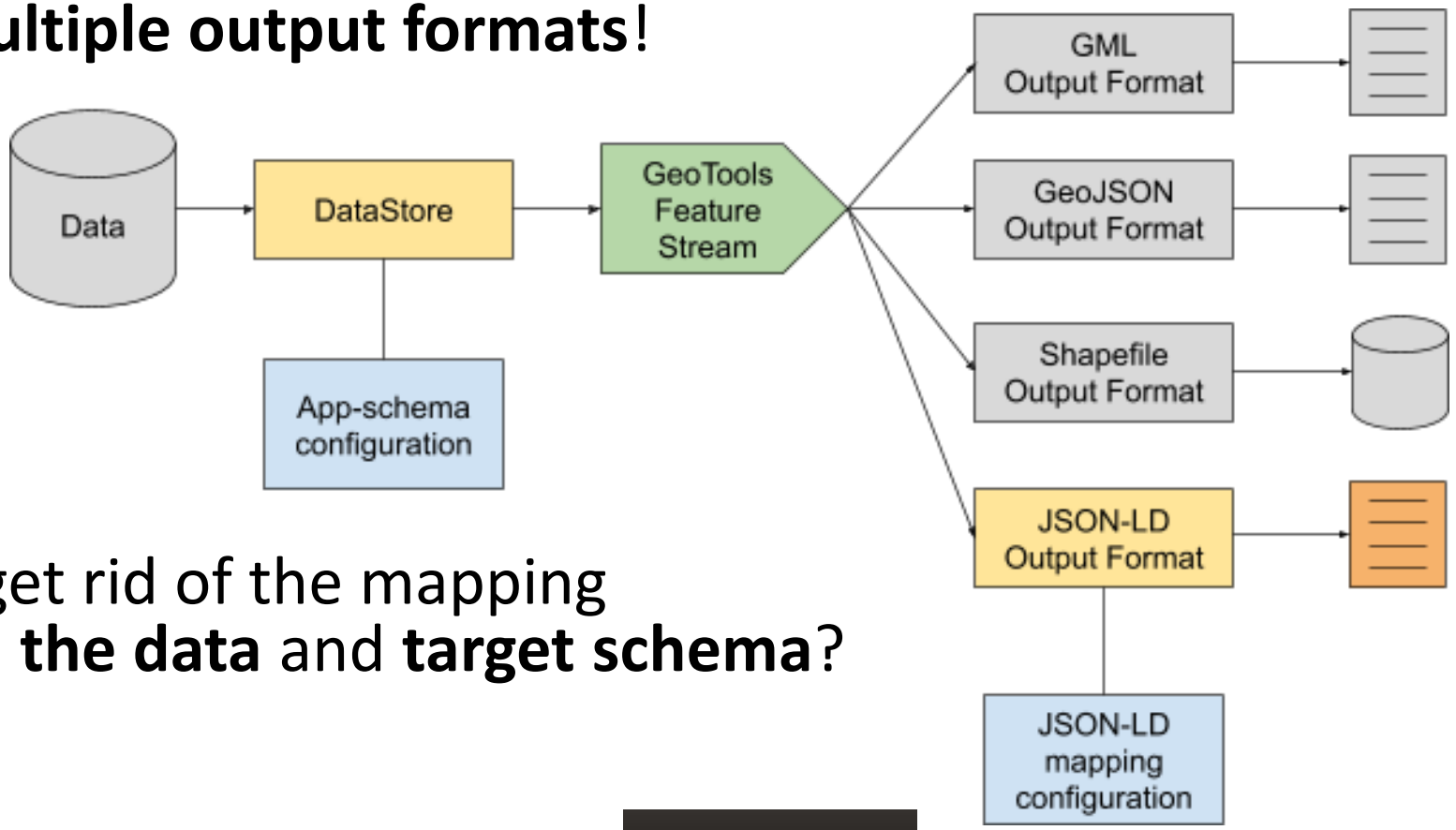
Geoscience for a sustainable Earth brgm

Geo Solutions

One step at a time

One mapping between the data and a target schema

➡ **multiple output formats**!



How to get rid of the mapping between **the data** and **target schema**?

DATA COVE

Geoscience for a sustainable Earth brgm

Geo Solutions

One step at a time

- Why not use templating instead of mappings?

- What You See Is What You Get!

DATA COVE

Geoscience for a sustainable Earth brgm

Geo Solutions

One step at a time

- Why not use templating instead of mappings?

- What You See Is What You Get!

```
"eposb:lifeCycleInfo": {
      "$source": "eposb:lifeCycleInfo/eposb:LifeCycleInfo",
      "@type": "LifeCycleInfo",
      "eposb:updateDate": {
        "@type": "time:Instant",
        "time:inXSDDateTime": "${eposb:updateDate/gml:TimeInstant/gml:timePosition}"
      },
      "status": {
        "@id": "${eposb:status/@xlink:href}",
        "name": "Validé"
      },
      "eposb:creationDate": {
        "@type": "time:Instant",
        "time:inXSDDateTime": "${eposb:creationDate/gml:TimeInstant/gml:timePosition}"
      }
    },
    "eposb:locatedOnAdminUnit": {
      "@id": "${eposb:locatedOnAdminUnit/@xlink:href}",
      "name": "SANGATTE"
    },
```

```
▾ "eposb:lifeCycleInfo": {
      "@type": "LifeCycleInfo",
    ▾ "eposb:updateDate": {
        "@type": "time:Instant",
        "time:inXSDDateTime": "2008/12/357 05:41:34"
      },
    ▾ status: {
        "@id": http://id.eaufrance.fr/nsa/390#XXX,
        name: "Validé"
      },
    ▾ "eposb:creationDate": {
        "@type": "time:Instant",
        "time:inXSDDateTime": "1998/03/70 11:55:17"
      }
  },
▾ "eposb:locatedOnAdminUnit": {
    "@id": "6262774",
    name: "SANGATTE"
  },
```

One step at a time

- Why not use templating instead of mappings?

- What You See Is What You Get!

```
"geometry": {
       "@type": "Point",
       "wkt":
"$${strConcat('<http://www.opengis.net/def/crs/OGC/1.3/CRS84>',toWKT(xpath('eposb:bholeHe
adworks/gwml-wellconstruction:BoreCollar/gwml-wellconstruction:collarLocation')))}"
     },
     "sam:sampledFeature": {
       "@id": "https://sweet.jpl.nasa.gov/2.3/realmEarthReference.owl#EarthLithosphere",
       "name": "Lithosphere"
     },
```

```
▾ geometry: {
       "@type": "Point",
       wkt: "<http://www.opengis.net/def/crs/OGC/1.3/CRS84>POINT (1.35559927237556 51.1020179771059)"
  },
▾ "sam:sampledFeature": {
       "@id": https://sweet.jpl.nasa.gov/2.3/realmEarthReference.owl#EarthLithosphere,
       name: "Lithosphere"
  },
```

Two steps at a time!

- Different communities require different data profiles
  → Need to create mapping for each community
- Far easier to create two templates!

Two s

- Differe
  → Nee
- Far easier to create two templates:

```
{
  "eposb:elevation": ${eposb:elevation},
  "eposb:elevation_srs_name": ${eposb:elevation@srsName},
  "eposb:elevation_srs_dimension": ${eposb:elevation@srsDimension},
  "eposb:elevation_uom_label": ${eposb:elevation@uomLabels},
  "eposb:elevation_measurement_method_href": ${eposb:elevationMeasurementMethod@xlink:href},
  "eposb:elevation_measurement_method_title": ${eposb:elevationMeasurementMethod@xlink:title},
  "eposb:elevation_type_href": ${eposb:elevationType@xlink:href},
  "eposb:elevation_type_title": ${eposb:elevationType@xlink:title}
}
```

```
{
  "eposb:elevation": 223.87,
  "eposb:elevation_srs_name": "http://www.opengis.net/def/crs/EPSG/0/5720",
  "eposb:elevation_srs_dimension": "1",
  "eposb:elevation_uom_label": "m",
  "eposb:elevation_measurement_method_href": "http://www.opengis.net/def/nil/OGC/0/unknown",
  "eposb:elevation_measurement_method_title": "unknown",
  "eposb:elevation_type_href": "https://epos.brgm-rec.fr/datalift/skos/vocabs/elevationtype/topOfCasing",
  "eposb:elevation_type_title": "Top of casing"
}
```

DATA COVE

Geoscience for a sustainable Earth brgm

Geo Solutions

Two steps at a time!

```
{
   "gsmlp:elevation_m": 223.87,
   "gsmlp:elevation_uom": "http://qudt.org/vocab/unit/M",
   "gsmlp:elevation_srs": "http://www.opengis.net/def/crs/EPSG/0/5720"
}
```

- Different communities require different data profiles
  → Need to create mapping for each community
- Far easier to create two templates!

```
{
   "gsmlp:elevation_m": ${eposb:elevation},
   "gsmlp:elevation_uom": ${eposb:elevation@uomLabels},
   "gsmlp:elevation_srs": ${eposb:elevation@srsName}
}
```

DATA COVE

Geoscience for a sustainable Earth brgm

Geo Solutions

More steps

- **Filtering using CQL** is supported:
  - http://.../geoserver/ogc/features/collections/eposb:Borehole/items?f=application/ld+json&limit=50&filter=**features.gsmlp:boreholeLength_m.om:amount>83**&filter-lang=cql-text

One step at a time

- **Filtering using CQL** is supported:
  - http://…/geoserver/ogc/features/collections/api4inspire:InitiativeFeature/items? f=application%2Fgeo%2Bjson& limit=50& filter=**features.properties.indicatorsInitiative.name='Cases24H'**& filter-lang=cql-text

Questionnaire Time!

Please go to:

https://www.menti.com/r6yuwqa2hh

Thank you and Stay tuned

Join the *ELISE* community in JoinUp

@eu_location

eulocation@ec.europa.eu

ELISE playlist