



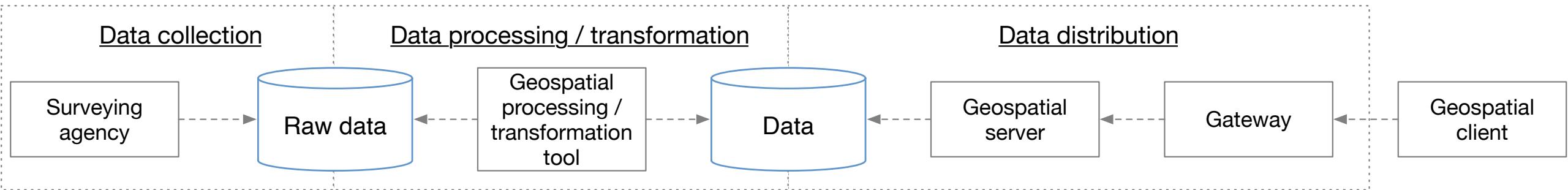
# **ETF testing framework**

**past, present and future**

Jon Herrmann, Marco Minghini, Clemens Portele, Michael Lutz, Pierpaolo Cira

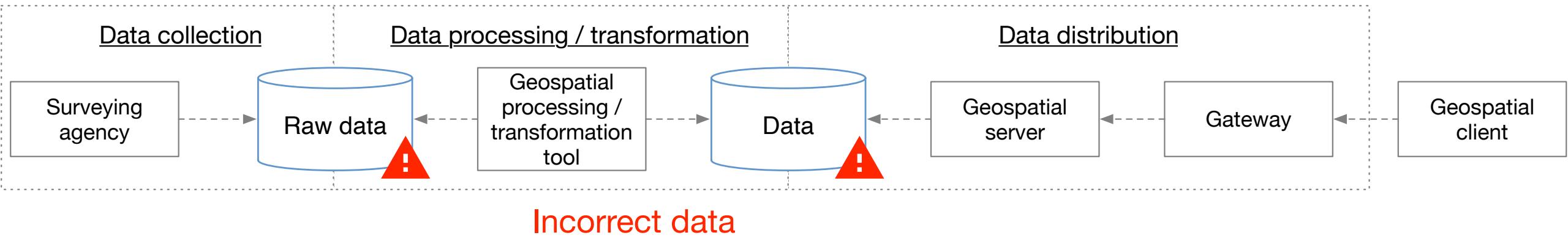
# Publishing data in an SDI

## What could possibly go wrong?



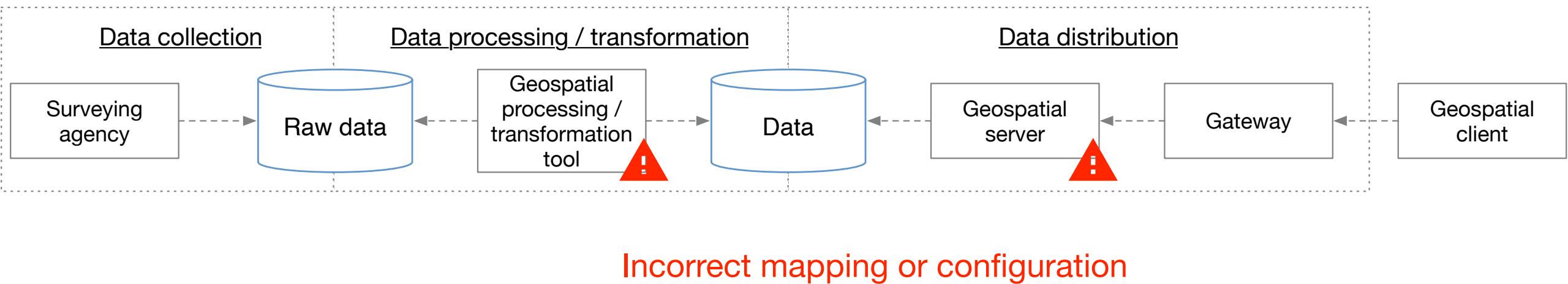
# Publishing data in an SDI

## What could possibly go wrong?



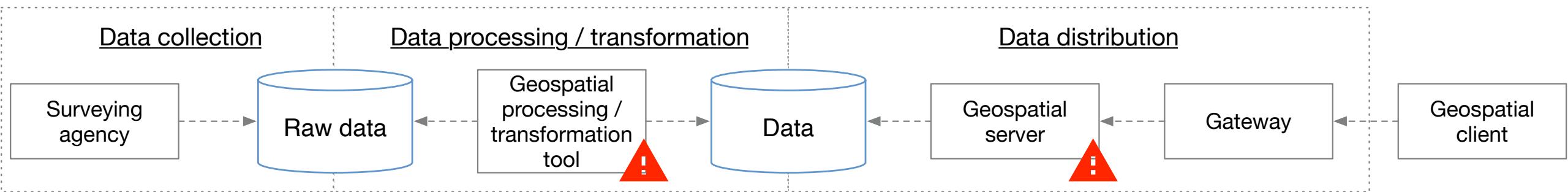
# Publishing data in an SDI

## What could possibly go wrong?



# Publishing data in an SDI

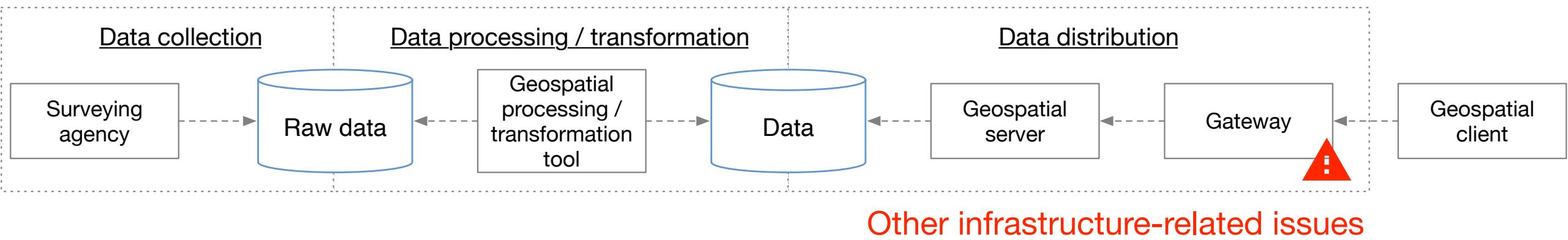
## What could possibly go wrong?



Software defects

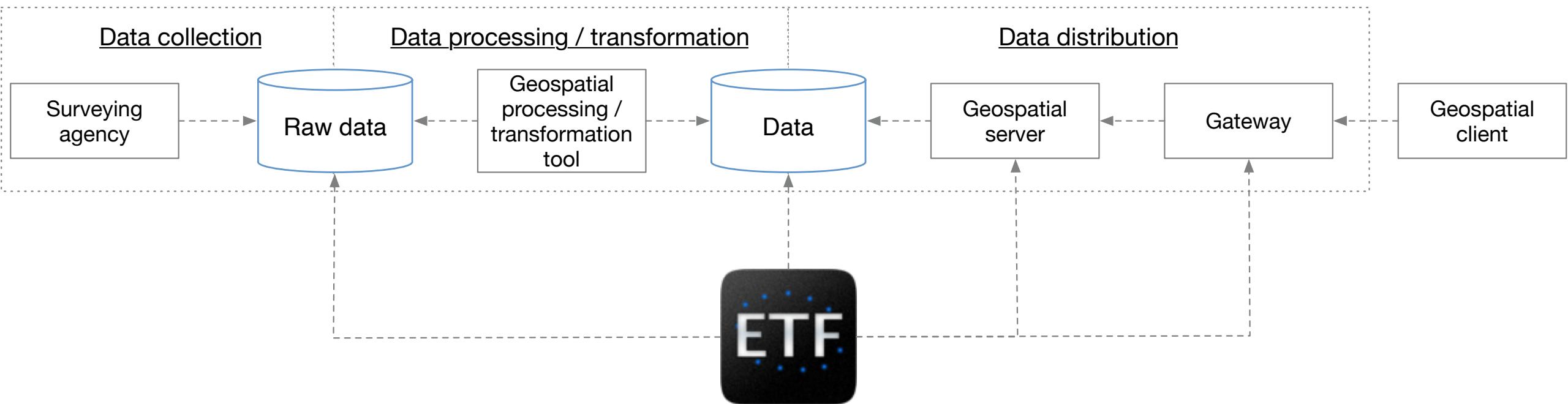
# Publishing data in an SDI

## What could possibly go wrong?



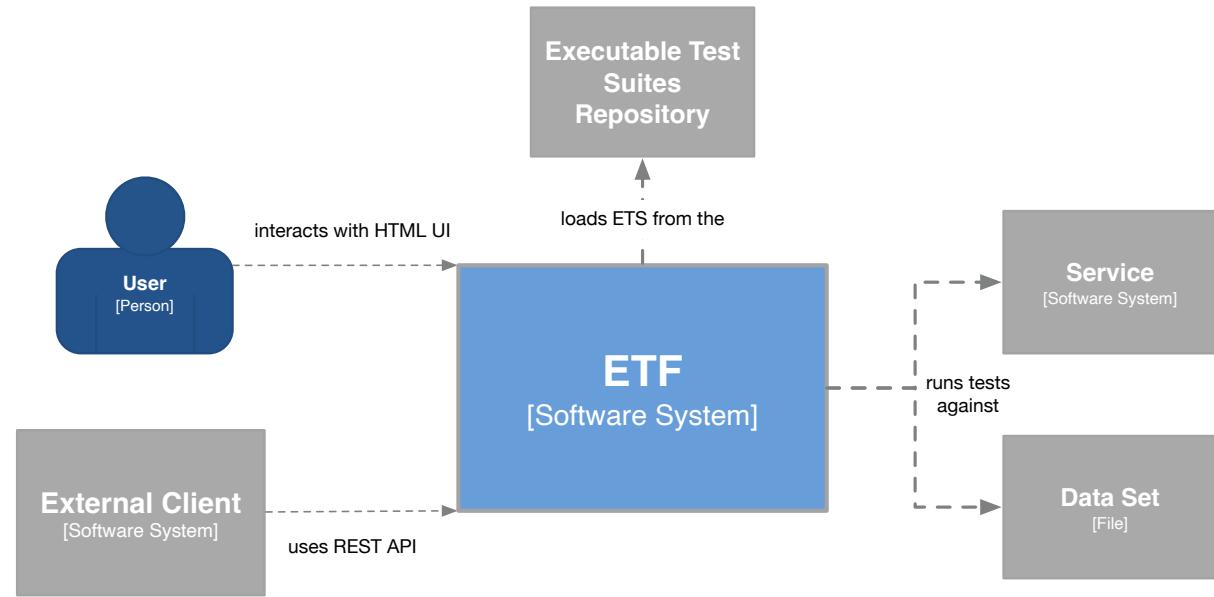
# **ETF testing framework**

## for spatial data, metadata & web services



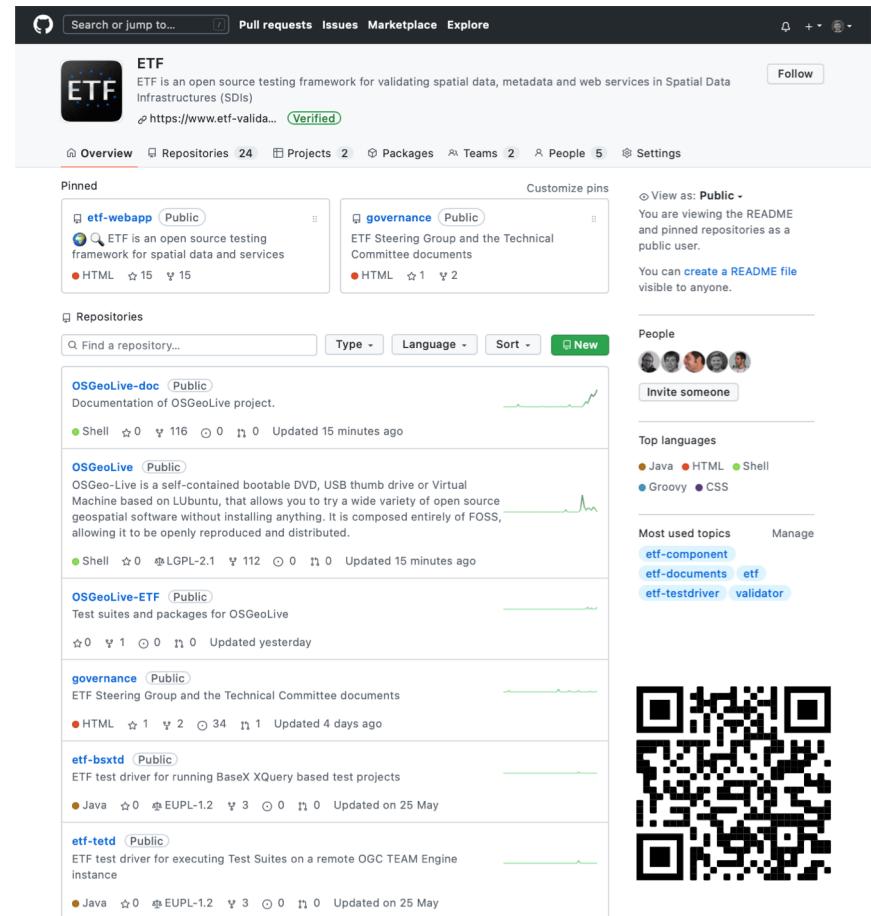
# ETF test framework

- An open source test framework to validate spatial data, metadata & web services in SDIs
  - developed since 2010
  - current version: 2.1.0
- Usable through a responsive web application and a REST API
- HTML, JSON or XML, CSV test reports
- Testable resources:
  - datasets (multiple hundreds GB): XML, GML, Shapefile, CSV
  - metadata: XML
  - view services: WMS / WMTS
  - download services: Atom, WFS, WCS, SOS
  - catalogue services: CSW



# Governance

- Open Source Project
  - European Union Public License 1.2 (EUPL)
  - Developer Certificate of Origin (DCO)
- Steering Group (SG):
  - voting system: consensus of all members
  - inaugural members: JRC, interactive instruments
  - ETF Improvement Proposals (EIPs)
- Technical Committee (TC):
  - members proven technical capacity over time
  - technical feedback on EIPs
  - merges Pull Requests
  - creates releases



<https://github.com/etf-validator/governance>

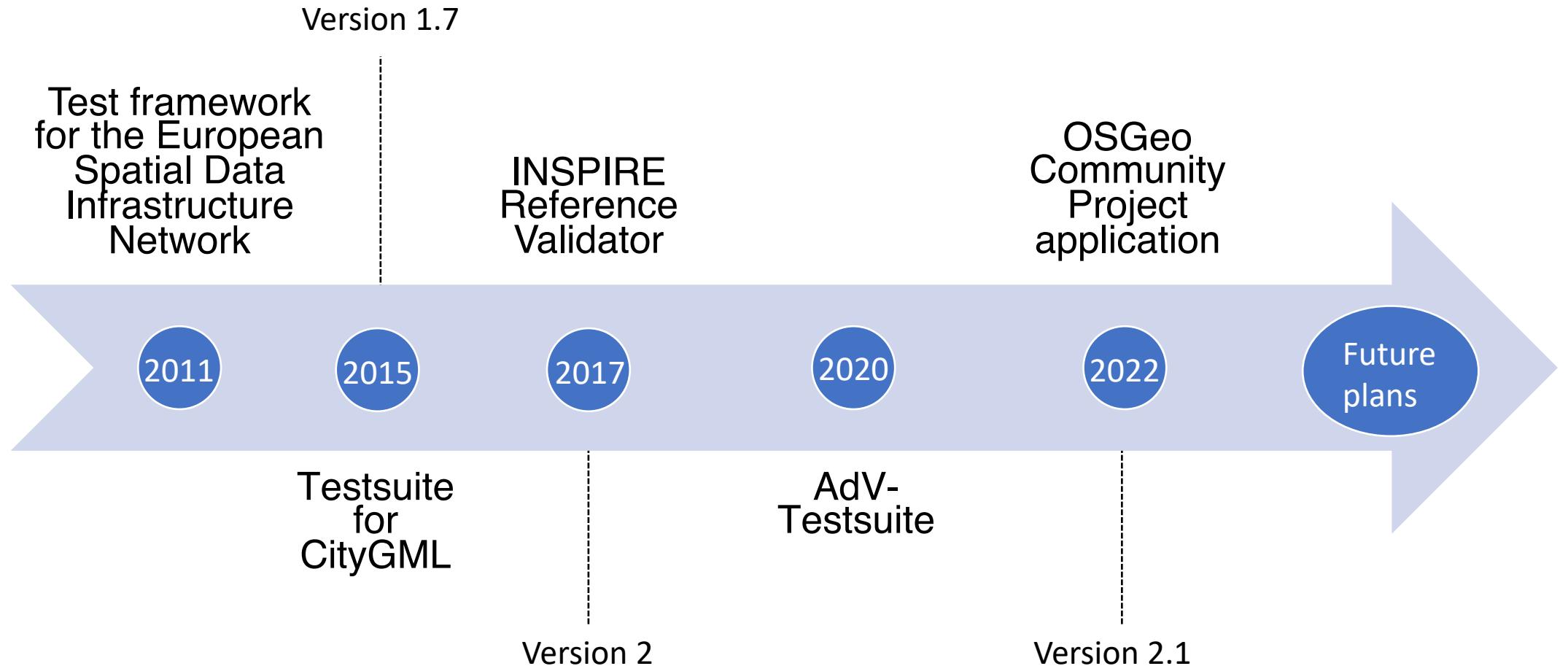
# ETF Improvement Proposals (EIPs)

ETF Improvement Proposals (EIPs)

Updated on 5 Jul

Proposer: draft/refine EIP	SG: discussion	TC: analysis	TC: in release planning	Implemented
<p>Test reports: Support printable version governance#31 opened by michellutz <span style="background-color: green; border-radius: 50%; padding: 2px 5px;">EIP-approved</span></p>	<p>New test driver for web service tests governance#77 opened by jonherrmann <span style="background-color: orange; border-radius: 50%; padding: 2px 5px;">EIP</span></p>	<p>Geometry orientation validation in GmlGeoX v2 governance#100 opened by carlospzurita <span style="background-color: orange; border-radius: 50%; padding: 2px 5px;">EIP-draft</span></p>	<p>Improve schema validation governance#49 opened by jonherrmann <span style="background-color: green; border-radius: 50%; padding: 2px 5px;">EIP-approved</span> <span style="background-color: purple; border-radius: 50%; padding: 2px 5px;">Impl.</span> Pre-processing in BaseX testdriver governance#93 opened by jonherrmann <span style="background-color: green; border-radius: 50%; padding: 2px 5px;">EIP-approved</span> <span style="background-color: purple; border-radius: 50%; padding: 2px 5px;">Impl.</span></p>	<p>Add Java Client to ETF validator organisation governance#87 opened by jonherrmann <span style="background-color: green; border-radius: 50%; padding: 2px 5px;">EIP-approved</span> <span style="background-color: purple; border-radius: 50%; padding: 2px 5px;">Impl.</span></p>
<p>Performance improvement for tests with small spatial test data sets [draft] governance#76 opened by jonherrmann <span style="background-color: orange; border-radius: 50%; padding: 2px 5px;">EIP-draft</span></p>	<p>Support additional test drivers in the etf-result-checker governance#99 opened by cportele <span style="background-color: orange; border-radius: 50%; padding: 2px 5px;">EIP-draft</span></p>	<p>Integration of API - Features on TEAM Engine Test Driver. governance#108 opened by carlospzurita <span style="background-color: orange; border-radius: 50%; padding: 2px 5px;">EIP</span></p>	<p>Invalid GML srsDimension on GmlGeoX v1 governance#102 opened by carlospzurita <span style="background-color: green; border-radius: 50%; padding: 2px 5px;">EIP-approved</span> <span style="background-color: purple; border-radius: 50%; padding: 2px 5px;">Impl.</span></p>	
<p>Use of BaseX with client-server architecture governance#75 opened by carlospzurita <span style="background-color: orange; border-radius: 50%; padding: 2px 5px;">EIP-draft</span></p>	<p>Support parallel execution of several ETSs in one test run governance#46 opened by michellutz <span style="background-color: orange; border-radius: 50%; padding: 2px 5px;">EIP-draft</span></p>	<p>Delete temporary test objects immediately governance#90 opened by cportele <span style="background-color: green; border-radius: 50%; padding: 2px 5px;">EIP-approved</span> <span style="background-color: purple; border-radius: 50%; padding: 2px 5px;">Impl.</span></p>	<p>Restructure repositories governance#70 opened by jonherrmann <span style="background-color: green; border-radius: 50%; padding: 2px 5px;">EIP-approved</span> <span style="background-color: purple; border-radius: 50%; padding: 2px 5px;">Impl.</span></p>	
<p>Support parallel execution of test steps/cases/modules inside an ETS governance#47 opened by michellutz <span style="background-color: orange; border-radius: 50%; padding: 2px 5px;">EIP-draft</span></p>	<p>Migrate to Java 11 governance#71 opened by jonherrmann <span style="background-color: green; border-radius: 50%; padding: 2px 5px;">EIP-approved</span> <span style="background-color: purple; border-radius: 50%; padding: 2px 5px;">Impl.</span></p>			
<p>Handling of low disk space governance#29 opened by michellutz <span style="background-color: orange; border-radius: 50%; padding: 2px 5px;">EIP-draft</span></p>				

# Important milestones of ETF



# Test framework for the European Spatial Data Infrastructure Network

2011

- Best Practice Network EU-project
- Help EU Member States to prepare for INSPIRE Directive
- Command line tool
- Used internally within organisations
- Executable Test Suites (**ETS**):
  - Web Feature Services
  - Web Map Services
- Junit XML reports

# Testsuite for CityGML 3D building models

- Let the Federal states of Germany validate their CityGML 3D building models
  - Validation up to 30 GB per state
  - Web UI
  - Used internally within organisations
  - Executable Test Suites:
    - CityGML data sets
    - Attributive data tests
  - PDF test report
- 
- Version 1.x: two different build configurations for services or for data sets



# **Version 2 (INSPIRE Validator) - Architecture**

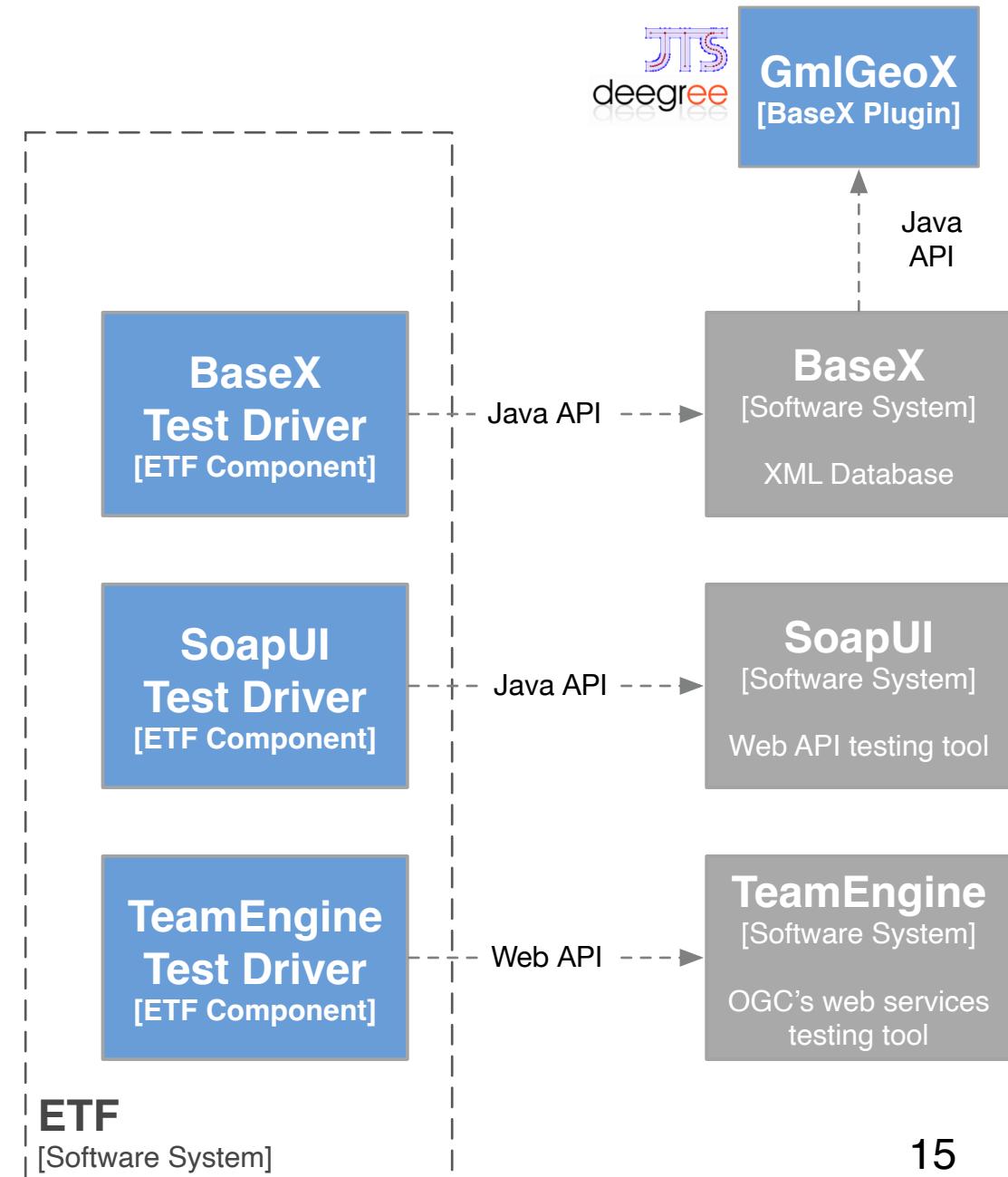
## Main design goals

- User-friendly and self-explanatory test reports
- Consistent with the standards (ISO/OGC)
- Capable of testing all resources in an SDI

# Version 2 - Architecture

## Test Drivers

- Test engines execute Executable Test Suites (ETS)
- Test Drivers are adapters which control, monitor and transform results into an internal model
- **BaseX** for testing sets of GML / XML documents
- **SoapUI** for testing web services
- **TEAM Engine** – the tool used by the OGC CITE tests
- GmlGeoX is a plugin to execute geometric queries



# INSPIRE Reference Validator

2017

- Design is basis for the current 2.x versions
- Custom UI to be in line with the visual identity of the European Commission (uses REST API)
- **Central cloud deployment**
- Option for local, organisation internal deployments
- Executable Test Suites:
  - INSPIRE metadata
  - Datasets (INSPIRE Annex I – III): attributive and **geometric data tests**
  - WMS, WFS, WCS, SOS, ATOM and OGC API - Features service tests
  - ~ 850 tests
  - ~ 200 test runs per day
- **⚠ Please note:**
  - Production instance still based on ETF Version 2.0.x
  - Version 2.1 update is planned for January 2023

The screenshot displays the 'INSPIRE Reference Validator - Test selection' interface. At the top, there's a navigation bar with links for Home, Test selection, Test reports, Get support, and More on the INSPIRE Reference Validator. Below this, a section titled 'Configure your test' allows users to select the INSPIRE resource type (Metadata, View Service, Download Service, Discovery Service, Data set) and the Technical Guidelines version (Version 1.3 - DEPRECATED, Version 2.0). Another section, 'Provide the resource to test', contains a file upload input field with a dropdown menu showing 'File upload' and 'Upload file'. Below the file input, there are instructions about file types and size limits, and a 'Choose files' button. At the bottom, there's a text input field for 'Provide a label for your test report (optional)' and a yellow 'Start test >' button.

<https://inspire.ec.europa.eu/validator>

# AdV-Testsuite

2020

- Used by Surveying Authorities of the German Laender to validate their datasets
- Decentralized Deployments with central ETS-Registry
- Executable Test Suites
  - Attributive tests
  - Geometrical tests
  - Topological tests
  - ~ 600 tests
- Challenging runtime requirements
- **New features merged into version 2.1**

# Version 2.1

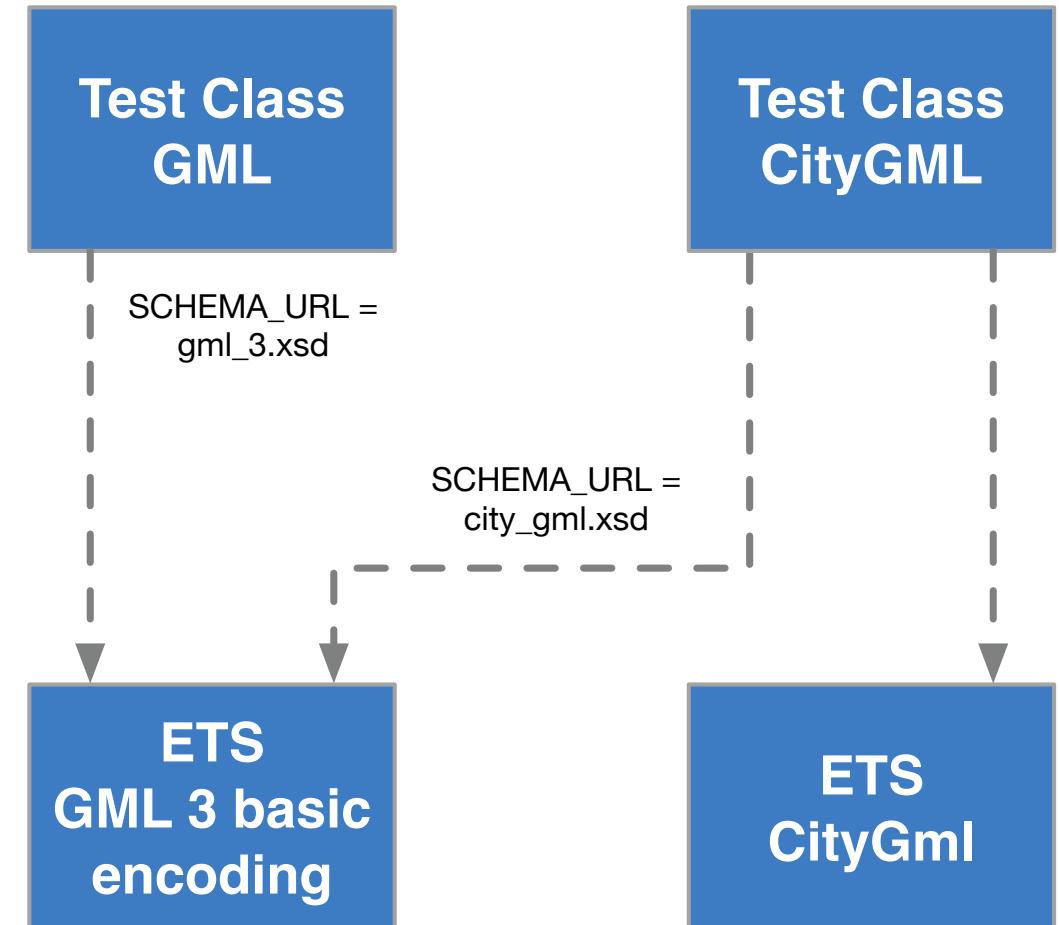
## Test Classes

Metadata (TG version 1.3)	
<input type="checkbox"/> +	Conformance class: INSPIRE Profile based on EN ISO 19115 and EN ISO 19119 <input type="checkbox"/> use
<input type="checkbox"/> +	Conformance class: Metadata for interoperability <input type="checkbox"/> use
Metadata (TG version 2.0) - BETA	
<input type="checkbox"/> +	Common Requirements for ISO/TC 19139:2007 based INSPIRE metadata records. <input checked="" type="checkbox"/>
<input type="checkbox"/> +	Conformance Class 1: INSPIRE data sets and data set series baseline metadata. <input checked="" type="checkbox"/>
<input type="checkbox"/> +	Conformance Class 2: INSPIRE data sets and data set series interoperability metadata. <input checked="" type="checkbox"/>
<input type="checkbox"/> +	Conformance Class 3: INSPIRE Spatial Data Service baseline metadata. <input type="checkbox"/> use
<input type="checkbox"/> +	Conformance Class 4: INSPIRE Network Services metadata. <input type="checkbox"/> use
<input type="checkbox"/> +	Conformance Class 5: INSPIRE Invocable Spatial Data Services metadata. <input type="checkbox"/> use
<input type="checkbox"/> +	Conformance Class 6: INSPIRE Interoperable Spatial Data Services metadata. <input type="checkbox"/> use

# Version 2.1

## Test Classes

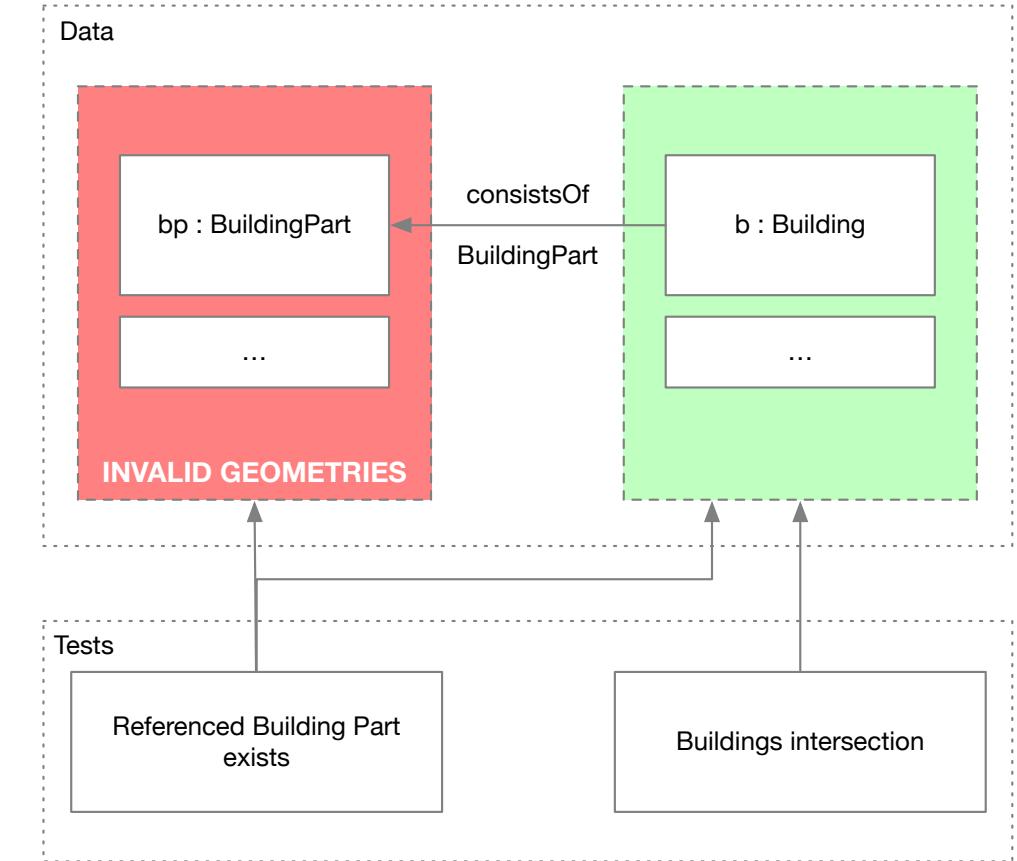
- Execution of a collection of Executable Test Suites in a test run
- Test Classes can pass parameters to generic ETSs to change test behaviour
- Improves:
  - **Reuse** of Executable Test Suites through parameterisation
  - **Usability**
- Example use case:  
URL for schema validation



# Version 2.1

## Isolated view on defective objects

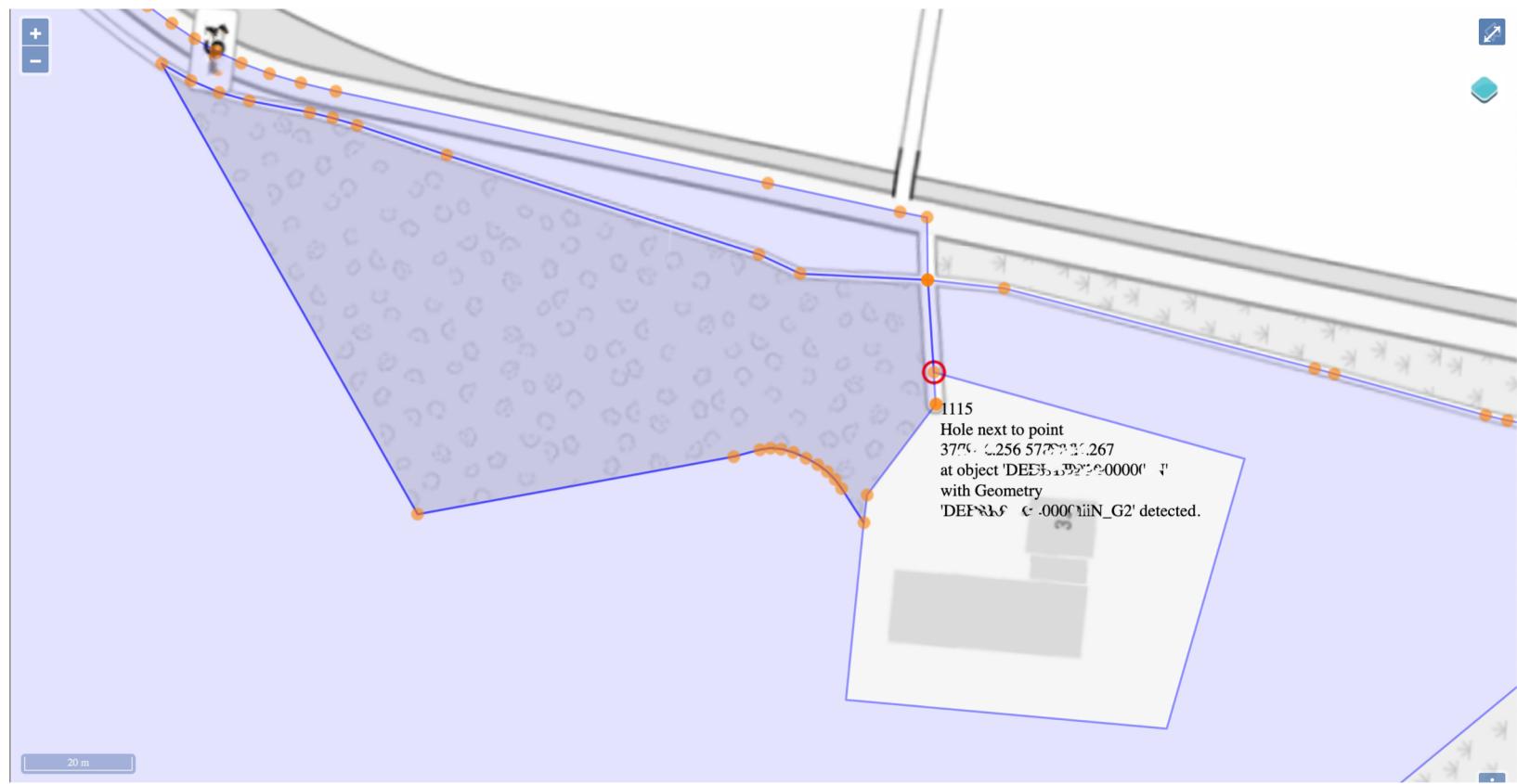
- Create isolated view on defective objects in pre-processing step
- Access the isolated objects for a specific set of tests
- E.g. isolate objects with invalid geometries but honour their ID when testing references
- Advantages:
  - No incomprehensible subsequent errors: reported only once, **keep report clean**
  - Improve test runtime



# Version 2.1

## TopoX

- Detects topological issues in surfaces:
  - Intersections and overlapping objects
  - Exclaves
  - Holes
  - ... 13 different error codes
- Runtime :
  - 20 GB data
  - 83.000 cadastral parcels
  - **gml:Arcs** and **gml:LineStringSegments**→ 8 minutes
- 20 GB without **gml:Arcs**  
→ 3 minutes
- System: 1 x vCore / 16 GB RAM / SATA SSD

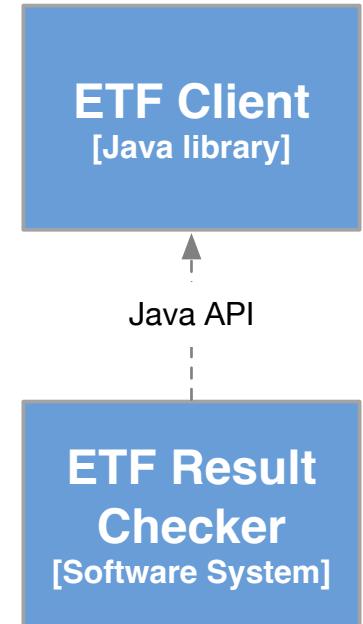


```
1 let $init := topoX:parse-surface($features, '*:position/gml:Surface')
2 let $holeCount := topoX:detect-holes($topoId)
3 let $freeStandingSurfaceCount := topoX:detect-free-standing-surfaces($topoId)
4 let $errorCodes := ('RING_OVERLAPPING_EDGES', 'RING_INTERSECTION',
5   'HOLE_EMPTY_INTERIOR', 'FREE_STANDING_SURFACE')
6 let $allErrors := topoX:topological-errors($errorCodes)
```

# Version 2.1

## ETF result checker

- Tool for comparing the actual result with an expected result (a.k.a. the test tool for ETF)
- **Designed for regression testing within pipelines**
- Inputs:
  - JSON control file
  - JSON file with expected results (can be generated on the first run)
  - For data tests: data file to test (ZIP or XML)
- Output formats:
  - Junit XML
  - Allure Reports



# OSGeo Community Project

- Application submitted – currently under evaluation
- Project page on the OSGeo website created as part of the process

The screenshot shows the OSGeo website's project page for ETF. At the top, there is a navigation bar with links for News, Wiki, Contact, Sign in, Projects, Resources, About OSGeo, Initiatives, and Community. Below the navigation is a large green header section for the ETF project. The header includes the OSGeo logo, a back-to-projects link, and a large image of the ETF logo. A "Visit our website" button is also present. The main content area contains a brief description of what ETF is, followed by a detailed description of its goals and supported test engines.

Home » Projects » ETF

# ETF

ETF is a testing framework for validating data and APIs in Spatial Data Infrastructures (SDIs). It is used by software solutions and data providers to validate the conformity of geospatial data sets, metadata and APIs.

Goals in designing the ETF software were to create test reports that are user-friendly and self-explanatory as well as to be able to validate large amounts of data, which can be several hundred GB in size. In order to cover different validation tasks and present them in a unified report, the architecture is modular and different test engines can be used. Currently the following test engines are supported: [SoapUI](#) for testing web services, [BaseX](#) database for testing XML data, [TEAM Engine](#) to validate WFS and OGC Web APIs using the OGC CITE tests, NeoTL Engine for testing WFS, OGC Web APIs and datasets.

ETF is the underlying framework used by the [INSPIRE Reference Validator](#) to validate metadata, datasets and services against the [INSPIRE](#) requirements. ETF is also used extensively in Germany by the Surveying Authorities of the Laender to validate their datasets. Other European Union (EU) Member States are also reusing the ETF to allow their data providers to test resources against national requirements. Finally, some software tools include validation based on the ETF API in their workflow.

[https://www.osgeo.org/  
projects/etf](https://www.osgeo.org/projects/etf)



Test run on 18:14 - 31.05.2022 with test suite Common conformance classes					
Status	Failed	Total	Count	Skipped	Failed
Started	31/05/2022 16:14:47 GMT	Test suites	5	0	2
	Duration 1.357 s	Test cases	12	0	2
		Assertions	36	0	2
					4

+ Conformance class: INSPIRE GML encoding [1]  
+ Conformance class: Reference systems, General requirements [Failed 1/2]  
+ Conformance class: Information accessibility, General requirements [Failed 1/1]  
+ Conformance class: INSPIRE GML application schemas, General requirements [4]  
+ Conformance class: Data consistency, General requirements [2]

Report generated by ETF

# OSGeo

## OSGeoLive

- Inclusion in OSGeoLive Version 15
- Contains:
  - ETF version 2
  - Related documentation
  - Sample ETS



[Home](#) [Contents](#) [Download](#) [Metrics](#) [Sponsors](#) [Contact Us](#)

[English](#) | [Deutsch](#) | [Español](#) | [Suomen kieli](#) | [Français](#) | [Hungarian](#) | [Italiano](#) | [日本語](#)

### ETF

ETF is an open source testing framework for validating data and APIs in Spatial Data Infrastructures (SDIs). It is used by software solutions and data providers to validate the conformity of geospatial data sets, metadata and APIs.

#### About

Goals in designing the ETF software were to create test reports that are user-friendly and self-explanatory as well as to be able to validate large amounts of data, which can be several hundred GB in size. In order to cover different validation tasks and present them in a unified report, the architecture is modular and different test engines can be used. Currently the following test engines are supported: SoapUI for testing web services, BaseX database for testing XML data, TEAM Engine to validate WFS and OGC Web APIs using the OGC CITE tests, NeoTL Engine for testing WFS, OGC Web APIs and datasets.

ETF is the underlying framework used by the INSPIRE Reference Validator to validate metadata, datasets and services against the INSPIRE requirements. ETF is also used extensively in Germany by the Surveying Authorities of the Laender to validate their datasets. Other European Union (EU) Member States are also reusing the ETF to allow their data providers to test resources against national requirements. Finally, some software tools include validation based on the ETF API in their workflow.

#### Core Features

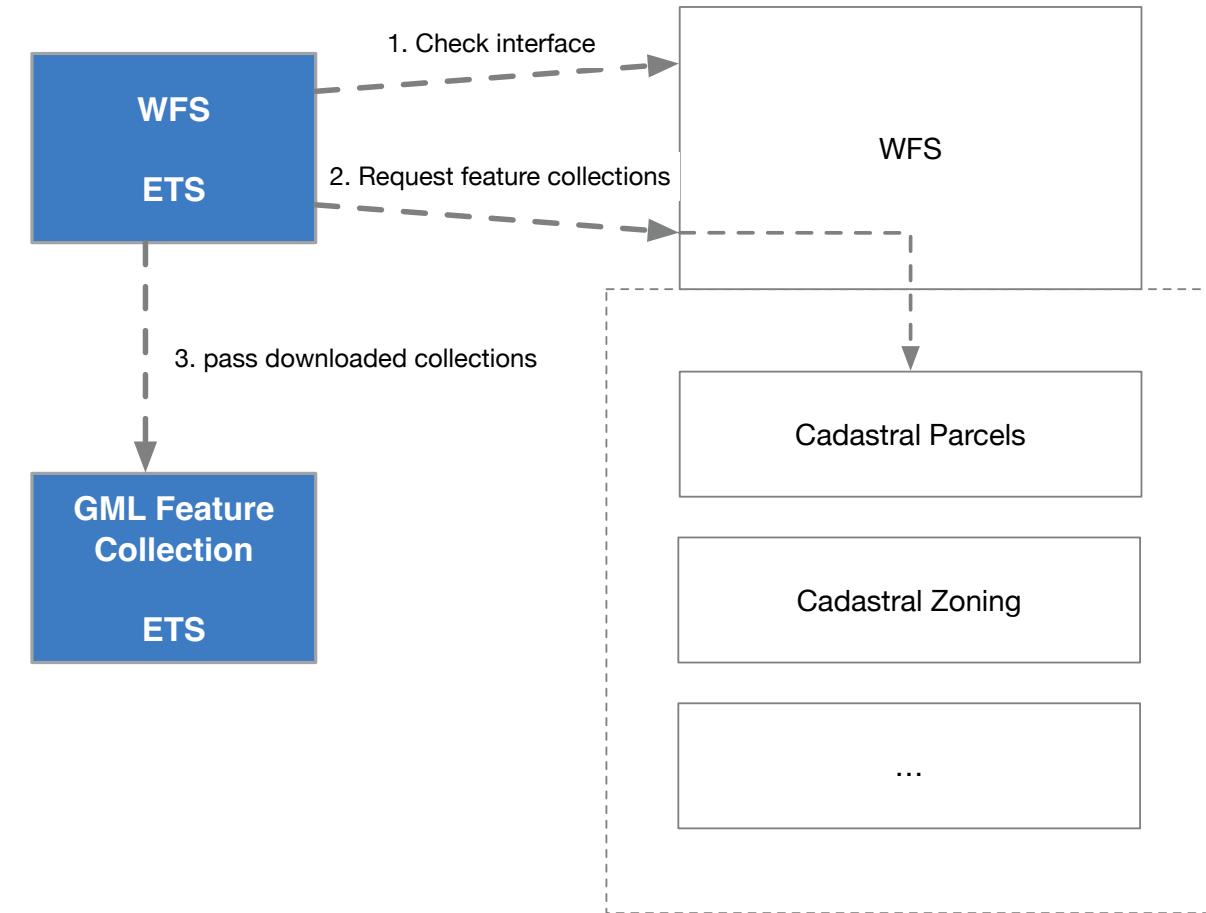
- Testable resources
  - datasets (up to multiple hundreds GB): GML
  - metadata: XML
  - view services: WMS/WMTS
  - download services: Atom, WFS, WCS, SOS
  - catalogue services: CSW

A screenshot of the ETF software's user interface. At the top, there are navigation buttons for 'Start test', 'Status', 'Test reports', and 'Help'. Below this is a search bar labeled 'Filter Executable Test Suites...'. A list of executable test suites is displayed, each with a checkbox labeled 'use'. The items listed are: 'Download Services (Technical Guidance version 3.1)', 'Conformance Class: Download Service - Pre-defined Atom', 'Conformance Class: Download Service - Pre-defined WFS', 'Metadata (Technical Guidance version 1.3)', 'Conformance class: INSPIRE Profile based on EN ISO 19115 and EN ISO 19119', 'Conformance class: Metadata for interoperability', 'Interoperable data sets in GML (Guidelines for the Encoding of Spatial Data version 3.3)', and 'Conformance class: Data consistency, General requirements'.

# Future Goals

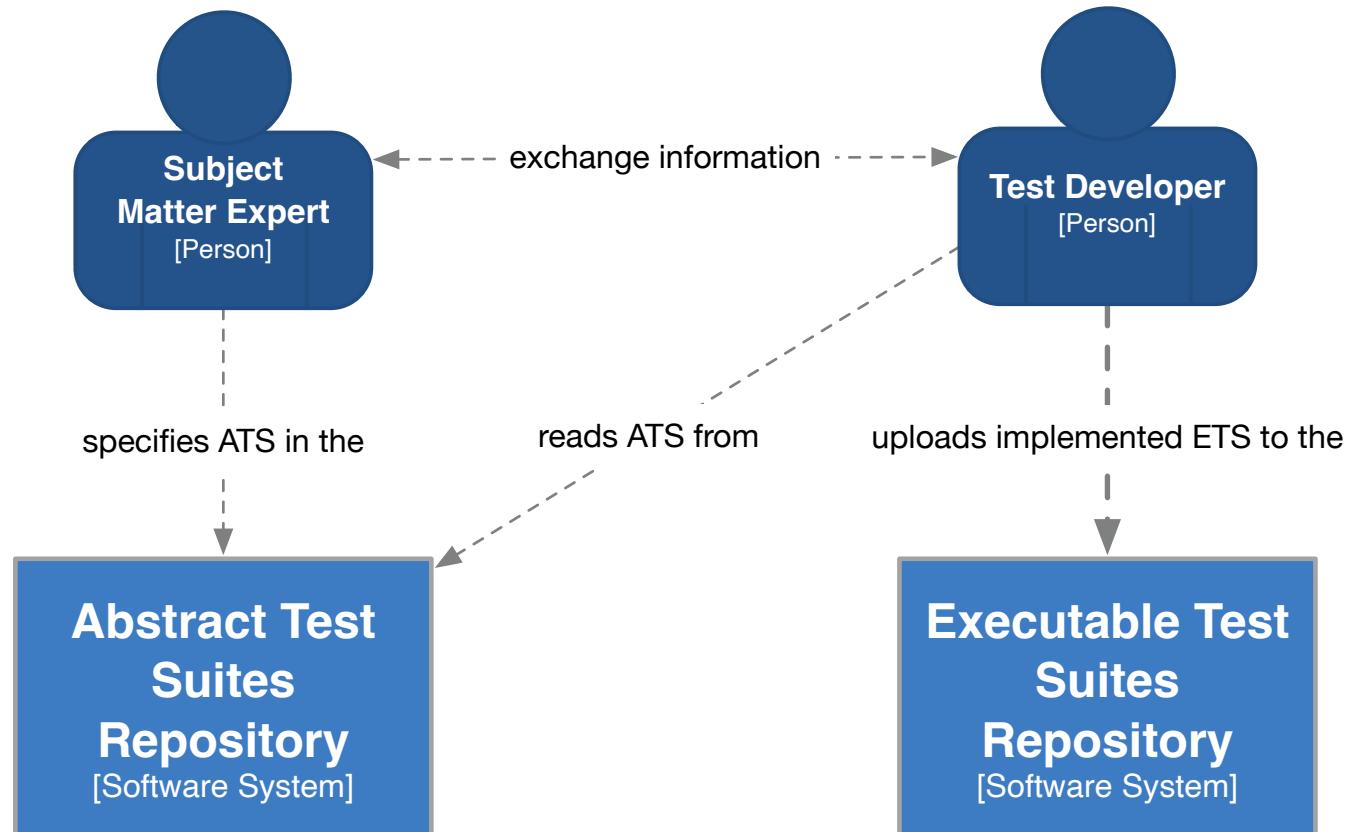
## Cross-driver Execution

- Close the gap between testing the server interface and the served data
- One ETS defines interface tests and which data are retrieved via the operations
- The data is then checked in a downstream ETS



# Future Goals

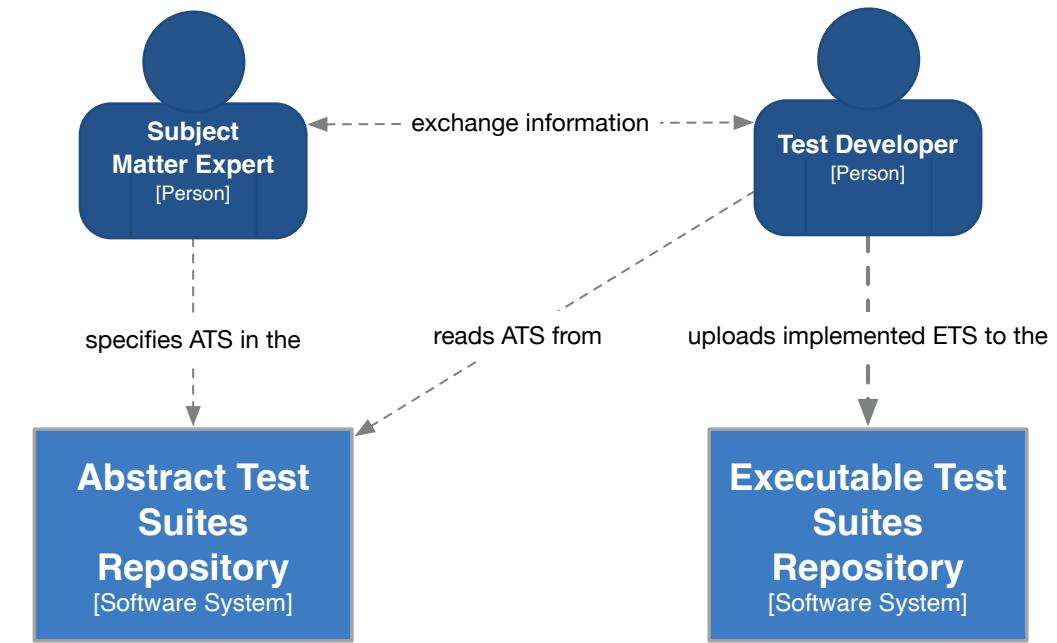
## Facilitate test development



# Future Goals

## Facilitate test development

- Improve communication between Subject Matter Experts and Test Developers
- Subject Matter Experts without deep Programming Language knowledge should implement simple tests and change existing ones
- Improve modularization of tests
- Provide a ready to use IDE and assist the test developer



# Domain Specific Language: NeoTL

- OGC Testbed 17
  - NeoTL
  - ETS for the OGC API - Processes standard
  - Plugin for Visual Studio Code
  - New test driver
  - Engineering Report
- DSL model structure is based on ISO 19105 Geographic information – Conformance and testing



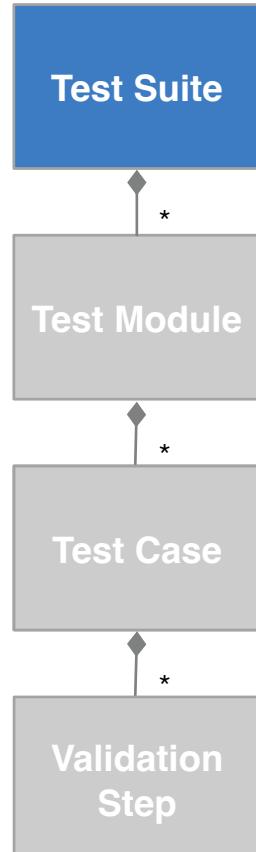
[https://docs.ogc.org/  
per/21-044.html#toc12](https://docs.ogc.org/per/21-044.html#toc12)



# NeoTL structure

## Test Suite

```
TestSuite "OGC API - Processes" {
    id: org.opengis.ets.ogcapi.processes
    version: 0.9.2-snapshot
    description: "Executable Test Suite for validating Web APIs that implement
                  the 'OGC API - Processes - Part 1: Core' standard. The Test Suites
                  are based on the normative Abstract Test Suites from Annex A of the
                  OGC Implementation Specification draft 1.0-draft.7-SNAPSHOT."
    references:
        - "OGC Implementation Specification"
          "https://docs.ogc.org/DRAFTS/18-062.html"
          ImplementationSpecification
    executes:
        - oapi.processes.core
        - oapi.processes.joblist
        // ...
    defines:
        - URL $schemaUrl = "https://raw.githubusercontent.com/opengeo/ogcapi-processes/master/core/openapi/schemas"
}
```



# NeoTL structure

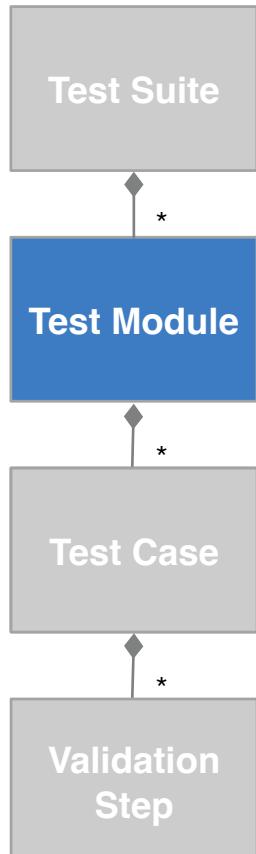
## Test Module

```
TestModule "Job List" {
    id: oapi.processes.joblist

    description: "The Job list requirements class specifies how to retrieve a job list from the API"

    references:
        - "Core ATS"
            "http://docs.ogc.org/DRAFTS/18-062.html#_conformance_class_core"
            AbstractTestSuite
        - "Conformance Class Job list"
            http://www.opengis.net/spec/ogcapi-processes-1/1.0/conf/job-list
            ConformanceClass

    executes:
        - joblist.list
        - joblist.list.links
        // ...
}
```



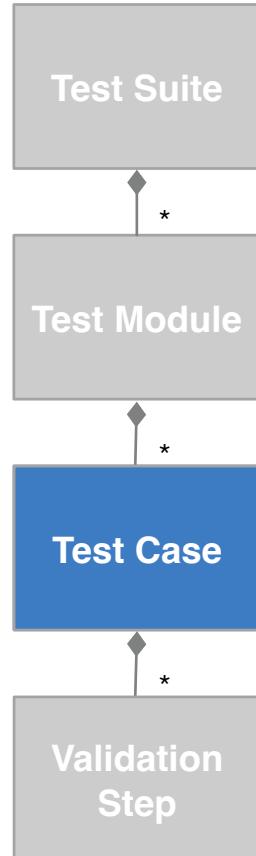
# NeoTL structure

## Test Case

```
TestCase "Validate links section in job list" {
    id: joblist.list.links
    description: "Validate that the proper links are included in a response."
    references:
        - "Conformance Class Job list"
          "https://docs.ogc.org/DRAFTS/18-062.html#_conformance_class_job_list"
        AbstractTestCase

    ValidationStep "Validate self link" {
        // ...
    }

    ValidationStep "Validate HTML link" {
        // ...
    }
}
```



# NeoTL structure

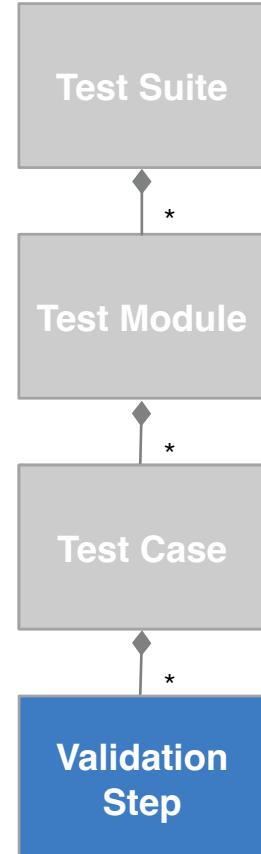
## Validation Step

```
ValidationStep "Request the Landing Page as JSON document" {
    id: step
    description: "Request an JSON document with an Accept application/json header"

    given:
        - ConformanceClass http://www.opengis.net/spec/ogcapi-processes-1/1.0/conf/json

    when: Request requests.landing.json executed

    then:
        - Assert OpenAPI3 {
            schema "${schemaUrl}/landingPage.yaml"
            validates
        }
        - Assert HTTP { statusCode "200" }
        - Assert HTTP { contentType "application/json" }
}
```



# NeoTL

## Example assertions

```
- Assert JSON {
    $.jobs[?(@.type == 'process')] exists
    or $.jobs empty
    otherwise FAIL with "No job entry exists with the
        requested process type 'process'."
}
```

```
- Assert JSON {
    $.jobs count == 1
    otherwise FAIL with "Expected exactly one entry in the Job List"
}
```

```
- Assert JSON {
    $.conformsTo contains some
        "http://www.opengis.net/spec/ogcapi-processes-1/1.0/conf/core"
    otherwise FAIL with "Expected an 'conformsTo' array containing
        the value
        'http://www.opengis.net/spec/ogcapi-processes-1/1.0/conf/core'."
}
```

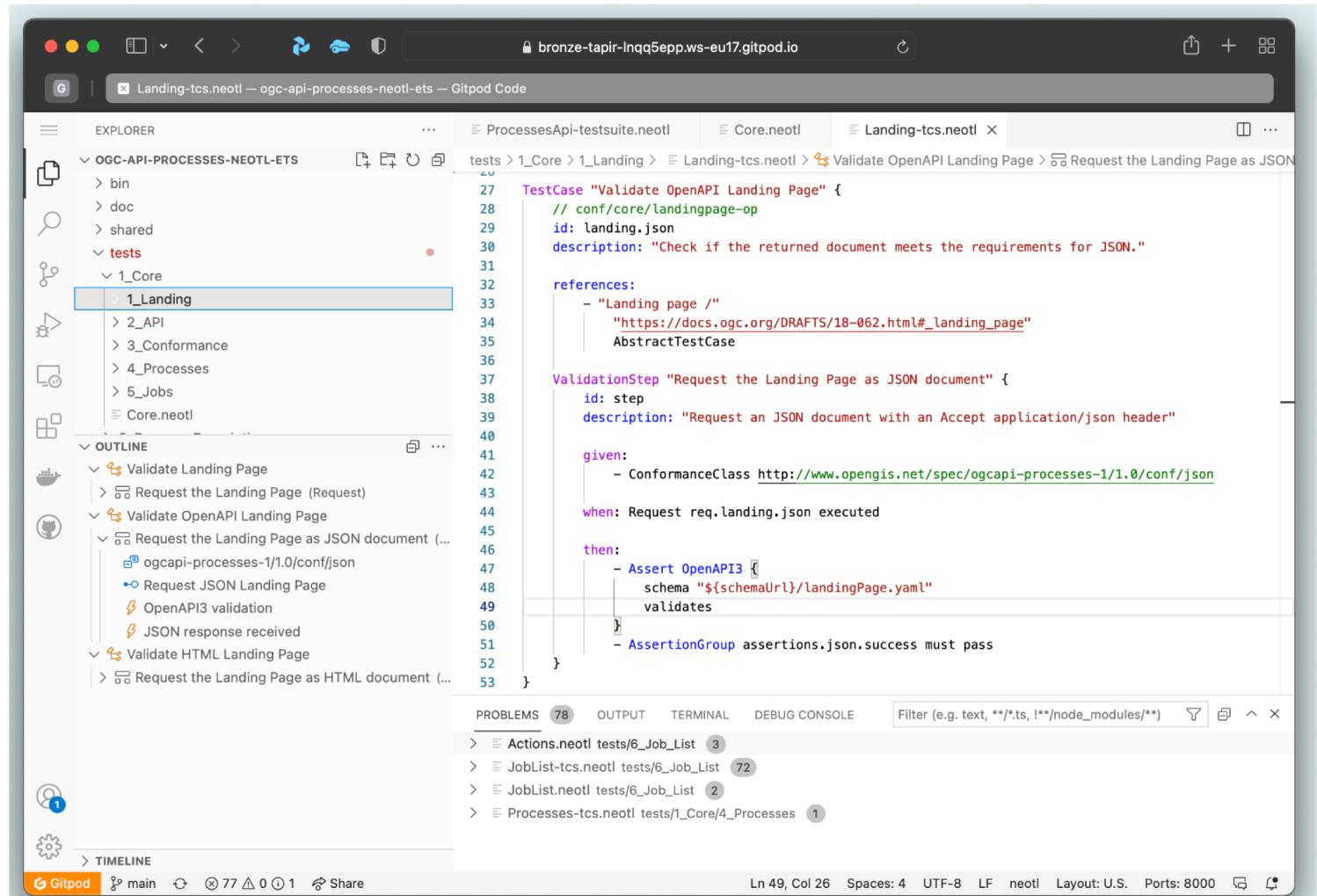
```
AssertionGroup "JSON response received" {
    id: assertions.json.success

    assertions:
        - Assert HTTP { statusCode "200" }
        - Assert HTTP { contentType "application/json" }
}
```

# NeoTL IDE assistance

- Tooltips and help in the cloud IDE
- Problem support e.g. broken Links
- Outline views of the Test structure
- Runs on gitpod.io

 Open in Gitpod



The screenshot shows the NeoTL IDE running on a Gitpod instance. The interface includes:

- EXPLORER**: Shows the project structure under "OGC-API-PROCESSES-NEOTL-ETS". The "tests/1\_Core/1\_Landing" folder is selected.
- CODE EDITOR**: Displays a JSON test case for validating an OpenAPI landing page. The code includes annotations for references, steps, given conditions, and assertions.
- OUTLINE**: Provides an outline view of the test cases and their steps.
- TERMINAL**: Shows the command "gitpod" entered.
- PROBLEMS**: A panel showing 78 problems across various files.
- TIMELINE**: A history of recent actions.

# Key takeaways

- New release 2.1 is available on GitHub
- OSGeoLive 15 will use the version 2.0
- INSPIRE Reference Validator update to 2.1 is planned for January 2023
- NeoTL Test Language will facilitate writing tests & will replace SoapUI
- Cross-driver execution will allow to test services including the served data
- **Community development around ETF – get involved if interested!**

# Thank you !

## Links to:

- **Contact Steering Group** on GitHub Discussions
- OSGeoLive release candidate
- ETF-Client library **for integrators**
- Installation manual
- Developer documentation
- GitHub organisation

<https://jonherrmann.github.io/foss4g>



## Contact Jon:

[herrmann@interactive-instruments.de](mailto:herrmann@interactive-instruments.de)

# **Backup**

# Version 2

## ETF result checker

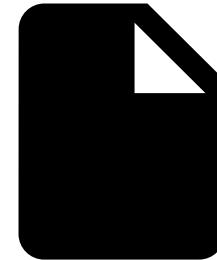
run.json

```
1 {  
2   "executableTestSuiteName": "Konformitätsklasse NAS Encoding",  
3   "arguments": {  
4     "model": "Basisschema",  
5     "specification": "GeoInfoDok_6.0.1"  
6   }  
7 }
```

expected  
results.json

```
1 {  
2   "DE.00001.A.a.012 - FF": {  
3     "expectedResult": "PASSED",  
4     "expectedMessageCount": 0  
5   },  
6   "DE.00001.A.a.013 - FF": {  
7     "expectedResult": "NOT_APPLICABLE",  
8     "expectedMessages": [  
9       "Due to the parameter ''specification'', the test is not applicable to the test data."  
10     ]  
11   },  
12   "DE.00001.V.a.001 - NA": {  
13     "expectedResult": "FAILED",  
14     "expectedMessages": [  
15       "The element type \"gml:Curve\" must be terminated by the matching end-tag"  
16     ]  
17   }  
18 }  
19
```

+



Test\_Data.zip

# Version 2

## Test Classes View

AdV-Testsuite (Prod-Umgebung)

Test Classes Test reports Test objects Executable Test Suites Status

### Test Classes

By clicking on the name, additional information on the respective test class and a button to start the test are displayed. In addition to the test classes available here, a local test class can also be executed by clicking on the right-hand button "Use local Test Class".

Use local Test Class

Zum Testklassenkonfigurator

Search Test Class by name or other properties...

- + Bestandsdaten AFIS (AdV) - AAA-AS 7.1
- + Bestandsdaten AFIS (AdV) - GID 6.0.1
- + Bestandsdaten ALKIS (AdV) - AAA-AS 7.1
- + Bestandsdaten ALKIS (AdV) - GID 6.0.1
- + Bestandsdaten Basis-DLM (AdV) - AAA-AS 7.1
- + Bestandsdaten Basis-DLM (AdV) - GID 6.0.1
- + Bestandsdaten DLM50 (AdV) - AAA-AS 7.1
- + Bestandsdaten DLM50 (AdV) - GID 6.0.1

ETF WebApp 2.1.0-b220530T0651 © 2017-2021 European Union, interactive instruments GmbH

39

# Version 2

## Test Classes - locked model parameters

**Configure Test Run**

Label: Bestandsdaten AFIS (AdV) - GID 6.0.1

Data source: File upload

Local geospatial files: Choose Files no files selected

**Test Suite parameters**

Modell:

- Basisschema
- DLKM
- Basis-DLM
- DLM50
- DFGM

Spezifikation:

- GeolInfoDok\_6.0.1 (selected)
- GeolInfoDok\_7.0.3
- GeolInfoDok\_7.1.0

AAA-AS\_7.1

Testet Multi Instanziell: Yes

Schema: DE/Ressourcen/schemas/adv/nas/6.0/aaa.xsd

**Selected Test Suites**

- Konformitätsklasse NAS Encoding (Version 1.0.0)
- Konformitätsklasse Basisschema (Version 1.0.0)
- Konformitätsklasse Fachschema (Version 1.0.0)
- Konformitätsklasse Festpunkte (Version 1.0.0)
- Konformitätsklasse Punktorte (Version 1.0.0)

Executable Test Suites Credentials Optional Parameters Start

**Configure Test Run**

Label: Bestandsdaten ALKIS (AdV) - GID 7.1.0

Data source: File upload

Local geospatial files: Choose Files no files selected

**Test Suite parameters**

Modell:

- Basisschema
- DLKM
- Basis-DLM
- DLM50
- DFGM

Spezifikation:

- GeolInfoDok\_6.0.1
- GeolInfoDok\_7.0.3
- GeolInfoDok\_7.1.0 (selected)

AAA-AS\_7.1

Testet Multi Instanziell: Yes

Schema: DE/Ressourcen/schemas/adv/nas/7.1/aaa.xsd

**Selected Test Suites**

- Konformitätsklasse NAS Encoding (Version 1.0.0)
- Konformitätsklasse Basisschema (Version 1.0.0)
- Konformitätsklasse Fachschema (Version 1.0.0)
- Konformitätsklasse Bauwerke, Einrichtungen und sonstige Angaben (Version 1.0.0)
- Konformitätsklasse Bodenschätzung, Bewertung (Version 1.0.0)
- Konformitätsklasse Eigentümer (Version 1.0.0)
- Konformitätsklasse Flurstücke (Version 1.0.0)
- Konformitätsklasse Gebäude, Bauwerke, Einrichtungen, Anlagen und Gestaltung 3D (Version 1.0.0)
- Konformitätsklasse Gebäude (Version 1.0.0)
- Konformitätsklasse Gebietseinheiten (Version 1.0.0)
- Konformitätsklasse Kataloge (Version 1.0.0)
- Konformitätsklasse Lage (Version 1.0.0)
- Konformitätsklasse Migration (Version 1.0.0)
- Konformitätsklasse Mitteilungsdaten (Version 1.0.0)
- Konformitätsklasse Öffentlich-rechtliche Festlegungen (Version 1.0.0)
- Konformitätsklasse Punktorte (Version 1.0.0)
- Konformitätsklasse Relief (Version 1.0.0)
- Konformitätsklasse Tatsächliche Nutzung (Version 1.0.0)

Executable Test Suites Credentials Optional Parameters Start

# Version 2 - Architecture

## Example ETS

```
● ● ●

1 import module namespace ggeo='de.interactive_instruments.etf.bsxm.GmlGeoX';
2
3 (: 
4 verify for each TransportNode that the geometry (a gml:Point) is located at a position that touches
  a TransportLink.centrelineGeometry (a gml:LineString or gml:Curve), i.e. that the node is at the
  start or end of a transport link. Otherwise report freeNode.
5 : )
6
7 let $candidate_geometry := ggeo:getGeometry($candidate/@gml:id,$candidate/*:geometry/*[1])
8 let $env := ggeo:envelope($candidate_geometry)
9 let $other_features := ggeo:search($env[1],$env[2],$env[3],$env[4])
10
11 let $touches :=
12   for $feature in
13     $other_features[(self::*:RoadLink|self::*:RailwayLink|self::*:CablewayLink|self::*:WaterwayLink|self
14     ::*:AirRouteLink|self::*:ProcedureLink|self::*:StandardInstrumentDeparture|self::*:InstrumentApproac
15     hProcedure|self::*:StandardInstrumentArrival)] return
16     let $other_geometry := ggeo:getGeometry(
17       $feature/@gml:id,$feature/*:centrelineGeometry/*[1]
18     )
19     return if (ggeo:touches($candidate_geometry,$other_geometry)) then $feature else ()
20   )
21
22 return if (not($touches)) then $candidate else ()
```