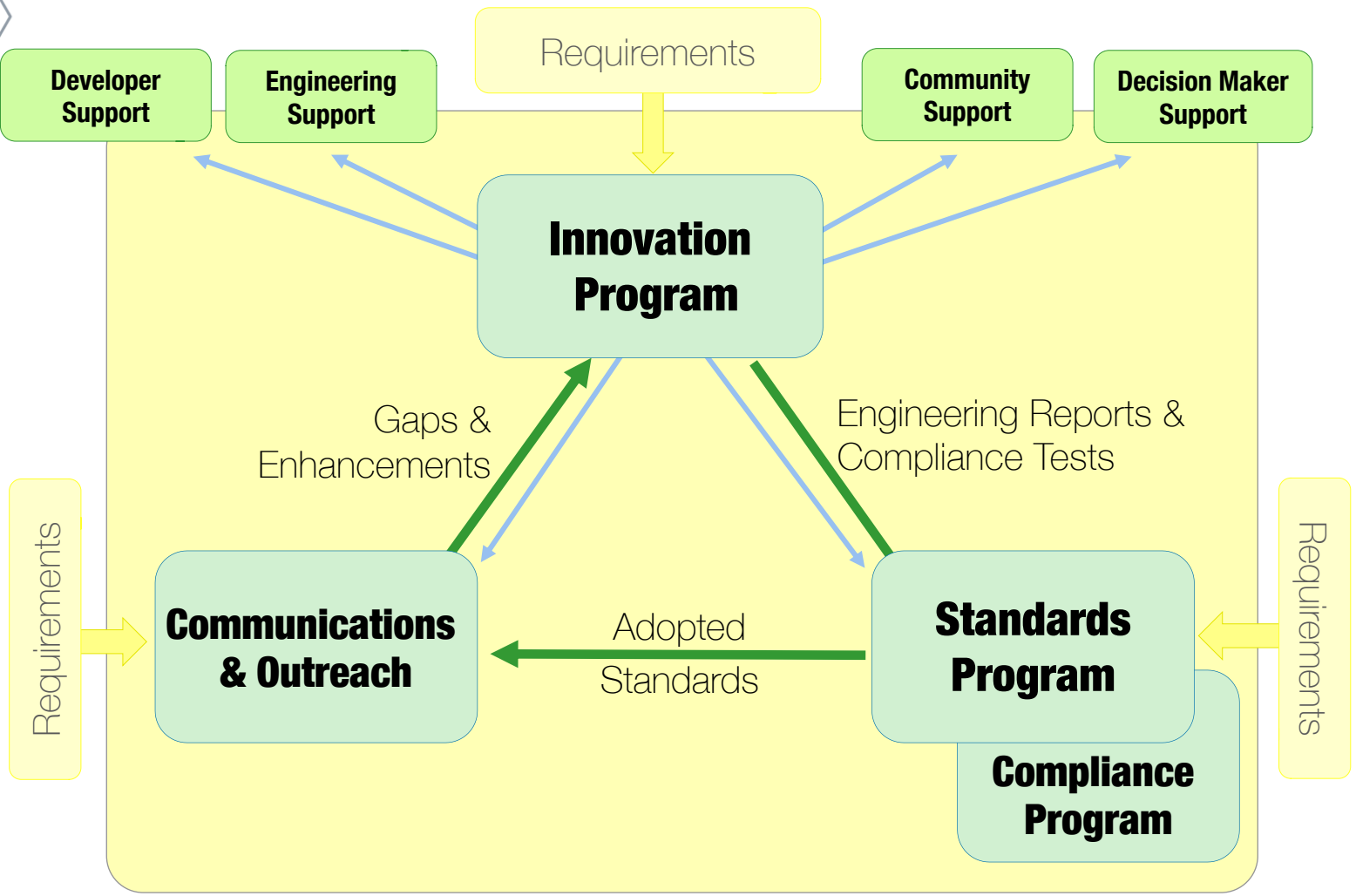
OGC Programs

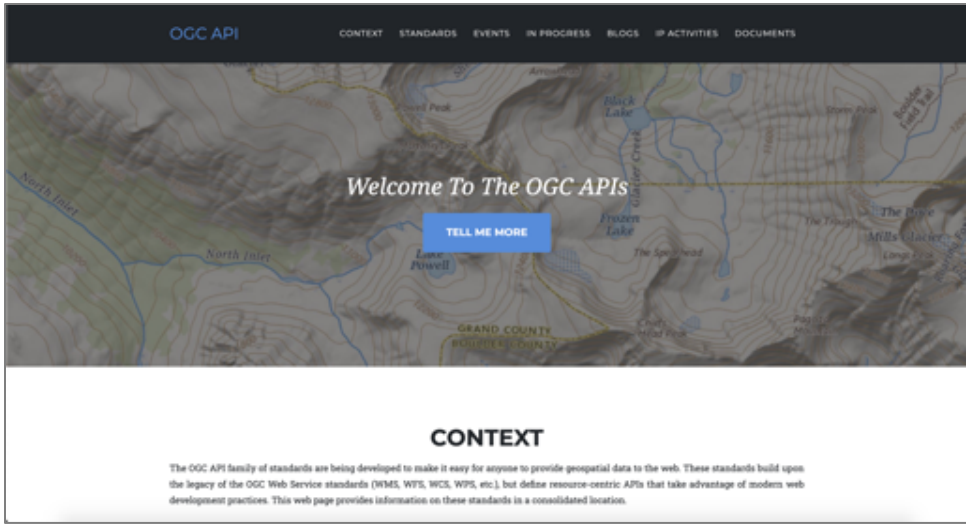


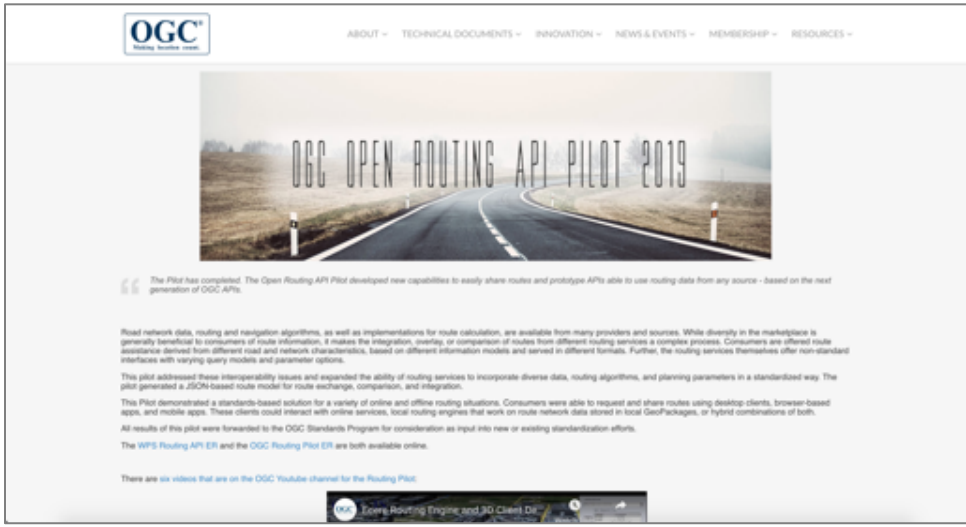
OGC®

Context

- New family of OGC API standards driven by Spatial Data on the Web Best Practices, OpenAPI, …
  - http://ogcapi.org/

- Routing API Pilot from May to September 2019
  - https://www.ogc.org/projects/initiatives/routingpilot - Overview page with demonstration videos
  - http://docs.opengeospatial.org/per/19-041r3.html - Summary report of the pilot including the specification of the Route Exchange Model
  - http://docs.opengeospatial.org/per/19-040.html - Draft specification of the Routing API





OGC®

Sponsor

AIRBUS

Open Routing API

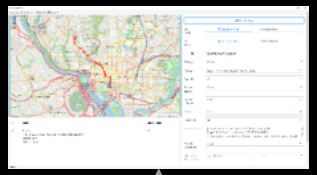
113th OGC Technical Committee
Toulouse, France
Clemens Portele
20 November 2019

OGC®
Making location count.

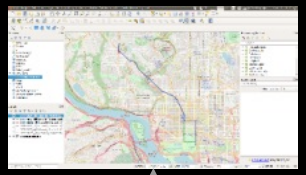
Open Routing API and Route Exchange Model…

Helyx

GIS-FCU



Skymantics



Ecere

Routing Applications

Route Exchange Model

Open Routing API

WPS (52North)	**WPS** (Skymantics)	**WPS** (Helyx)	**WPS** (GIS-FCU)

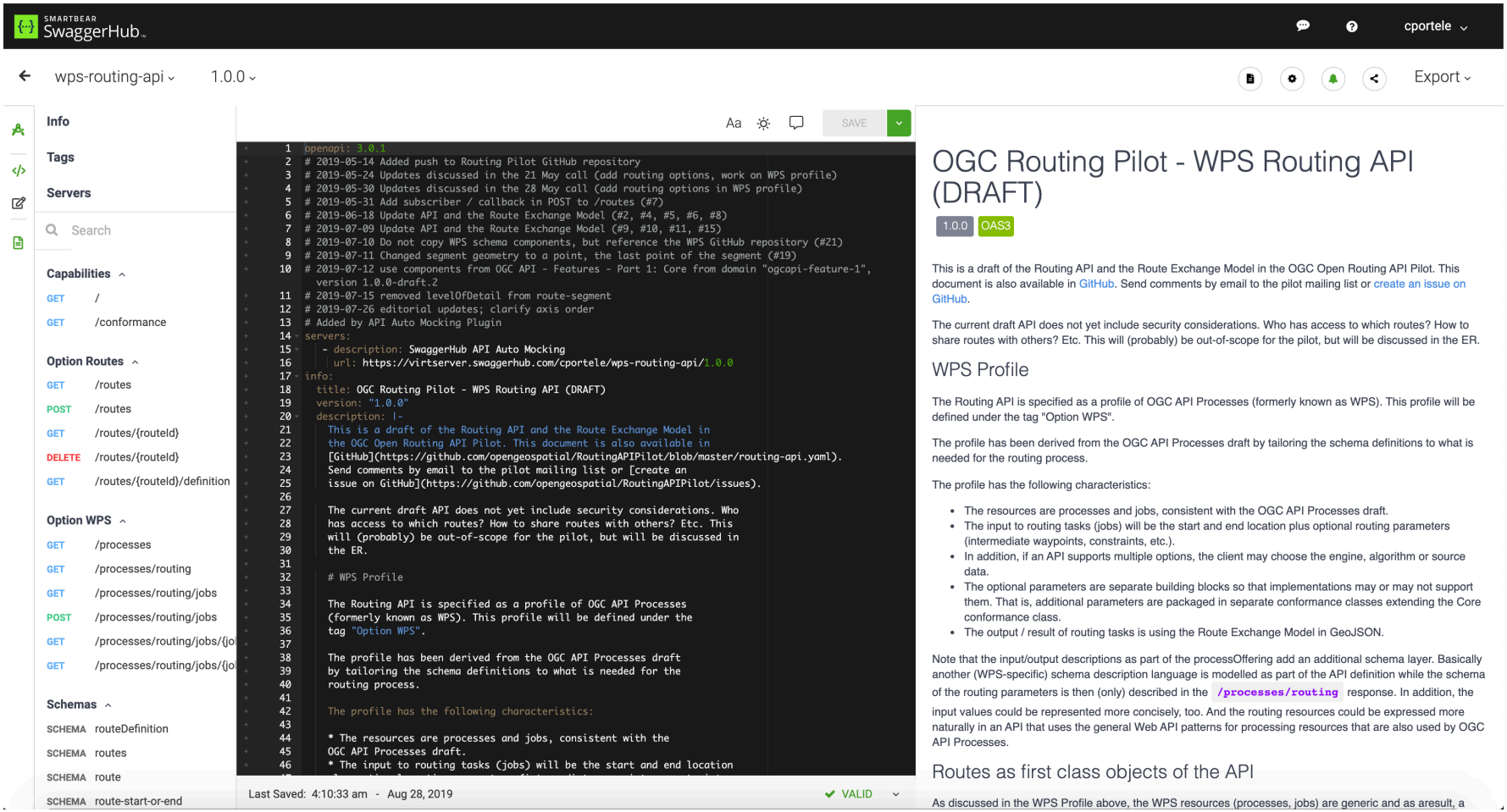
Routing Engines

Ecere	Skymantics	HERE

Source Datasets

OSM **HERE** **NSG**

Step 1: API design on SwaggerHub
Step 2: Specification drafted in the ER



https://app.swaggerhub.com/apis/cportele/wps-routing-api/1.0.0

OGC®

Routing API overview

Table 1. Overview of resources and applicable HTTP methods

Resource	Path	HTTP method	Document reference
Landing page	`/`	GET	API landing page
Conformance declaration	`/conformance`	GET	Declaration of conformance classes
Routes	`/routes`	GET	Fetch routes
		POST	Compute a new route
Route	`/routes/{routeId}`	GET	Fetch a route
		DELETE	Delete a route
Route definition	`/routes/{routeId}/definition`	GET	Fetch the definition of a route

OGC®

Conformance Classes / API building blocks

- Core: minimal routing capability (asynchronous routing based on start/end point)
- Delete route: Cancel/delete routes
- Callback: Support for call to a webhook after the route is computed
- Synchronous execution: return the route in the response to the routing request
- Result set: request parts of the Route Exchange Model, e.g., in DDIL situations
- Route definition options
  - Intermediate waypoints: Pass through additional points along the route
  - Height restriction: Consider height restrictions
  - Load restriction: Consider load restrictions
  - Obstacles: Avoid obstacles
  - Temporal constraints: Specify departure/arrival time
- Backend options
  - Routing engine: Select a specific routing engine
  - Routing algorithm: Select the algorithm to use
  - Source dataset: Select the network dataset to use

OGC®

Conformance declaration

- Additional information has been included in the conformance declaration to simplify the workflow for clients familiar with the Routing API
- Potentially useful in the OGC API framework in general

Example 2. Conformance declaration


```
{
  "conformsTo": [
    "http://www.opengis.net/spec/ogcapi-processes-1/1.0/conf/core",
    "http://www.opengis.net/orapip/routing/1.0/conf/core",
    "http://www.opengis.net/orapip/routing/1.0/conf/intermediate-waypoints",
    "http://www.opengis.net/orapip/routing/1.0/conf/max-height",
    "http://www.opengis.net/orapip/routing/1.0/conf/max-weight",
    "http://www.opengis.net/orapip/routing/1.0/conf/obstacles",
    "http://www.opengis.net/orapip/routing/1.0/conf/routing-engine",
    "http://www.opengis.net/orapip/routing/1.0/conf/routing-algorithm",
    "http://www.opengis.net/orapip/routing/1.0/conf/source-dataset",
    "http://www.opengis.net/orapip/routing/1.0/conf/time",
    "http://www.opengis.net/orapip/routing/1.0/conf/callback",
    "http://www.opengis.net/orapip/routing/1.0/conf/result-set",
    "http://www.opengis.net/orapip/routing/1.0/conf/sync-mode",
    "http://www.opengis.net/orapip/routing/1.0/conf/delete-route"
  ],
  "http://www.opengis.net/orapip/routing/1.0/conf/core": {
    "values": [
      "fastest",
      "shortest"
    ]
  },
  "http://www.opengis.net/orapip/routing/1.0/conf/routing-engine": {
    "values": [
      "Skymantics",
      "Ecere",
      "HERE"
    ]
  },
  "http://www.opengis.net/orapip/routing/1.0/conf/routing-algorithm": {
    "values": [
      "Dikjstra",
      "Floyd Marshall",
      "A*"
    ]
  },
  "http://www.opengis.net/orapip/routing/1.0/conf/source-dataset": {
    "values": [
      "NSG",
      "OSM",
      "HERE"
    ]
  }
}
```


OGC®

Route Exchange Model

- GeoJSON Feature Collection with additional member “status”
- “Full” representation:
  - Overview (Line String) ← “Overview” representation: just this feature
  - Start (Point)
  - Segment 1 (Point) ← “Segments” representation: a segment with a next link
  - …
  - Segment n (Point)
  - End (Point)
- During computation
  - Overview (null)
  - Start (Point)
  - End (Point)

OGC®

Requirements

- Consistent with current OGC API implementations
- GeoJSON to benefit from existing tooling
- Support for fetching parts of a route in environments with low bandwidth or intermittent connectivity
- Modular and extensible

OGC®

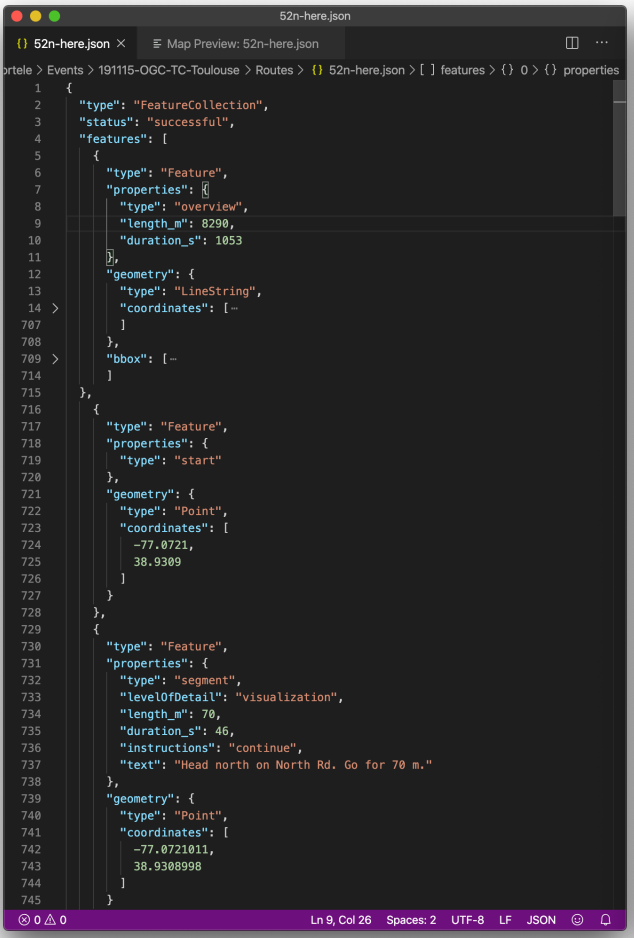
Conformance classes

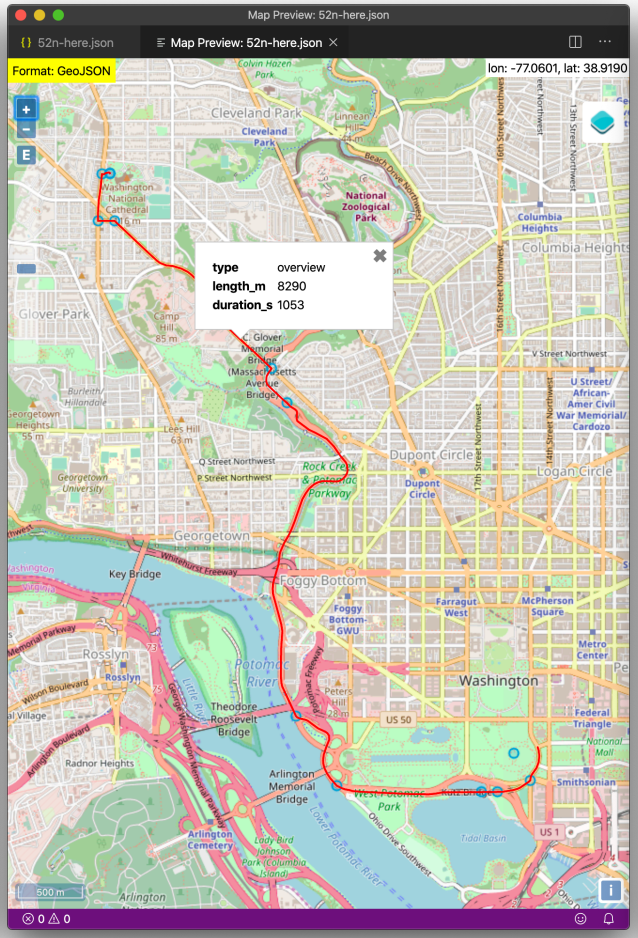
- Full
  - all information
- Overview
  - Just the RouteOverview feature
- Segments
  - Just the first RouteSegment with a “next” link

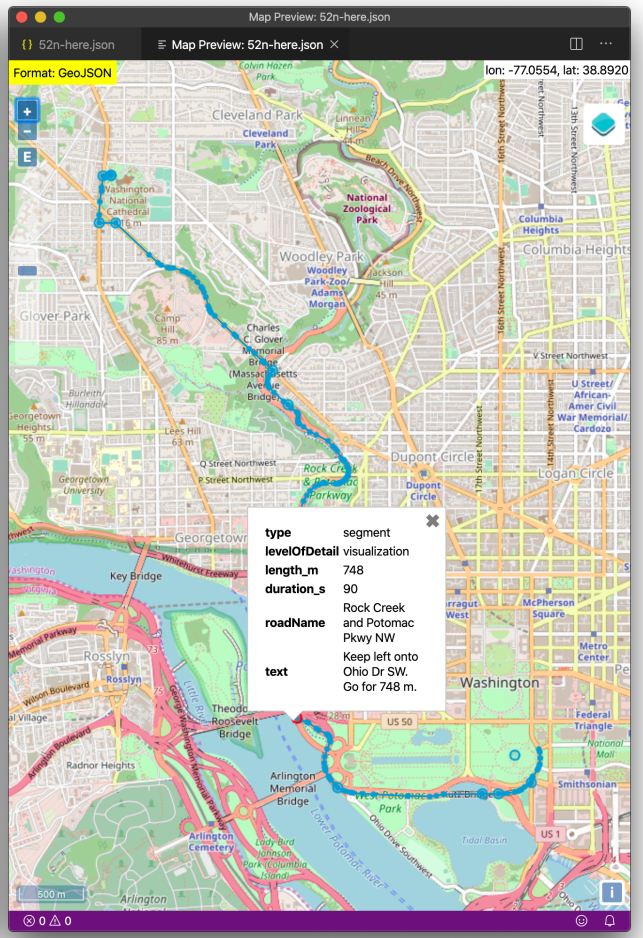
OGC®

A Route example

in Visual Studio Code editor







OGC®

Sponsor

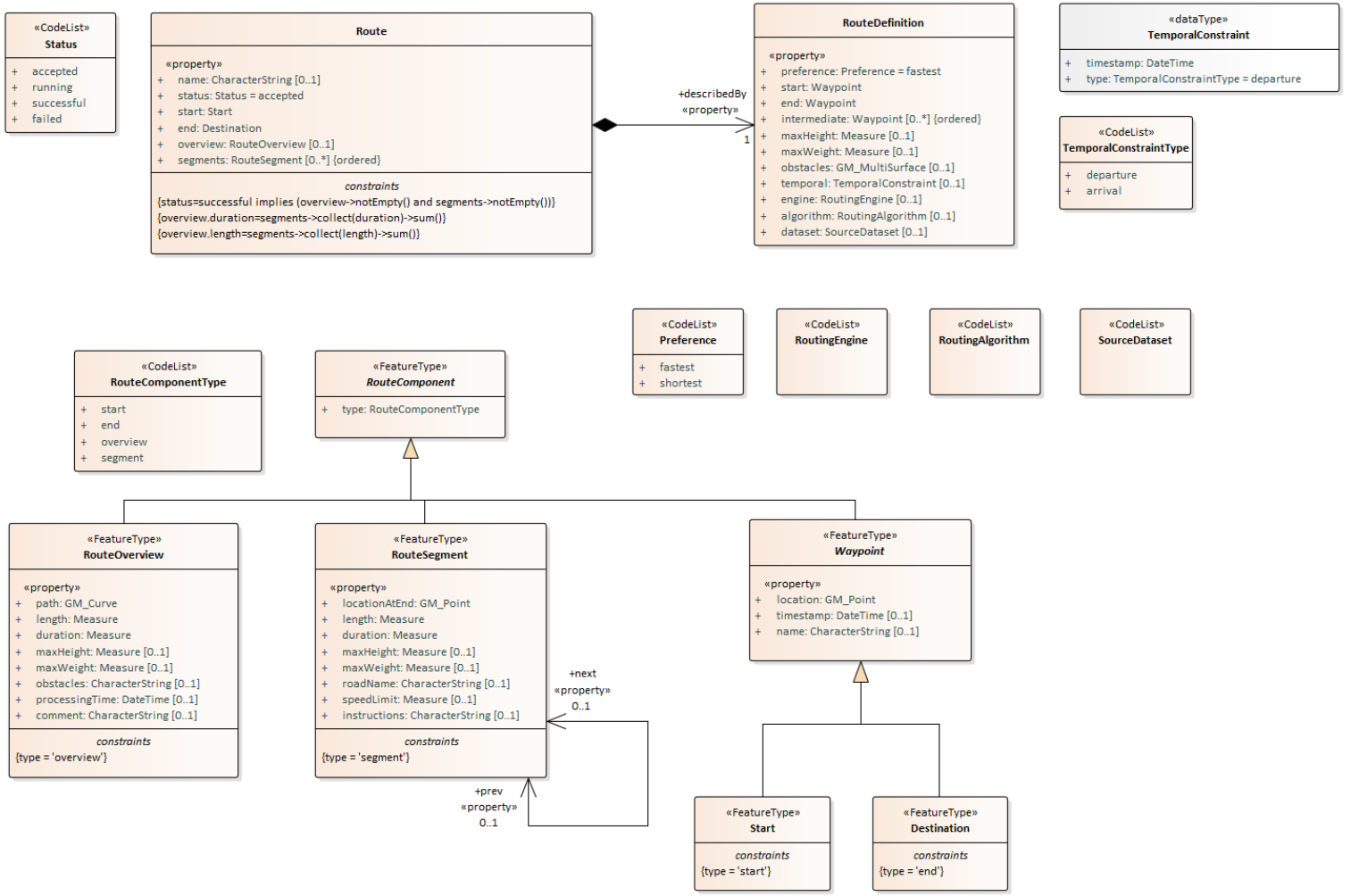
AIRBUS

Route Ontology and Encoding

113th OGC Technical Committee
Toulouse, France
Clemens Portele
20 November 2019

OGC®
Making location count.

Route Model



OGC®

Route Ontology ?

- An ontology could be abstracted from the Route Model and the Routing API specified in the pilot
  - Does not cover everything as the pilot has covered only a selected set of use cases and requirements
- Consistency with the Route Exchange Model and the continued use of GeoJSON is important
  - It should be possible to amend the GeoJSON with a JSON-LD context based on the ontology
  - Requires JSON-LD 1.1 due to the use of nested lists for geometries

OGC®

Step 1:
Derive the vocabulary from the Route Model

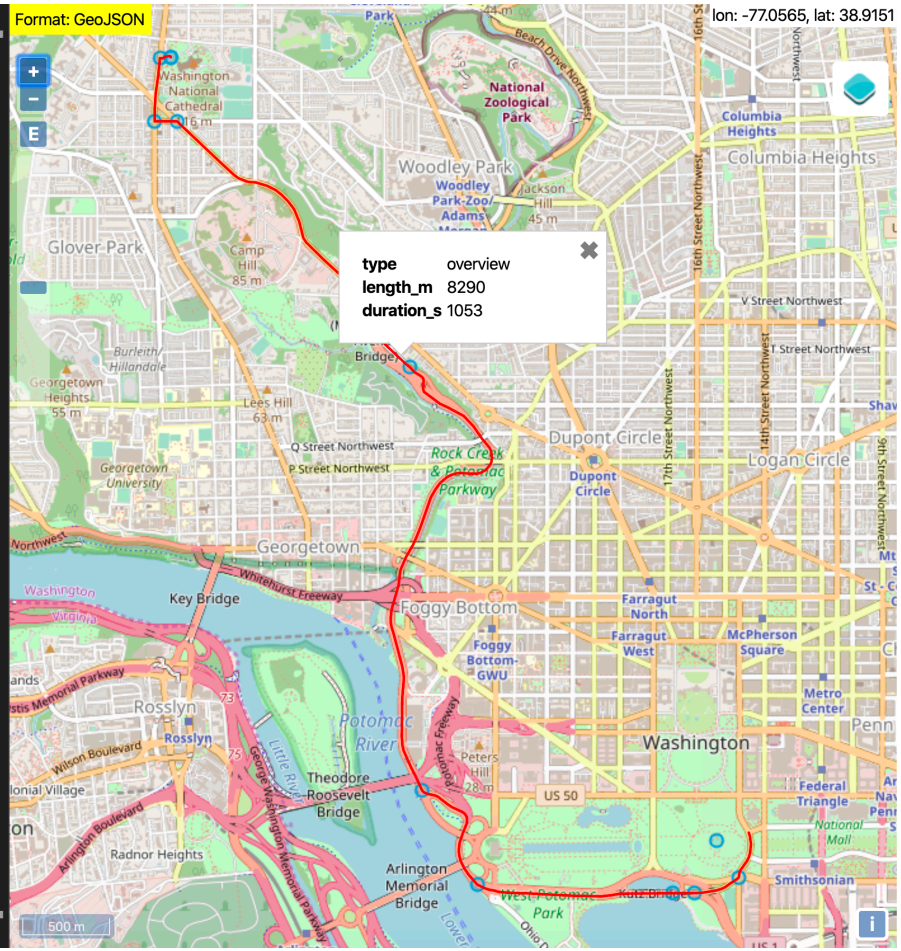
```
@prefix rte:   <https://raw.githubusercontent.com/cportele/route/master/route#> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl:   <http://www.w3.org/2002/07/owl#> .
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#> .
@prefix skos:  <http://www.w3.org/2004/02/skos/core#> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .

rte:Route
        a                   owl:Class ;
        skos:definition     "A way to get from a starting point to a destination."@en .

rte:RouteComponent
        a                   owl:Class ;
        skos:definition     "A component of the route."@en ;
        rdfs:comment        "For example, an overview of the complete route or a segment or waypoint along the route."@en .

rte:RouteOverview
        a                   owl:Class ;
        rdfs:subClassOf     rte:RouteComponent ;
        skos:definition     "An overview of the complete route from start to destination with the main properties of the route."@en .

rte:RouteSegment
        a                   owl:Class ;
        rdfs:subClassOf     rte:RouteComponent ;
        skos:definition     "An overview of the complete route from start to destination with the main properties of the route."@en .

rte:Waypoint
        a                   owl:Class ;
        rdfs:subClassOf     rte:RouteComponent ;
        skos:definition     "An overview of the complete route from start to destination with the main properties of the route."@en .

rte:Start
        a                   owl:Class ;
```

OGC®

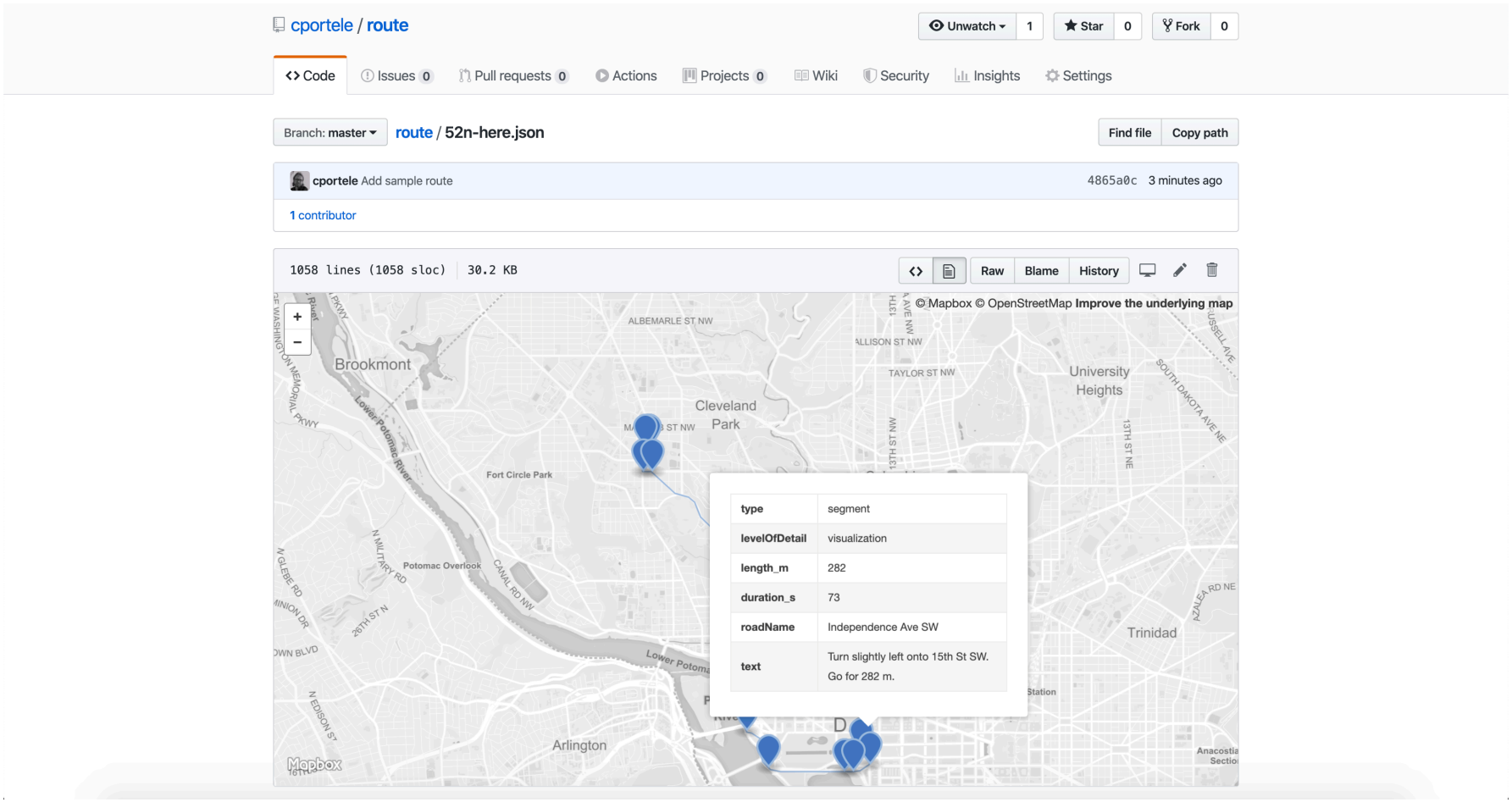
Step 2:
Define a context, amend the GeoJSON

Users > portele > Documents > GitHub > ogc > route > {} 52n-here.jsonld > ...

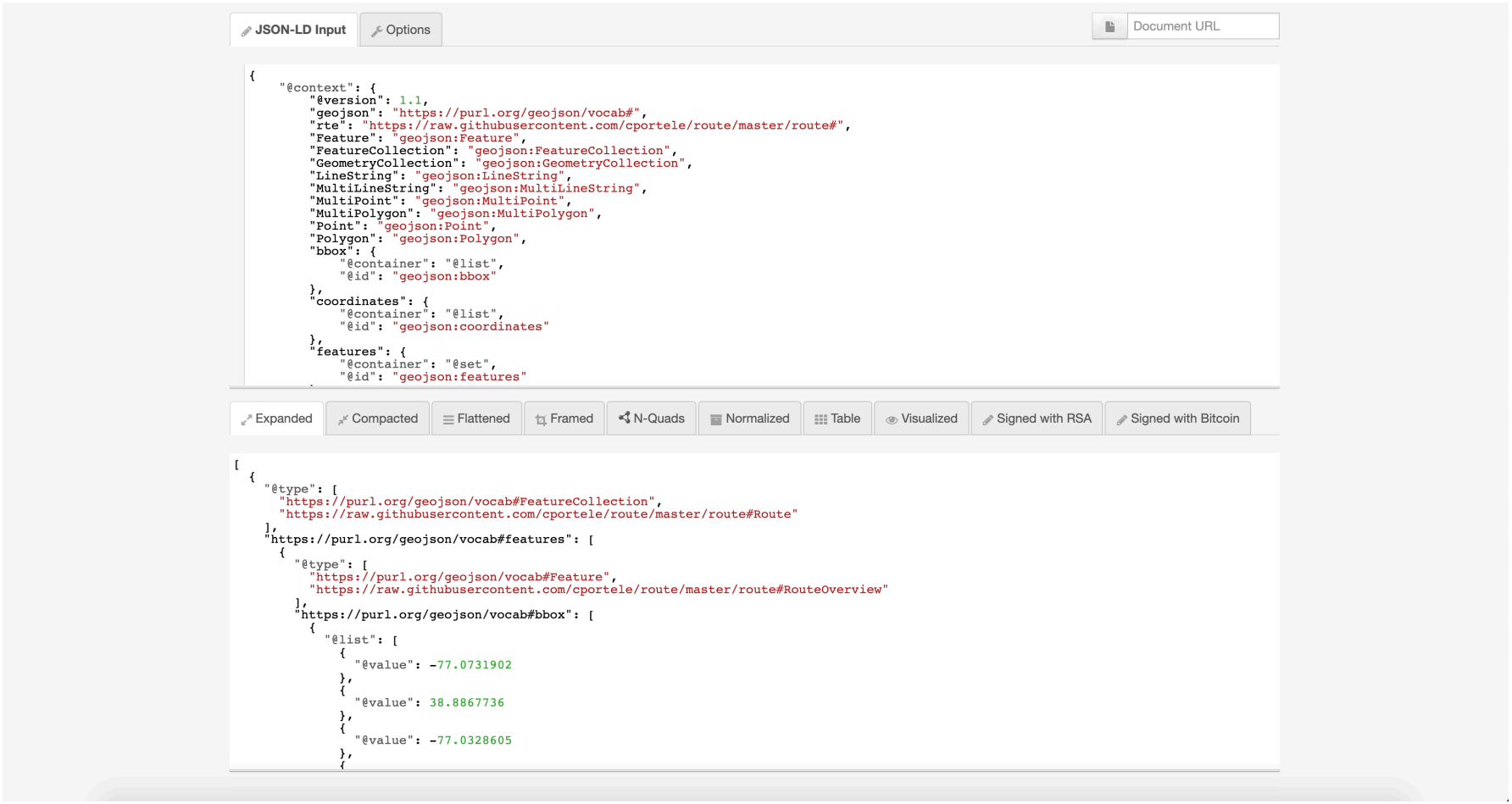
```
{
    "@context": {
        "@version": 1.1,
        "geojson": "https://purl.org/geojson/vocab#",
        "rte": "https://raw.githubusercontent.com/cportele/route/master/route#",
        "Feature": "geojson:Feature",
        "FeatureCollection": "geojson:FeatureCollection",
        "GeometryCollection": "geojson:GeometryCollection",
        "LineString": "geojson:LineString",
        "MultiLineString": "geojson:MultiLineString",
        "MultiPoint": "geojson:MultiPoint",
        "MultiPolygon": "geojson:MultiPolygon",
        "Point": "geojson:Point",
        "Polygon": "geojson:Polygon",
        "bbox": {
            "@container": "@list",
            "@id": "geojson:bbox"
        },
        "coordinates": {
            "@container": "@list",
            "@id": "geojson:coordinates"
        },
        "features": {
            "@container": "@set",
            "@id": "geojson:features"
        },
        "geometry": "geojson:geometry",
        "id": "@id",
        "properties": "@nest",
        "name": "rdfs:label",
        "status": "rte:status",
        "length_m": "rte:length",
        "duration_s": "rte:duration",
        "instructions": "rte:instructions",
        "text": "rte:text",
        "roadName": "rte:roadName"
    },
    "type": "FeatureCollection",
    "id": "https://api.example.com/v1/routes/4711",
    "@type": [
```



Same document in GitHub



Same document in the JSON-LD Playground showing the semantic annotations



OGC®

Some open issues

- What media type to use for such documents?
  - application/geo+json – it works whereever GeoJSON works
  - application/ld+json – it has semantic annotations
- GeoJSON has a predefined context that I could have referenced directly, but I wanted to deviate from it
  - Use JSON-LD 1.1 (lists of lists, support for @nest)
  - Support multiple values for @type
- How to handle a JSON representation of Web Links based on RFC 8288 in a JSON-LD context?

OGC®

Strategic Member Support by

Natural Resources Canada | Ressources naturelles Canada

Canada

Additional Sponsorship

esri Canada

Routing SWG Charter

114th OGC Technical Committee

Ottawa, Canada

Jeff Harrison

5 March 2020

OGC®

Making location count.

Routing SWG Purpose

- The purpose of this Standards Working Group is to:
- Develop and maintain a routing conceptual model, a routing ontology, and a route exchange model.
- Develop and maintain an OGC API – Open Routing core standard.
- Develop and maintain extensions of the OGC API - Open Routing standard as identified in section 4 (Scope).

The formal proposed name of the new standard is "OGC API – Open Routing - Part 1: Core". The short name “Routing API” may also be used to refer to this effort in the charter and work products

OGC®

Routing SWG Business Value

- The Routing SWG will define and document a common exchange model for routing information and create an OGC API- Routing specification.
  - The route exchange model provides a common means for route consumers and routing services to exchange information to enable interoperability and expand the ability of routing services to incorporate diverse data, routing algorithms, and planning parameters in a standardized way. This will allow a variety of users and systems to request and receive routing solutions using a networked routing analytic workflow.
  - The OGC API defines a baseline suite of Routing API functions, capabilities, and encodings to address a common standard interface for network routing functionality, develop guidance for extending Routing APIs to account for various routing data models, and provide guidance for network routing engine configuration for the Routing API. The proposed API provides the methods and apparatus to support:
    - Discovery of route resources and metadata;
    - Execute basic commands to GET, PUT, PATCH, POST, and DELETE route resources, requests, and response; and
    - Interface with other web resources using OGC API capabilities and OGC web services.
- For providers of routing engines and solvers, the exchange model and API provide a uniform means to publish and offer those routing resources and capabilities for use by other systems.
- Users of routing engines and solvers are provided with a common exchange model and API that will allow them to access multiple capabilities and routing datasets to provide efficient access and application of multiple route resources offered through the API.

OGC®

Existing and Related Routing Efforts

- The starting point for the work will be:
  - OGC 19-041r3: Routing Pilot Engineering Report and
  - OGC 19-040 WPS Routing API (draft specification) Engineering Report
- The work shall also be informed by the following specifications and by recommendations found in:
  - OGC/W3C Spatial Data Working Group on the Web Best Practices (https://www.w3.org/TR/sdw- bp/);
  - OGC Geospatial API White Paper [OGC 16-019r4]; and
  - OGC API - Features - Part 1: Core standard, [OGC 17-069r3].
- The following standards work may be applicable to the work of the proposed SWG:
  - 16-120r3, OGC Moving Feature Access
  - 16-140, OGC Moving Feature Encoding Extension – JSON (Best Practice document includes API, being updated in 19-045r2, OGC Moving Features Encoding Extension – JSON (candidate standard)
- Additionally, the proposed SWG will monitor other OGC API work ongoing in various Standards and Innovation Program activities (e.g., OGC API - Common and API work on Processing, Coverages, Tiles, Moving Features SWG, etc.).
- Each of these documents recommends an emphasis on resource-oriented APIs in future OGC standards development including use of tools such as OpenAPI.

OGC®

Routing SWG Scope

- Route Exchange Model and Routing API were demonstrated in the OGC Open Routing API Pilot
  - OpenAPI frameworks have helped make describing and sharing API definitions more suitable for interoperability standardization.
- The Routing SWG will build on those preliminary efforts
  - More fully develop and document a Route Exchange Model and Routing API candidate standard that will provide a modernized, common, and consistent interface to services
  - Work will align with the current architecture of the Web and the Spatial Data on the Web Best Practices and be done in coordination with Joint W3C/OGC SDWIG
- Routing API candidate standard informed by emerging OGC API best practices and prior API standards examples (e.g., OGC API - Features)
  - Define core API functions of GET, PUT, PATCH, POST, DELETE applied to routes as resources.
  - Document metadata requirements for routes to enhance discovery and exchange of routing resources.
- Implementer and standards development interaction
  - Developers encouraged to implement the draft API specification early and provide feedback.
  - Public access to draft versions of the standard using GitHub
  - Before finalizing versions of the "OGC API - Routing", completion of goals should be verified:
    - Working implementations of all capabilities must be available and tested; and
    - Implementation feedback must be taken into account.

OGC®

Routing SWG Scope: Modularization

- OGC API - Open Routing - Part 1: Core will define a basic capabilities in multiple conformance classes
  - The minimal conformance class will specify a simple interface to access metadata from routing resources that is sufficient for interfaces to exchange and perform basic web functions with the routing resources.
  - Additional conformance classes will define additional capabilities based on the requirements and requirements classes defined in the core to meet the needs of use cases that require such capabilities.
  - The Open Routing API Pilot identified two approaches for a routing API; one aligned more specially to the current draft OGC API - Processes specification and another aligned more generally to OGC API - Common.
    - The OGC API – Routing Part 1: Core will initially focus on the routing/common approach and will incorporate the processing approach as it matures.
- The Route Exchange Model, containing a conceptual model and route ontology, will be developed independent of the API specification.
  - Provides a more general specification for routing approaches that could be developed independent of an API but provide a shared basis to ensure interoperability of information exchange.
- The Open Routing API Pilot identified work items that will be refined and assigned to the API Core, API Extensions, or the Route Exchange Model.
  - Supporting additional routing constraints (e.g. elevation, restricted maneuvers, speed limits, street hierarchies)
  - Incorporating additional transportation methods as parameters (e.g. pedestrians, cycling, motor vehicles)
  - Addressing alternative routing output (e.g. catchment and distance area and traveling salesman problem (multi-node optimization)
- Other extensions may be proposed and addressed in revisions to this charter.

OGC®

Routing SWG Deliverables

- The following deliverables will result from the work of this SWG:
  - A final version of the Route Exchange Model document for submission to the TC.
  - A final version of the OGC API - Open Routing - Part 1: Core document for submission to the TC;
  - Identification of at least three prototype implementations of the core based on the standard — although more would be preferred; and
  - Zero or more additional parts as time and community interest permits.
- Part 1 will cover basic capabilities to GET, PUT, PATCH, POST, and DELETE routes and define route metadata.
  - Capabilities for richer routing interfaces or extension for unique geospatial resource considerations will be specified in additional parts.
- The targeted start date is in June 2020 once this charter is approved.
- Initial candidate route exchange model and API specifications are anticipated by the end of calendar year 2020 with demonstrated implementations and formal approval of the core Open Routing API in 2021.

OGC®