OGC® DOCUMENT: 21-038R1

External identifier of this OGC® document: https://www.opengis.net/doc/is/ogcapi-dggs/{m.n}

Open Geospatial Consortium

OGC API - DISCRETE GLOBAL GRID SYSTEMS - PART 1: CORE

STANDARD
Implementation

CANDIDATE SWG DRAFT

Version: 1.0
Submission Date: 2025-02-13
Approval Date: 2025-05-31
Publication Date: 2025-05-31
Editor: Dr. Matthew Brian John Purss, Jérôme Jacovella-St-Louis

I ABSTRACT

The OGC API — Discrete Global Grid Systems — Part 1: Core Standard defines requirements that can be implemented and integrated into a deployed Web API. The DGGS API allows to retrieve geospatial data for a specific area, time and resolution of interest, based on a specific Discrete Global Grid Reference System (DGGRS). Additionally, the DGGS API defines operations to query the list of DGGRS zones from which data is available and/or matching a specified query. Such queries are expressed by combining HTTP query parameters, including the possibility to express more complex filtering queries using the OGC Common Query Language (CQL2).

Readers are strongly encouraged to first familiarize themselves with the Discrete Global Grid Systems (DGGS) concepts defined in OGC Abstract Specification Topic 21 laying the foundation for this Standard.

Please note that in the remainder of this document, the Standard will be referred to as the "OGC API – DGGS Standard", "the DGGS API" or "this Standard".

II KEYWORDS

The following are keywords to be used by search engines and document catalogues.

ogcdoc, OGC document, API, openapi, html, ogcapi, DGGS, discrete global grid, ISEA, hexagonal grids

III

PREFACE

This document defines the *OGC API — Discrete Global Grid Systems — Part 1: Core* Standard. Suggested additions, changes and comments on this standard are welcome and encouraged. Such suggestions may be submitted as an issue on the *OGC API — DGGS* GitHub repository.

OGC API Standards define modular API components to spatially enable Web APIs in a consistent way. OGC API Standards use the OpenAPI specification for describing the API components.

The *OGC API — Discrete Global Grid Systems* defines API components to retrieve data and query zones based on Discrete Global Grid Systems (DGGS) concepts defined in OGC Abstract Specification Topic 21. Additional parts for the *OGC API — Discrete Global Grid Systems* Standard may be defined in the future to provide additional capabilities.

IV SECURITY CONSIDERATIONS

OGC API — DGGS — Part 1 only defines HTTP GET operations. As such the security considerations are limited to those applicable to a "read only" service. However, implementations of the OGC API — DGGS Standard will have to process resource paths and query parameters in a way that cannot be used by a client to inject malicious queries that makes available unforeseen data or even force the server to perform unwanted or dangerous actions. The following paragraphs enumerate some security considerations.

Due to the flexibility in querying DGGS zones, implementations of the *OGC API — DGGS* Standard that are not optimized can easily encounter requests that take some time to resolve. If several of these requests are processed simultaneously, the server can become slow or unresponsive. Servers should take advantage of the space partitioning structures of the implemented DGGRSs that guarantee a deterministic maximum amount of data to process, ensuring that maximum refinement level is used for regions covering a larger area, and refusing requests which would otherwise be problematic for the server. Servers should also set and apply reasonable limits (e.g., maximum number of zones to return, etc.) to prevent Denial of Service attacks.

Some deployments of the OGC API — DGGS Standard may assign different roles to different users that may result in accessing different collections or geographical areas for which data can be retrieved or zones can be queried. The access control can be described in the OpenAPI definition as discussed in the OGC API — Common Standard (https://docs.ogc.org/is/19-072/19-072.html#rc_oas30-security). Servers should take care that all resources in all representations and ways they can be requested (e.g., adding query parameters) are managed consistently.

Another security consideration is that DGGS data or DGGS zone lists returned by a DGGS API endpoint may be positionally incorrect and potentially guide users to wrong locations. One possible way this situation could happen is the manipulation by a malicious actor of a projection library installed on a server deployment.

Using HTTPS queries is preferred to using HTTP. The obvious reason is the DGGS response returned by the DGGS API endpoint should be only accessible to the requesting user. Another reason is related to the public accessibility of the request itself. Request to a DGGS API endpoint reveal geographical extents that might be associated with private or sensitive information. For instance, users commonly visit "home" or "work" or "target places" (such as holiday destinations or churches). These requests can be used to predict personal or private activities.

V SUBMITTING ORGANIZATIONS

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

- Pangaea Innovations Pty. Ltd.
- Ecere Corporation
- Landcare Research New Zealand
- Peking University Collaborative Innovation Center for Geospatial Big Data
- European Commission Joint Research Centre
- Natural Resources Canada
- University of Tartu
- European Space Agency
- Open Geospatial Foundation
- University of Calgary
- GeoInsight AG

VI SUBMITTERS

All questions regarding this submission should be directed to the editors or the submitters:

Name	Affiliation
Matthew Brian John Purss (editor)	Pangaea Innovations Pty. Ltd.
Jérôme Jacovella-St-Louis (editor)	Ecere Corporation
Robert Gibb	Landcare Research New Zealand
Fuhu Ren / 任伏虎	Peking University Collaborative Innovation Center for Geospatial Big Data
Peter Strobl	European Commission Joint Research Centre
Ryan Ahola	Natural Resources Canada

Alexander Kmoch	University of Tartu
Ferran Gascon	European Space Agency
Luís Moreira de Sousa	Open Geospatial Foundation
Perry Peterson	University of Calgary
Michael Jendryke	GeoInsight AG

VII ACKNOWLEDGEMENTS

This OGC Standard builds on a vast amount of research on Discrete Global Grid Systems spanning multiple decades, in particular from John P. Snyder (Snyder1992), Michael F. Goodchild (Goodchild1992), Kevin Sahr (Sahr2003), Robert Gibb (Gibb2016), Perry R. Peterson (Peterson2019), and several others.

Initial implementations of a previous iteration of a draft DGGS API and an Engineering Report produced thanks to funding from OGC Testbed 16 (OGC 20-039r2) informed the development of this Standard. Note that the API defined by this Standard differs significantly from that earlier iteration, with a greater emphasis on client requests asking a Web API implementation questions of the type *"What is here?"* and *"Where is it"*? (Peterson2019). This Standard assumes a shared understanding of a particular Discrete Global Grid Reference System between the client and the server, which could be implemented by a local DGGRS software library, avoiding the need to define HTTP resources to query the conceptual grid hierarchy structure. The definition of an API in the traditional software development sense for such DGGRS libraries is beyond the scope of this Standard.

Key work leading to the creation of this OGC Standard was undertaken as part of GeoConnections 2020-2021 project number GNS20-21IFP02 (*Modular OGC API Workflows*) (OGC 21-033) thanks to funding from Natural Resources Canada. Research activity and Technology Integration Experiments performed as part of the OGC Federated Marine SDI — Phase 3 (OGC 23-010) pilot project also played a key role in refining the design of the API and the Discrete Global Grid Reference Systems based on the ISEA projection included in Annex B.

1

SCOPE

1 SCOPE

This OGC API — DGGS Standard specifies a Web API that enables the direct retrieval of geospatial data for a specified area, time and resolution of interest, based on a specific Discrete Global Grid Reference System (DGGRS). Additionally, the DGGS API defines operations to query the list of DGGRS Zones from which data is available and/or matching a particular query. Such queries are expressed by combining HTTP query parameters, including the possibility to express more complex filtering queries using the OGC Common Query Language (CQL2).

This Standard also specifies several encodings for retrieving both DGGS data and lists of zones, including JSON encodings suitable for data quantized to DGGRS zones of arbitrary geometries as well as widely adopted geospatial data formats. Compact binary representations for zone data as well as for lists of zones are proposed as well.

An informative annex presents a JSON schema for describing a Discrete Global Grid Reference System (DGGRS), as well as examples of such DGGRS definitions which can be implemented in connection with this standard.

2 CONFORMANCE

2 CONFORMANCE

The one Standardization Target for this Standard is Web APIs.

OGC API — Common provides a common foundation for OGC API Standards. Some conformance classes of this standard have a dependency on, or are designed to be easily integrated with, conformance classes defined in OGC API — Common — Part 1 and/or *Part 2*, as well as within a Web API conforming to additional OGC API Standards.

This Standard identifies twenty-five Conformance Classes. Each Conformance Class has an associated Requirements Class. The Requirements Classes define the functional requirements which will be tested through the associated Conformance Class. Only the Core requirements class is mandatory, all others are optional. However, a Web API with practical use needs to implement either the Data Retrieval and/or the Zone Query requirements class.

Conformance with this Standard shall be checked using all the relevant tests specified in Annex A (normative) of this document. The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in the OGC Compliance Testing Policies and Procedures and the OGC Compliance Testing web site.

To conform to this OGC API — DGGS Standard, a software implementation shall implement at minimum the "Core" requirements class, as well as either the "Data Retrieval" and/or the "Zone Query" requirements class.

All requirements-classes and conformance-classes described in this document are owned by the standard(s) identified.

The Requirements Classes for *OGC API — Discrete Global Grid Systems* are:

2.1. Requirements classes defining resources

In addition to the Core requirements class defining the `../dggs`, `../dggs/{dggrsId}` and `../dggs/{dggrsId}/zones/{zoneId}` resources, either the Zone Data retrieval (`../dggs/{dggrsId}/zones/{zoneId}/data`) or Zone Query requirements classes (`../dggs/{dggrsId}/zones` with support for query parameters such as `zone-level=`, `parent-zone=`, `compact-zones=`, `bbox=`, `bbox-crs`, `datetime=`, `subset=` and `subset-crs=`) need to be implemented for a DGGS API implementation to have practical use.

- **Core**
- **Zone Data Retrieval**
- **Zone Query**

2.2. Requirements classes for integration within the OGC API framework

These requirements classes define DGGS resources attached to the root of the API (`/dggs`) and collections (`/collections/{collectionId}/dggs`) origin resources defined by the OGC API — Common framework.

- **Root DGGS**
- **Collection DGGS**

The *Operation IDs* Requirements Class defines requirements for using specific operation IDs in API definitions, which can be combined with requirements classes for specific API definition, such as the OpenAPI 3.0 requirements class of *OGC API — Common — Part 1: Core*, facilitating the identification of DGGS resources.

- **Operation IDs**

2.3. Requirements classes defining query parameters

Three requirements classes add support for the `subset`, `properties`, `exclude-properties`, `zone-depth` and `filter` query parameters to zone data resources:

- **Zone data subsetting**
- **Zone data custom depths**
- **Filtering zone data with CQL2**

A requirements class adds support for the `filter` parameter for zone query resources:

- **Filtering Zone Queries with CQL2**

2.4. Requirements classes defining resource representations

2.4.1. Zone data encodings

The Data Encoding Requirements Classes address support for formats commonly used for encoding geospatial data, as well as encodings based on JSON and UBJSON using a schema specifically designed for zone data referenced to a particular DGGRS. DGGS-(UB)JSON allows encoding one or more depths of DGGS-quantized raster data, including optional support for multiple fields (properties) and additional dimensions beyond those of the DGGS, whereas DGGS-(UB)JSON-FG encode vector features and geometry based on OGC Features Geometry JSON using sub-zone order indices as coordinates at an arbitrary relative depth / precision.

- DGGS-JSON Data
- DGGS-UBJSON Data
- DGGS-JSON-FG Data
- DGGS-UBJSON-FG Data
- GeoJSON Data
- GeoTIFF Data
- netCDF Data
- Zarr Data
- CoverageJSON Data
- JPEG XL Data
- PNG Data

2.4.2. Zone list encodings

The Zone List Encoding Requirements Classes address support for formats efficiently and/or intuitively encoding a list of DGGS zone identifiers, in addition to the simple JSON encoding.

- HTML Zone List
- 64-bit Binary Zone List
- GeoJSON Zone List

- GeoTIFF Zone List

Implementations supporting GeoJSON for either zone data or zone lists are strongly encouraged to also support GeoJSON extensions defined by the Features Geometry JSON (JSON-FG) Standard. Conformance to this Standard can be indicated in the Web API deployment's conformance declaration using its associated conformance class URIs, such as `http://www.opengis.net/spec/json-fg-1/0.2/conf/core`.

2.5. Summary of conformance URIs

Table 1 — Conformance class URIs

CORRESPONDING REQUIREMENTS CLASS	CONFORMANCE CLASS URI
Core	**https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/core**
For Data Retrieval	
Zone Data Retrieval	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-retrieval
Data subsetting	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-subsetting
Data custom depths	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-custom-depths
Filtering Zone Data with CQL2	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-cql2-filter
For Zone Queries	
Zone Query	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-query
Filtering Zone Queries with CQL2	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-query-cql2-filter
OGC API integration	
Root DGGS	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/root-dggs
Collection DGGS	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/collection-dggs
Operation IDs	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/operation-ids
Zone Data Encodings	

CORRESPONDING REQUIREMENTS CLASS	CONFORMANCE CLASS URI
DGGS-JSON Data	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-json
DGGS-UBJSON Data	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-ubjson
DGGS-JSON-FG Data	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-dggs-jsonfg
DGGS-UBJSON-FG Data	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-dggs-ubjsonfg
GeoJSON Data	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-geojson
GeoTIFF Data	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-geotiff
netCDF Data	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-netcdf
CoverageJSON Data	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-coveragejson
Zarr Data	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-zarr
JPEG XL Data	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-jpegxl
PNG Data	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-png
Zone List Encodings	
HTML Zone List	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-html
64-bit Binary Zone List	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-uint64
GeoJSON Zone List	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-geojson
GeoTIFF Zone List	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-geotiff

3 NORMATIVE REFERENCES

3 NORMATIVE REFERENCES

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

Robert Gibb: OGC 20-040r3, *Topic 21 — Discrete Global Grid Systems — Part 1 Core Reference system and Operations and Equal Area Earth Reference System*. Open Geospatial Consortium (2021). http://www.opengis.net/doc/AS/dggs/2.0.

T. Bray (ed.): IETF RFC 8259, *The JavaScript Object Notation (JSON) Data Interchange Format*. RFC Publisher (2017). https://www.rfc-editor.org/info/rfc8259.

Charles Heazel: OGC 19-072, *OGC API — Common — Part 1: Core*. Open Geospatial Consortium (2023). http://www.opengis.net/doc/is/ogcapi-common-1/1.0.0.

Charles Heazel: OGC API — Common — Part 2: Geospatial Data (Draft). OGC 20-024, Open Geospatial Consortium, https://docs.ogc.org/DRAFTS/20-024.html

Open API Initiative: OpenAPI Specification 3.0.3, https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.3.md

Ben Domenico: OGC 10-090r3, OGC Network Common Data Form (NetCDF) Core Encoding Standard version 1.0. Open Geospatial Consortium (2011) https://portal.ogc.org/files/?artifact_id=43732

Panagiotis (Peter) A. Vretanos: OGC Common Query Language (CQL2). OGC 21-065r2, Open Geospatial Consortium, https://www.opengis.net/doc/IS/cql2/1.0

H. Butler, M. Daly, A. Doyle, S. Gillies, S. Hagen, T. Schaub: IETF RFC 7946, *The GeoJSON Format*. RFC Publisher (2016). https://www.rfc-editor.org/info/rfc7946.

OGC Features & Geometry JSON (Draft), https://docs.ogc.org/DRAFTS/21-045r1.html

Adobe Developers Association: TIFF Specification Revision 6.0. (1992) https://www.itu.int/itudoc/itu-t/com16/tiff-fx/docs/tiff6.pdf

Emmanuel Devys, Ted Habermann, Chuck Heazel, Roger Lott, Even Rouault: OGC 19-008r4, *OGC GeoTIFF Standard*. Open Geospatial Consortium (2019). http://www.opengis.net/doc/IS/GeoTIFF/1.1.0.

ISO/IEC: ISO/IEC 18181-1, *Information technology — JPEG XL image coding system — Part 1: Core coding system*. International Organization for Standardization, International Electrotechnical Commission, Geneva https://www.iso.org/standard/85066.html.

ISO/IEC: ISO/IEC 18181-2, *Information technology — JPEG XL image coding system — Part 2: File format*. International Organization for Standardization, International Electrotechnical Commission, Geneva https://www.iso.org/standard/85253.html.

ISO/IEC: ISO/IEC 15948, *Information technology — Computer graphics and image processing — Portable Network Graphics (PNG): Functional specification*. International Organization for Standardization, International Electrotechnical Commission, Geneva https://www.iso.org/standard/29581.html.

Chris Little, Jon Blower, Maik Riechert: OGC 21-069r2, *OGC CoverageJSON Community Standard*. Open Geospatial Consortium (2023). http://www.opengis.net/doc/CS/covjson/1.0.

ISO/IEC: ISO/IEC 21320-1, *Information technology — Document Container File — Part 1: Core*. International Organization for Standardization, International Electrotechnical Commission, Geneva https://www.iso.org/standard/60101.html.

Zarr Developers: OGC 21-050r1, *Zarr Storage Specification 2.0 Community Standard*. Open Geospatial Consortium (2022). http://www.opengis.net/doc/CS/zarr/2.0.

4

TERMS AND DEFINITIONS

4 TERMS AND DEFINITIONS

This document uses the terms defined in OGC Policy Directive 49, which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this document and OGC documents do not use the equivalent phrases in the ISO/IEC Directives, Part 2.

This document also uses terms defined in the OGC Standard for Modular specifications (OGC 08-131r3), also known as the ‘ModSpec’. The definitions of terms such as standard, specification, requirement, and conformance test are provided in the ModSpec.

For the purposes of this document, the following additional terms and definitions apply.

4.1. child zone

immediate descendant of a parent zone

Note 1 to entry: adapted from *child cell* in OGC Abstract Specification Topic 21

[**SOURCE:** OGC 20-040r3]

4.2. discrete global grid (DGG)

set of zones at the same refinement level, that uniquely and completely cover a globe

Note 1 to entry: Adapted to use `zones` rather than `cells` to avoid potential confusion with the similar concept of grid cells in a *coverage*. As noted in OGC Abstract Specification Topic 21, *“While the terms cell and zone are often used interchangeably, strictly zone is the preferred term.”*.

[**SOURCE:** OGC 20-040r3]

4.3. discrete global grid hierarchy (DGGH)

series of discrete global grids organized in a hierarchy of successive levels of zone refinement, using a specific set of parameters fully establishing the geometry of all zones

Note 1 to entry: Example of parameters part of the DGGH definition include the underlying Earth model spheroid, and the orientation of a base polyhedron.

Note 2 to entry: This definition corresponds to the term "hierarchy" (4.26) in OGC Abstract Specification Topic 21, and refers strictly to a mathematical structure, excluding the concept of zonal identifiers, which are introduced as part of a discrete global grid reference system, and excluding notions of software functionality or interoperability implied by a DGGS.

Note 3 to entry: In some DGGS literature, the term DGGS refers to this DGGH meaning, where the word *system* in "system of discrete global grids" is used in the sense of a "series" of grids (e.g., *"A Discrete Global Grid System (DGGS) is a series of discrete global grids"* [Sahr2003]), as opposed to a software or data integration / processing system.

Note 4 to entry: The ISEA3H (Icosahedral Snyder Equal-Area Aperture 3 Hexagonal) Discrete Global Grid Hierarchy, using the standard ISEA orientation and the WGS84 authalic sphere as the underlying Earth model, irrespective of a particular indexing scheme or sub-zone ordering, is an example of a DGGH.

[**SOURCE:** OGC 20-040r3]

4.4. discrete global grid reference system (DGGRS)

integrated system comprised of a specific discrete global grid hierarchy, spatiotemporal referencing by zone identifiers and deterministic sub-zone ordering

Note 1 to entry: the term DGGRS did not appear in the terms and definitions of version 2.0.0 of OGC Abstract Specification Topic 21, but is defined in Section 8 as the `DGG_ReferenceSystem` interface.

Note 2 to entry: Annex B provides an informative schema for defining a DGGRS with accompanying examples of specific DGGRSs.

Note 3 to entry: this standard introduces the concept of sub-zone ordering, which is not part of the `DGG_ReferenceSystem` interface defined in topic 21, for the purpose of compact storage and transmission of data quantized to a particular DGGRS.

[**SOURCE:** OGC 20-040r3]

4.5. discrete global grid system (DGGS)

integrated system comprising a hierarchy of discrete global grids, spatiotemporal referencing by zonal identifiers and functions for quantization, zonal query, and interoperability

Note 1 to entry: In some DGGS literature, the term DGGS is sometimes used with the meaning of a discrete global grid hierarchy.

Note 2 to entry: An implementation of the OGC API — DGGS Standard would be considered a DGGS.

[**SOURCE:** OGC 20-040r3]

4.6. globe

region of space-time enclosing a celestial body

[**SOURCE:** OGC 20-040r3]

4.7. parent zone

zone in a coarser refinement level of a discrete global grid with immediate descendants

Note 1 to entry: adapted from *parent cell* in OGC Abstract Specification Topic 21

Note 2 to entry: Adapted to say *coarser* rather than *higher*, where *coarser* refers to a zone covering a larger area of space-time.

[**SOURCE:** OGC 20-040r3]

4.8. refinement level

numerical order of a discrete global grid in the tessellation sequence

Note 1 to entry: The discrete global grid with the least number of zones has a refinement level of 0.

[**SOURCE:** OGC 20-040r3]

4.9. relative depth

the number of refinement levels separating a finer discrete global grid from a coarser discrete global grid in a discrete global grid hierarchy

4.10. slicing

extracting a hyperplane of data along a particular dimension, resulting in a dataset of reduced dimensionality

[**SOURCE:**]

4.11. sub-zone

zone at a greater refinement level than a parent zone whose geometry is at least partially contained within the geometry of the parent zone

Note 1 to entry: At a relative depth of 1, the sub-zones of the parent zone are its immediate children.

Note 2 to entry: In some DGGHs, the children of the children of a parent zone (relative depth of 2) are not all sub-zones of that parent zone.

Note 3 to entry: A discrete global grid reference system defines a deterministic order for sub-zones

4.12. trimming

extracting a subset of data between a lower and upper bound along one or more dimensions, preserving dimensionality

[**SOURCE:**]

4.13. Web API

An Application Programming Interface (API) using an architectural style that is founded on the technologies of the Web

Note 1 to entry: See Best Practice 24: Use Web Standards as the foundation of APIs (W3C Data on the Web Best Practices) for more detail.

[**SOURCE:** OGC 19-072]

4.14. zone identifier

spatiotemporal reference in the form of a label or code that uniquely identifies a zone

Note 1 to entry: synonym of “zonal identifier” and “zone ID”.

Note 2 to entry: This Standard and the DGGRS definitions described in Annex B require textual identifiers (which may or may not be comprised of only digit characters), while optionally supporting 64-bit integer identifiers for the purpose of compact transmission and internal representation.

[**SOURCE:** OGC 20-040r3]

4.15. zone identifier reference system (ZIRS)

reference system establishing a specific association of zone identifiers to zones for one or more discrete global grid hierarchy

Note 1 to entry: synonym of “zonal identifier reference system” and “zone indexing scheme”

4.16. zone

particular region of space-time

[**SOURCE:** OGC 20-040r3]

5

CONVENTIONS

5 CONVENTIONS

This section provides details of conventions used in this document.

5.1. Identifiers

The normative provisions in this standard are denoted by the URI `https://www.opengis.net/spec/ogcapi-dggs-1/1.0`.

All requirements and conformance tests that appear in this document are denoted by partial URIs which are relative to this base.

5.2. Link relations

To express relationships between resources, RFC 8288 (Web Linking) is used.

Note that the CURIE based on the pattern `[ogc-rel:<relation>]` is also considered equivalent. For example, `[ogc-rel:dggrs-definition]` corresponds to `https://www.opengis.net/def/rel/ogc/1.0/dggrs-definition`.

The following IANA link relation types are used in this document:

- **alternate**: Refers to a substitute for this context.
- **self**: Conveys an identifier for the link's context.
- **service-desc**: Identifies service description for the context that is primarily intended for consumption by machines (Web API definitions are considered service descriptions).
- **service-doc**: Identifies service documentation for the context that is primarily intended for human consumption.

The following link relation types which were previously defined for other OGC API standards are applicable:

- **https://www.opengis.net/def/rel/ogc/1.0/dataset**: The target IRI points to a resource representing the dataset (e.g., the root of an OGC Web API).
- **https://www.opengis.net/def/rel/ogc/1.0/geodata**: The target IRI points to a resource representing a collection of geospatial data.

In addition, the following link relation types are used for which no applicable registered link relation type could be identified:

- **https://www.opengis.net/def/rel/ogc/1.0/dggrs**: The target IRI points to the description, which includes a link to the definition as well as deployment-specific elements, of the Discrete Global Grid Reference System related to the link's context.
- **https://www.opengis.net/def/rel/ogc/1.0/dggrs-list**: The target IRI points to the descriptive list of the Discrete Global Grid Reference Systems available for the resource of the link's context.
- **https://www.opengis.net/def/rel/ogc/1.0/dggrs-definition**: The target IRI points to the definition of the Discrete Global Grid Reference System using a well-defined schema (see Annex B), for example as available from an authoritative DGGRS registry.
- **https://www.opengis.net/def/rel/ogc/1.0/dggrs-zone-info**: The target IRI points to information about a particular zone of a Discrete Global Grid Reference System, such as its geometry, area/volume, and any relevant links (e.g., to retrieve data from the zone).
- **https://www.opengis.net/def/rel/ogc/1.0/dggrs-zone-data**: The target IRI points to data pertaining to a particular zone of a Discrete Global Grid Reference System. This link relation is typically used with a link template where a zone identifier can be substituted.
- **https://www.opengis.net/def/rel/ogc/1.0/dggrs-zone-query**: The target IRI points to a zone listing resource allowing to perform queries for a particular Discrete Global Grid Reference System.
- **https://www.opengis.net/def/rel/ogc/1.0/dggrs-zone-parent**: The target IRI points to the information resource of a zone which is a parent of the link's context zone.
- **https://www.opengis.net/def/rel/ogc/1.0/dggrs-zone-child**: The target IRI points to the information resource of a zone which is an immediate child of the link's context zone.
- **https://www.opengis.net/def/rel/ogc/1.0/dggrs-zone-neighbor**: The target IRI points to the information resource of a zone which is a neighbor of the link's context zone.

Each resource representation includes an array of links. Implementations are free to add additional links for all resources provided by the Web API implementing OGC API — DGGS requirement(s).

Used in combination OGC API — Common — Part 1: Core, other link relation types will be used, including:

- **https://www.opengis.net/def/rel/ogc/1.0/conformance**: Refers to a resource that identifies the specifications that the link's context conforms to.

Used in combination with OGC API — Common — Part 2: Geospatial Data, other link relation types will be used, including:

- **https://www.opengis.net/def/rel/ogc/1.0/data**: Refers to the list of collections available for a dataset.

5.3. Use of HTTPS

For simplicity, the OGC API — DGGS Standard only refers to the HTTP protocol. This is not meant to exclude the use of HTTPS and simply is a shorthand notation for “HTTP or HTTPS.” Following the recent push towards enhancing web security, public facing web servers are expected to use HTTPS rather than HTTP. In the context of the OGC API — DGGS Standard, use of HTTPS provides a layer of security to prevent leaking of information related to users requesting DGGS-structured data or zones list for a particular geographical area, which may be private or sensitive in nature (for example, their own location, or the location of endangered species).

6

OVERVIEW

6 OVERVIEW

A conforming implementation of OGC API — DGGS supports either or both the Zone Data Retrieval requirements class answering questions of the type ***"What is here?"***, and/or the Zone Query requirements class answering questions of the type ***"Where is it?"***. The resource paths defined by these two requirements classes are tied to information resources defined in the Core requirements class that the implementation also needs to support.

Both of these types of request are always made in the context of a specific Discrete Global Grid Reference System (DGGRS). Three examples of DGGRSs which can be used with this DGGS API are presented in Annex B. An implementation of OGC API — DGGS may support one or more DGGRS. Requests using the native DGGRS of the data would normally provide better performance. Because the use of any DGGRS implies the concepts of hierarchical refinement levels and global coverage, having the data stored in any DGGRS should still be practical to support queries using any other DGGRS supported by the implementation with sub-optimal but acceptable performance.

> ***IMPORTANT***
>
> *This DGGS API is intended primarily for client / server exchanges where both parties are familiar and can efficiently deal with the same DGGRS. Use cases where the client itself also acts as a server as part of data integration workflows are particularly relevant, where the combination of advanced spatial queries with the ability to request data for specific zones of interest can potentially greatly accelerate analytic processing through parallel distributed processing and lazy evaluation.*

As a fallback, a client without native support for a particular DGGRS can still retrieve and interpret data from a server implementing support for both zone queries and data retrieval by performing a two-steps operation. First, the client can perform a zone query for an area of interest using the `bbox` query parameter. Then, the client can request data for the zones returned by this query in a georeferenced encoding which it understands and which is not tightly coupled with the DGGRS (excluding e.g., the DGGS-JSON encoding), such as GeoJSON or GeoTIFF. However, for such use cases, OGC API — DGGS does not present significant advantages over other OGC API data access mechanisms such as OGC API — Coverages, OGC API — Features or OGC API — Tiles.

An implementation of OGC API — DGGS may also support other OGC API access mechanisms for the same data collection for this purpose, while still leveraging the advantages of Discrete Global Grid Systems under the hood to respond effectively and in a scalable manner to requests for these other APIs.

Although this Standard defines a number of possible encodings for retrieving Zone Data and encodings for querying zone lists, no particular encoding of Zone Data is mandated by this DGGS API, whereas only the simple JSON representation of Zone Query responses (which consists of an array of textual zone identifiers) is required.

Servers may support additional representations of zone lists beyond JSON, such as a more compact binary 64-bit integers encoding, a GeoJSON representation including zone geometry or an HTML representation for exploring datasets in a browser through the DGGS API. Note that regardless of encoding, zone query responses can also be compressed for additional bandwidth efficiency if the client and server negotiate a content encoding e.g., using `Content-Encoding: gzip`.

> **CAUTION**
>
> *The transfer of zone geometry in zone queries is not an efficient implementation use of the DGGS API for practical analytics purposes. This is because clients should already have built-in knowledge of the selected DGGRS in their local software libraries and can efficiently infer the geometry of a zone from its zone ID alone as needed. In many use cases, the geometry may not be needed at all.*

The ability to retrieve zone lists in a geospatial data format such as GeoJSON or GeoTIFF which can readily be visualized in GIS tools is useful for **educational**, **demonstration** and **debugging** purposes. For example, this functionality can be used to generate grids, such as those seen in Annex B (a `crs` parameter to request an ISEA planar projection was used in that case to override the default GeoJSON CRS84), as well as the zone visualizations seen in the examples in Annex C.

Implementers may wish to start by reading the Annex C: Examples, which presents use cases alongside example requests and responses, and refer to the requirements as they implement or review the associated functionality.

Readers are also encouraged to refer as needed to OGC Abstract Specification Topic 21, which describes the conceptual foundation for this Standard.

6.1. API Definition

An example OpenAPI 3.0 definition for OGC API — DGGS is provided, defined as individual components which can be assembled with the swagger-cli tool after customizing the API definition for a particular implementation or deployment.

Developers may prefer to first read the documentation for OGC API — DGGS generated from the example OpenAPI definition, which can be visualized with SwaggerUI or ReDocly.

6.2. Resource paths, responses and parameters

The following table summarizes the resources, responses, path parameters and query parameters defined by this API, as well as in which requirements class they are defined. The table links to the relevant API definition components for the responses, parameters and schemas associated with each resource path defined by the API. Links to relevant examples in Annex C are also included.

Table 2 — Summary of DGGS API resource paths, responses and parameters, with links to relevant examples in Annex C

RESOURCE PATH	RESPONSE	PARAMETERS
Defined in requirements class Root DGGS (integration with OGC API — Common — Part 1: Core):		
`/`	Landing Page Response (OpenAPI) / Schema	
`/dggs…`	DGGS Resources (defined in Core, as well as Zone Data Retrieval and/or Zone Query) for the API deployment and/or dataset as a whole	
Defined in requirements class Collection DGGS (integration with OGC API — Common — Part 2: Geospatial Data):		
`/collections`	Collections List Response (OpenAPI) / Schema	Query parameters for discovery within a large number of collections to be defined in a proposed OGC API — Common part (optional for implementation to support): **Searchable Collections**: `limit`, `bbox`, `datetime`, `q`, `query`, `below-sd` **Hierarchical Collections:** `parent`, `descendants` **Filtering Collections with CQL2:** `filter` **Sortable Collections:** `sortby`
`/collections/{collectionId}` (see Examples C.1)	Collection Description Response (OpenAPI) / Schema	`{collectionId}` path parameter (OpenAPI)
`/collections/{collectionId}/dggs…`	DGGS Resources (defined in Core, as well as Zone Data Retrieval and/or Zone Query) for a collection	
`/collections/{collectionId}/schema` (if implementing Data subsetting)	Logical schema for the data returned for this collection (JSON Schema) to be defined	

RESOURCE PATH	RESPONSE	PARAMETERS
	in proposed OGC API — Common part, based on OGC API — Features — Part 5: Schemas	
`/collections/{collectionId}/queryables` (if implementing Filtering Zone Queries with CQL2 or Filtering Zone Data with CQL2)	Schema for the queryables that can be used in CQL2 queries for this collection (JSON Schema) as defined in OGC API — Features — Part 3: Filtering	
Defined in requirements class Core:		
`/dggrs/{dggrsId}`, **`/dggs/{dggrsId}/definition`** (example paths — could also exist on an authoritative DGGRS register) (see Annex B)	DGGRS Definition Schema	
…**`/dggs`** (see Examples C.2)	DGGRS List Response (Open API)	
…**`/dggs/{dggrsId}`** (see Examples C.3)	DGGRS Description Response (OpenAPI) / Schema	`{dggrsId}` path parameter (OpenAPI)
…**`/dggs/{dggrsId}/zones/{zoneId}`** (see Examples C.4)	Zone Information Response (OpenAPI) / Schema	`{zoneId}` path parameter (OpenAPI)
Defined in requirements class Zone Query:		

…**`/dggs/{dggrsId}/zones`**
(see Examples C.5 and Examples C.6)
	Zone Query Response (Open API)
	(defines a mandatory JSON zone list encoding)
		Required query parameters:
		`compact-zones` (see Examples C.5 and Examples C.6)
		`zone-level` (see Examples C.5.1)
		`subset` (see Examples C.5.2)
		`subset-crs`
		`bbox` (see Examples C.5.2)
		`bbox-crs`
		`datetime` (see Examples C.5.2)
		`parent-zone` (see Examples C.5.4)

RESOURCE PATH	RESPONSE	PARAMETERS
		Recommended query parameter (optional for server to support): `limit` (clients should rely on hierarchical paging with `parent-zone` and `zone-level` instead)
	Optional requirement classes complementing Zone Query	

	Some additional possible encodings defined in the following requirements classes:
	
	For practical efficient use in DGGS clients / distributed DGGS workflows:
	64-bit Binary Zone List
	
	For allowing users to explore DGGS zones / data in a hierarchical manner in the browser:
	HTML Zone List
	
	For easily visualizing zone query responses in traditional GIS software:
	GeoJSON Zone List (GeoJSON or JSON-FG)
	GeoTIFF Zone List (GeoTIFF, for DGGRSs with axis-aligned rectangular zones)
		Required by Filtering Zone Queries with CQL2:
		`filter` (CQL2 expression) (see Examples C.7)
		
		Recommended by GeoJSON Zone List (optional for server to support):
		`profile`
		`geometry`

Defined in requirements class Zone Data Retrieval:		
…/dggs/{dggrsId}/zones/{zoneId}/data (see Examples C.8 and Examples C.10)	Zone Data Response (OpenAPI) Encoding negotiated with HTTP `Accept:` request header	Recommended query parameters (optional for server to support): `crs`
	Optional requirement classes complementing Zone Data Retrieval	
	Some possible encodings defined in the following requirements classes:	**Required by Data custom depths:** `zone-depth`

RESOURCE PATH	RESPONSE	PARAMETERS

	For DGGS-quantized raster data:
	DGGS-JSON Data (DGGS-JSON Schema)
	DGGS-UBJSON Data (DGGS-JSON encoded as UBJSON)
	
	For raster data either DGGS-quantized or using traditional coordinate reference system (based on `profile` query parameter):
	netCDF Data (OGC netCDF: 3.0 / classic or 4.0 / HDF5)
	Zarr Data (zipped OGC Zarr 2.0 and/or GeoZarr)
	CoverageJSON Data (CoverageJSON)
	
	For DGGS-quantized vector data:
	DGGS-JSON-FG Data (JSON-FG with `dggsPlace`)
	DGGS-UBJSON-FG Data (DGGS-JSON-FG encoded as UBJSON)
	
	For 2D raster data (interoperable with non-DGGS clients):
	GeoTIFF Data (GeoTIFF)
	JPEG XL Data (JPEG XL)
	PNG Data (PNG)
	
	For vector data using traditional coordinate reference system (interoperable with non-DGGS clients):
	GeoJSON Data (GeoJSON or JSON-FG)
		(see Examples C.9)
		
		Required by Data subsetting:
		`datetime` (see Examples C.10.1)
		`subset` (see Examples C.10.2)
		`properties` (see Examples C.10.3)
		`exclude-properties`
		
		Required by Filtering Zone Data with CQL2:
		`filter` (CQL2 expression) (see Examples C.10.4)
		
		Recommended by DGGS-JSON-FG Data, DGGS-UBJSON-FG Data, GeoJSON Data (optional for server to support):
		`geometry` (see Examples C.8.1.3)
		
		Recommended by DGGS-JSON-FG Data, DGGS-UBJSON-FG Data, GeoJSON Data, netCDF Data, Zarr Data, CoverageJSON Data (optional for server to support):
		`profile` (see Examples C.8.1.4)
		
		Recommended by PNG Data (optional for server to support):
		`values-offset`
		`values-scale`

7 REQUIREMENTS CLASS “CORE”

7 REQUIREMENTS CLASS “CORE”

7.1. Overview

An implementation of the DGGS API “Core” Requirements Class enables a client to list available DGGRSs for a given resource, retrieve additional information about a particular DGGRS, and retrieve information about a particular DGGRS zone.

The “Core” Requirements Class is the only mandatory Requirements Class. However, an implementation with practical use is expected to additionally implement either the “Zone Data Retrieval” Requirements Class, so that a client can ask a question of the type ***“What is here?”*** and/or the “Zone Query” Requirements Class, so that a client can ask a question of the type ***“Where is it?”***

Requirements class 1: Requirements Class Core	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/core`
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.1: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/core`
NORMATIVE STATEMENTS	Requirement 1: `/req/core/dggrs-list` Requirement 1-2: `/req/core/dggrs-info` Requirement 3: `/req/core/zone-info`

7.2. Listing available DGGRSs (…/dggs)

Requirement 1	
IDENTIFIER	`/req/core/dggrs-list`
INCLUDED IN	Requirements class 1: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/core`
STATEMENT	For retrieving the list of available discrete global grid reference systems:

Requirement 1	
A	The Implementation SHALL support an HTTP GET operation at a resource path ending with .../`dggs`.
B	The Implementation SHALL support a JSON representation of this .../`dggs` resource.
C	The .../`dggs` resource SHALL include a `dggrs` array property listing all Discrete Global Grid Reference Systems (DGGRSs) supported by the Implementation.
D	Each element of the `dggrs` array SHALL include a summary description (a subset of the information available in the individual resources describing each DGGRS as per the next requirement below), including at minimum the `id`, `title`, `uri` (if applicable), and `links`.
E	The `links` property within each DGGRS element SHALL include at minimum a link to the discrete global grid reference system using the `self` link relation type, as well as a link to the discrete global grid reference system definition using the `[ogc-rel:dggrs-definition]` relation type.
F	The link relation type to use for linking from a particular resource of origin (such as a collection at /`collections/{collectionId}`, or landing page) to the list of available DGGRSs for that resource (for example at `/collections/{collectionId}/dggs` or `/dggs`) SHALL be `[ogc-rel:dggrs-list]`.

NOTE: An optional array of links may also be present at the top-level of the DGGRS list resource, for example to link back to the parent resource such as the data collection using the `[ogc-rel:geodata]` link relation.

See also Examples C.2.

7.3. Discrete Global Grid Reference System (DGGRS) description (.../dggs/{dggrsId})

Requirement 2	
IDENTIFIER	`/req/core/dggrs-description`
STATEMENT	For retrieving the description of a particular available discrete global grid reference system:
A	The Implementation SHALL support an HTTP GET operation at a resource path ending with .../`dggs/{dggrsId}`.
B	The Implementation SHALL support a JSON representation of this .../`dggs/{dggrsId}` resource.
C	The .../`dggs/{dggrsId}` resource SHALL include an `id` property consistent with the `{dggrsId}` resource path parameter.

Requirement 2	
D	The .../dggs/{dggrsId} resource SHALL include a link to the resource itself using the self link relation type.
E	The .../dggs/{dggrsId} resource SHALL include a link to a definition of the discrete global grid reference system (describing both the discrete global grid as well as the particular indexing scheme to identify zones by identifiers), using the link relation type [ogc-rel:dggrs-definition]. The schema for that definition is likely to evolve in order to support describing a growing number of classes of DGGHs. A first draft of this schema is available in Annex B — Discrete Global Grid Reference System Definitions.
F	The .../dggs/{dggrsId} resource SHALL include a templated link in the linkTemplates array to request information for a particular zone using the link relation type [ogc-rel:dggrs-zone-info] and the template variable {zoneId}.
G	If the discrete global grid reference system (the combination of the discrete global grid and indexing system) is registered with an authority, the resource SHALL include a uri property corresponding to that registered discrete global grid reference system.
H	If the discrete global grid reference system is based on a particular coordinate reference system, the resource SHALL specify that CRS in a crs property, preferably as a URI (if one is available).
I	The Implementation SHALL include a short title property identifying the discrete global grid reference system intended for display to a human.
J	The Implementation SHALL include a description property providing a summary description of the discrete global grid reference system.

Recommendation 1	
IDENTIFIER	/rec/core/max-refinement
STATEMENT	For specifying a maximum refinement level for a discrete global grid reference system associated with a particular geospatial data resource:
A	For DGGS resources associated with a data source, the Implementation SHOULD include a maxRefinementLevel integer property in the .../dggs/{dggrsId} resource specifying the maximum refinement level at which the full resolution of the data can be retrieved (using a zone-depth relative depth of 0) and/or used for performing the most accurate zone queries (using that value for zone-level).

Permission 1	
IDENTIFIER	/per/core/beyond-max-refinement
STATEMENT	For handling requests beyond maxRefinementLevel:

Permission 1	
A	The Implementation MAY return a 4xx error for data retrieval and/or zone query requests beyond a `maxRefinementLevel` specified in the DGGRS description associated with the request.
B	Alternatively, the Implementation MAY process the request by properly oversampling the data values for handling refinement levels beyond the `maxRefinementLevel`.

See also Examples C.3.

7.4. Retrieving zone information (.../dggs/{dggrsId}/zones/{zoneId})

Requirement 3	
IDENTIFIER	`/req/core/zone-info`
INCLUDED IN	Requirements class 1: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/core`
STATEMENT	For retrieving information for a particular DGGRS zone:
A	The Implementation SHALL support an HTTP GET operation at a resource path ending with .../`dggs/{dggrsId}/zones/{zoneId}` providing information for valid individual zones of the discrete global grid reference system.
B	The zone information resource SHALL support a JSON representation.
C	The zone information resource SHALL include an `id` property corresponding to the `{zoneId}` resource path parameter.
D	The zone information resource SHALL include a link back to the corresponding DGGRS resource (.../`dggs/{dggrsId}`) using the `[ogc-rel:dggrs]` link relation type.

Recommendation 2	
IDENTIFIER	`/rec/core/zone-info`
STATEMENT	For recommending additional things that should be included in zone information resources:
A	The zone information resource SHOULD include a `shapeType` property indicating the shape type of the zone's geometry (e.g., hexagon or pentagon).

Recommendation 2	
B	The zone information resource SHOULD include a `level` property indicating the refinement level of the zone.
C	The zone information resource SHOULD include a `crs` property indicating the Coordinate Reference System (CRS) in which the `geometry`, `centroid` and `bbox` property are specified.
D	The zone information resource SHOULD include a `centroid` property indicating the centroid of the zone. In the JSON representation, this should be an array of two numbers in the CRS specified in `crs`.
E	The zone information resource SHOULD include a `bbox` property indicating the extent (envelope / bounding box) of the zone. In the JSON representation, this should be an array of four (4) coordinates for a DGGRS with two spatial dimensions or six (6) coordinates for a DGGRS with three spatial dimensions, in the CRS specified in `crs`.
F	The zone information resource SHOULD include links to its parent(s) zone(s) using link relation `[ogc-rel:dggrs-zone-parent]`.
G	The zone information resource SHOULD include links to its immediate children zone using link relation `[ogc-rel:dggrs-zone-child]`.
H	The zone information resource SHOULD include links to its neighboring zones using link relation `[ogc-rel:dggrs-zone-neighbor]`.
I	The zone information resource SHOULD include an `areaMetersSquare` property indicating the surface area of the zone in square meters.
J	For a DGGS with three spatial dimension, the zone information resource SHOULD include a `volumeMetersCube` property indicating the volume of the zone in cubic meters.
K	For a temporal DGGS, the zone information resource SHOULD include a `temporalDurationSeconds` property indicating the amount of time covered by the zone in seconds.
L	The zone information resource SHOULD include a `geometry` property indicating the 2D and/or 3D spatial geometry of the zone using GeoJSON or OGC Features Geometry JSON for the JSON encoding, and including intermediate points between the vertices of the geometry so as to accurately represent the shape of the zones for DGGRSs defined in a CRS other than the CRS used to express the coordinates of the vertices. The coordinates should be in the CRS specified in the `crs` property.
M	For a temporal DGGS, the zone information resource SHOULD include a `temporalInterval` property indicating the start and end time of the zone.
N	The implementation SHOULD support a GeoJSON and/or OGC Features Geometry JSON representation of the zone information resource where the top-level object is a feature representing the zone geometry, a `zoneID` property of that feature corresponds to the textual identifier (the `{zoneId}`), the `id` of the feature corresponds to either a sequential feature identifier starting at 1, the textual identifier of the zone, or the 64-bit unsigned integer identifier of the zone, and the other properties described in this recommendation are properties of that feature.
O	For a zone associated with a particular collection, the Implementation SHOULD provide summary statistics (`minimum`, `maximum`, `average`, `stdDev`) pertaining to this zone for each field (fields of the range of a coverage, or relevant numeric properties of a feature collection) of the data. In the JSON encoding, this SHOULD be implemented as a `statistics` JSON dictionary property mapping field names to an object containing each statistic.

Recommendation 2	
P	For a zone associated with a particular collection, the Implementation SHOULD provide `areaMetersSquareWithData`, `volumeMetersCubeWithData`, `temporalDurationSecondsWithData` properties corresponding to the respective properties defined above for the overall zones, but considering only the portions of the zone where there is data (e.g., regions of the zone excluding NODATA values for a gridded coverage, or within geometry for a feature collection).

See also Examples C.4.

Recommendation 3	
IDENTIFIER	`/rec/core/robots-txt`
STATEMENT	For discouraging automated crawling of zone information resources:
A	Implementations SHOULD include a robots.txt file at the root of their Web API discouraging robots from crawling the DGGS zone resources.
B	The content of that Robots.txt file SHOULD include `Disallow: */dggs/*/zones/*` to prevent crawling all DGGS resources under the `/zones/` resource path.

NOTE: The presence of a `robots.txt` file is not a security measure and relies on the voluntary compliance of well-intended crawlers to minimize unnecessary requests. This measure does not prevent malicious clients from overwhelming the server with numerous requests which may result in Denial of Service attacks.

8

REQUIREMENTS CLASS “DATA RETRIEVAL”

8 REQUIREMENTS CLASS “DATA RETRIEVAL”

8.1. Overview

The OGC API — DGGS “Data Retrieval” requirements class allows retrieving data from a specific Discrete Global Grid Reference System (DGGRS), that is a specific hierarchy of Discrete Global Grids combined with a particular zone indexing reference system (ZIRS), for an individual zone. It describes an HTTP GET operation, as well as its response. The selected DGGRS is listed as available and described in the Core requirements class, and conforms to OGC Topic 21.

The data for a particular zone is retrieved based on a URI template including a variable representing a Zone ID.

The resource from which data is retrieved could be for a particular collection of geospatial data (Collection DGGS Requirements Class) or for a dataset as a whole (Root DGGS Requirements Class).

The collection could also be virtual in nature, as described in the *OGC API — Processes — Part 3: Workflows* candidate Standard “Collection Output” requirements class, allowing to perform on-demand processing for a spatiotemporal region of interest associated with the zone for which data is being requested.

An implementation of this requirements class enables a client to ask a question of the type ***“What is here?”***, where the *“what”* refers to data values associated with a particular dataset, collection, or processing workflow, and the *here* refers to a particular spatiotemporal region identified by the Zone ID.

Requirements class 2: Requirements Class Data Retrieval	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval`
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.2: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-retrieval`
PREREQUISITE	https://www.opengis.net/spec/ogcapi-common-1/1.0/req/core
NORMATIVE STATEMENT	Requirement 4: `/req/data-retrieval/zone-data`

8.2. Retrieving data from a zone (../dggs/{dggrsId}/zones/{zoneId}/data)

The following requirements describe how a client can retrieve data from a single DGGS zone at the resource path .../dggs/{dggrsId}/zones/{zoneId}/data.

Requirement 4	
IDENTIFIER	/req/data-retrieval/zone-data
INCLUDED IN	Requirements class 2: https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval
STATEMENT	For retrieving data for a single DGGS zone:
A	The Implementation SHALL support an HTTP GET operation at a resource path ending with .../dggs/{dggrsId}/zones/{zoneId}/data.
B	The Implementation SHALL include a templated link to this resource path in the "Core" .../dggs/{dggrsId} resource link templates, and regular link in the .../dggs/{dggrsId}/zones/{zoneId} resource links using the link relation type [ogc-rel:dggrs-zone-data] for all zones for which data is available.
C	The response of the HTTP GET operation SHALL have a status code of 200.
D	The content of the response SHALL be a data packet corresponding precisely to the area covered by the DGGS zone.
E	The selection of an encoding for the response SHALL be consistent with HTTP content negotiation.
F	The .../dggs/{dggrsId} resource SHALL include a defaultDepth property indicating the Implementation's default depth for when the zone-depth query parameter is omitted. This default value could be any valid value and/or form as defined in the /req/data-custom-depths/zone-depth-parameter requirement (single depth, range of depths, or list of depths, relative to the {zoneId} hierarchy level).
G	Unless a zone-depth query parameter is specified, the response SHALL return a data packet consistent with this defaultDepth property, in accordance with the capabilities of the negotiated data packet encoding.

See also Examples C.8.

8.3. Recommendation for a crs query parameter

The following recommendation describe how a server should support a `crs` query parameter for the …`/zones/{zoneId}/data` resource if the data can be returned in multiple applicable CRSs.

Recommendation 4	
IDENTIFIER	`/rec/data-retrieval/crs`
STATEMENT	For retrieving data in alternate CRSs:
A	If the data is available in multiple CRSs, the Implementation SHOULD support a `crs` query parameter for the resource path ending with …`/dggs/{dggrsId}/zones/{zoneId}/data`.
B	The value of this query parameter SHOULD support values for this query parameter specified either as full URIs (e.g., `http://www.opengis.net/def/crs/OGC/1.3/CRS84`) or as CURIEs (e.g., `[OGC:CRS84]`, `[EPSG:4326]`).
C	The list of supported CRSs SHOULD be included in a `crs` property of the DGGS origin resource (such as the collection description) as an array of URIs or CURIEs.

9

REQUIREMENTS CLASS “DATA SUBSETTING”

9 REQUIREMENTS CLASS “DATA SUBSETTING”

9.1. Overview

The OGC API — DGGS “Data Subsetting” requirements class extends the zone data retrieval requirements with the ability to subset data for additional dimensions beyond those defined by the discrete global grid reference systems (DGGRS), for example along time or vertical dimension if those axes are not part the DGGRS definition, or along additional dimensions such as atmospheric pressure levels. Two query parameters are defined for this purpose: `datetime` (specifically for time) and `subset` (for any dimension, including time). Additionally, this requirements class defines the `properties` query parameter to select specific fields to include in the response instead of all fields which are included by defaults.

Requirements class 3: Requirements Class Data Subsetting	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-subsetting`
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.3: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-subsetting`
PREREQUISITE	Requirements class 2: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval`
NORMATIVE STATEMENTS	Requirement 5: `/req/data-subsetting/subset` Requirement 6: `/req/data-subsetting/datetime` Requirement 7: `/req/data-subsetting/properties`

9.2. subset query parameter

Requirement 5	
IDENTIFIER	`/req/data-subsetting/subset`
INCLUDED IN	Requirements class 3: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-subsetting`

Requirement 5	
STATEMENT	For specifying a multi-dimensional subset for the zone data being retrieved (excluding dimensions of the discrete global grid reference system):

A	The Implementation SHALL support a `subset` query parameter for the zone data retrieval operation (resource path ending with .../`dggs/{dggrsId}/zones/{zoneId}/data`) conforming to the following Augmented Backus Naur Form (ABNF) fragment:
	`SubsetSpec:       "subset"=axisName(intervalOrPoint)`
	`axisName:         {text}`
	`intervalOrPoint:  interval | point`
	`interval:         low : high`
	`low:              point | *`
	`high:             point | *`
	`point:            {number} | "{text}"`
	
	`Where:`
	`\" = double quote = ASCII code 0x42,`
	`{number} is an integer or floating-point number, and`
	`{text} is some general ASCII text (such as a time and date notation in ISO 8601).`

B	The Implementation SHALL support as an axis names`time` for a temporal dataset, unless this temporal axis is an axis of the discrete global grid reference system.
C	If a third vertical spatial dimension is supported (if the resource's spatial extent bounding box is three dimensional) and that dimension is not part of the discrete global grid system definition, the Implementation SHALL also support a `h` dimension (elevation above the ellipsoid in EPSG:4979 or CRS84h) for geographic CRS and `z` for projected CRS, which are to be interpreted as the vertical axis in the CRS definition.
D	The Implementation SHALL support as axis names any additional dimension (beyond spatial and temporal) as described in the `extent` property of the collection or dataset description.
E	The Implementation SHALL return a 400 error status code if an axis name does not correspond to one of the axes of the Coordinate Reference System (CRS) of the data or an axis defined in the relevant `extent` property.
F	If a `subset` query parameter including any of the dimensions corresponding to the axes of the discrete global grid reference system is used, the server SHALL return a 400 client error.
G	The Implementation SHALL interpret multiple `subset` query parameters, as if all dimension subsetting values were provided in a single `subset` query parameter (comma separated). Example: `subset=time("2018-02-12T16:00:0Z":"2018-02-12T20:00:00Z")&subset=atm_pressure_hpa(500:750)` is equivalent to `subset=time("2018-02-12T16:00:0Z":"2018-02-12T20:00:00Z"),atm_pressure_hpa(500:750)`

NOTE: When the *interval* values fall partially outside of the range of valid values defined by the CRS for the identified axis, the service is expected to return the non-empty portion of the data resulting from the subset.

See also Examples C.10.2.

9.3. datetime query parameter

Requirement 6	
IDENTIFIER	/req/data-subsetting/datetime
INCLUDED IN	Requirements class 3: https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-subsetting
STATEMENT	For specifying a time instant or interval for which to retrieve data from a zone for a non-temporal DGGS:

A	The Implementation SHALL support a datetime query parameter expressed corresponding to either a date-time instant or a time interval, conforming to the following syntax (using ABNF):
	`interval-bounded       = instant "/" instant`
	`interval-bounded-start = [".."] "/" instant`
	`interval-bounded-end   = instant "/" [".."]`
	`interval-unbounded     = [".."] "/" [".."]`
	`interval               = interval-bounded / interval-bounded-start /`
	`interval-bounded-end / interval-unbounded`
	`datetime               = instant / interval`

B	The implementation SHALL support an instant defined as specified by RFC 3339, 5.6, with the exception that the server is only required to support the Z UTC time notation, and not required to support local time offsets.
C	Only the portions of the data within the specified interval SHALL be part of the zone data response, performing a trim operation for an interval or a slicing operation for an instant (in the case of a gridded coverage), or a filtering operation for feature data.
D	Time intervals unbounded at the start or at the end SHALL be supported using a double-dot (..) or an empty string for the start/end.
E	If a datetime query parameter is specified requesting zone data where no temporal dimension applies, the Implementation SHALL either ignore the query parameter or return a 4xx client error.

Note: ISO 8601-2 distinguishes unbounded start/end timestamps (double-dot) and unknown start/end timestamps (empty string). For queries, an unspecified start/end has the same effect as an unbounded start/end.

Examples:

Example 1 — A date-time: February 12, 2018, 23:20:52 GMT:

```
datetime=2018-02-12T23:20:52Z
```

Example 2 — Intervals: February 12, 2018, 00:00:00 GMT to March 18, 2018, 12:31:12 GMT:

```
datetime=2018-02-12T00:00:00Z/2018-03-18T12:31:12Z
```

February 12, 2018, 00:00:00 UTC or later:

```
datetime=2018-02-12T00:00:00Z/..
```

March 18, 2018, 12:31:12 UTC or earlier:

```
datetime=../2018-03-18T12:31:12Z
```

See also Examples C.10.1.

9.4. properties query parameter

Requirement 7	
IDENTIFIER	`/req/data-subsetting/properties`
INCLUDED IN	Requirements class 3: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-subsetting`
STATEMENT	For specifying fields to include when retrieving zone data
A	The zone data retrieval operation SHALL support a `properties` query parameter where the value is a comma-separated list of fields to be returned.
B	The Implementation SHALL support selecting a field using the identifier corresponding to the top-level property keys of the logical schema of the resource associated with the DGGRS zone data request, and return 400 status code for an unrecognized selected field.
C	Only the selected fields SHALL be returned from the zone data request.
D	If the zone data encodings response can self-describe its list of fields (as with the `schema` property of the DGGS-JSON encoding), the field description SHALL correspond to the requested list of fields.
E	If the negotiated format of the response has a concept of field order, then the fields SHALL be in the same order as the requested list of selected fields.

See also Examples C.10.3.

9.5. exclude-properties query parameter

Requirement 8	
IDENTIFIER	`/req/data-subsetting/exclude-properties`

Requirement 8	
STATEMENT	For including all but specific fields when retrieving zone data
A	The zone data retrieval operation SHALL support an `exclude-properties` query parameter where the value is a comma-separated list of fields not to be returned.
B	The Implementation SHALL support selecting fields not to return using the identifier corresponding to the top-level property keys of the logical schema of the resource associated with the DGGRS zone data request, and return 400 status code for an unrecognized selected field.
C	All but the selected fields SHALL be returned from the zone data request.
D	The Implementation SHALL return a 4xx error when using the both the `properties` and `exclude-properties` query parameters in the same request.

10

REQUIREMENTS CLASS “DATA CUSTOM DEPTHS”

10 REQUIREMENTS CLASS “DATA CUSTOM DEPTHS”

10.1. Overview

The OGC API — DGGS “Data Custom Depths” requirements class extends the zone data retrieval requirements class allowing a client to customize which depths to include as part of a zone data response relative to the requested hierarchy level for which data is being requested. This can be either a single depth, a range of depths, or a list of depths. The depths to include in the response are specified using the `zone-depth` query parameter. This depth customization ability can be restricted by the capability of the particular zone data response encoding negotiated.

Requirements class 4: Requirements Class Data Custom Depths	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-custom-depths`
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.4: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-custom-depths`
PREREQUISITE	Requirements class 2: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval`
NORMATIVE STATEMENT	Requirement 9: `/req/data-custom-depths/zone-depth`

10.2. zone-depth query parameter

Requirement 9	
IDENTIFIER	`/req/data-custom-depths/zone-depth`
INCLUDED IN	Requirements class 4: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-custom-depths`

Requirement 9	
STATEMENT	Query parameter to specify the DGGS refinement levels beyond the specified DGGS zone's refinement level to include in the response, when retrieving data for that zone
A	The Implementation SHALL support a `zone-depth` query parameter for the HTTP GET operation on a resource path ending with .../`dggs/{dggrsId}/zones/{zoneId}/data`.
B	The Implementation SHALL accept the following types of values for the `zone-depth` query parameter: • A single positive integer value — representing a specific zone depth to return (e.g., `zone-depth=5`); • A range of positive integer values in the form "{low}-{high}" — representing a continuous range of zone depths to return (e.g., `zone-depth=1-8`); or, • A comma separated list of at least two (2) positive integer values — representing a set of specific zone depths to return (e.g., `zone-depth=1,3,7`). Some or all of these forms of the `zone-depth` query parameter may not be supported with particular data packet encodings (the data encoding may support a fixed depth, a range of depths, and/or an arbitrary selection of depths).
C	For each zone depth to be included in the response, the interpretation of a selected depth (whether requesting a single depth, a range of depths, or a list of depths) SHALL be: • 0 corresponding to a single set of range (properties / field) value(s) for the requested zone, • 1 corresponding to all zones of the next deeper hierarchy level associated with the requested zone by the indexing scheme, • .. • *n* corresponding to all zones for the *n*'th deeper level in the hierarchy level associated with the requested zone by the indexing scheme.
D	The association of zones of deeper hierarchy levels with the requested zone SHALL be based on the DGGRS reference system, which takes into consideration both the grid definition as well as the indexing system in use for the DGGS resource.
E	If a `zone-depth` is specified, the operation SHALL return the data at the refinement level(s) specified.

NOTE 1:A use case for a `zone-depth` of 0 would be to query the single set of values for a specific DGGS zone.

NOTE 2: For use cases such as visualization and performing analysis over a certain area, a non-zero `zone-depth` would normally be used to avoid an overwhelming number of server round-trips. In this case, more than a single value would be returned for each zone request, with values returned for descendent zones at `zone-depth` levels deeper than the requested zone's level. For example, requesting data for a level 10 zone with a `zone-depth` of 8 would return individual values for each level 18 zones contained within that level 10 zone being requested.

Recommendation 5	
IDENTIFIER	`/rec/data-custom-depths/beyond-max-depth`

Recommendation 5	
STATEMENT	For informing clients of a maximum relative depth:
A	For DGGS resources associated with a data source, the Implementation SHOULD include a `maxRelativeDepth` integer property in the …`/dggs/{dggrsId}` resource specifying the maximum relative depth at which data can be retrieved.
B	The Implementation SHOULD return a 4xx error for data retrieval beyond the `maxRelativeDepth` specified in the DGGRS description associated with the request.

See also Examples C.9.

11

REQUIREMENTS CLASS “FILTERING ZONE DATA WITH CQL2”

11 REQUIREMENTS CLASS “FILTERING ZONE DATA WITH CQL2”

11.1. Overview

The OGC API — DGGS “Filtering Zone Data with CQL2” requirements class defines a `filter` query parameter where an OGC Common Query Language (CQL2) expression can be used to filter the data returned for a zone, for example based on the properties / fields of the data, or for performing spatial and/or temporal intersections.

Requirements class 5: Requirements Class Filtering Zone Data with CQL2	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-cql2-filter`
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.5: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-cql2-filter`
PREREQUISITES	Requirements class 2: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval` https://www.opengis.net/spec/cql2/1.0/req/cql2-text
NORMATIVE STATEMENT	Requirement 10: `/req/data-cql2-filter/filter`

11.2. filter query parameter

The following requirements describe how a client can specify a filtering CQL2 expression to apply on a zone data retrieval request.

Requirement 10	
IDENTIFIER	`/req/data-cql2-filter/filter`
INCLUDED IN	Requirements class 5: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-cql2-filter`

Requirement 10	
STATEMENT	For specifying a CQL2 expression filtering the data returned from a zone data request:
A	The Implementation SHALL support a `filter` query parameter specified using the CQL2-Text encoding of the OGC Common Query Language for the zone data retrieval operation (resource path ending with .../`dggs/{dggrsId}/zones/data`).
B	For a rasterized representation, the data returned SHALL have null or NODATA values for sub-zones where the CQL2 expression evaluates to false when considering the geometry and the data of that sub-zone.
C	For a vector representation, the data returned SHALL only include features where the CQL2 expression evaluates to true when considering the geometry and the data of that sub-zone.
D	The CQL2 expression evaluator SHALL support the queryables declared in the JSON Schema resource linked to from the origin of the DGGRS resources using the `[ogc-rel:queryables]` link relation type.

NOTE: The available queryables would typically include the fields of the data source being returned in the response.

See also Examples C.10.4.

12

REQUIREMENTS CLASS “ZONE QUERY”

12 REQUIREMENTS CLASS “ZONE QUERY”

12.1. Overview

The OGC API — DGGS “Zone Query” requirements class supports requesting the list of zones from a specific Discrete Global Grid Reference System (DGGRS), that is a specific hierarchy of Discrete Global Grids combined with a particular zone indexing reference system (ZIRS), for which there is data available, or matching a particular query (e.g., using a filtering query parameter). This requirements class describes an HTTP GET operation, as well as its response. The selected DGGRS is listed as available and described in the Core requirements class, and conforms to OGC Topic 21. The list of zones from a Web API using this requirements class can either be for a particular collection of geospatial data, for a dataset as a whole, or in connection with *OGC API — Processes — Part 3: Workflows & Chaining*, the output of a processing workflow.

This requirements class enables a client to ask a question of the type ***“Where is it?”***. The *“where”* refers to the spatiotemporal response encoded as a list of Zone IDs. The *it* refers to either the availability of data associated with a particular collection, dataset, or processing workflow (where the result itself could simply be this spatiotemporal region), as well as the evaluation of any additional filtering query expressed as per the CQL2 filtering requirements class.

Implementations of this requirements class are required to support at minimum the JSON encoding of zones list. This encoding provides an easy to implement, reasonably efficient and interoperable mechanism to exchange lists of zone identifiers. The JSON response consists of a simple object with a `"zones"` array property listing the zone identifiers as strings. The object can also include a `"links"` objects linking back to other resources of the DGGRS.

By default, zones are returned as a compacted list of zones, which recursively replaces children zones by their parent if all children of the parent are included in the response. A client can override this default behavior by setting the `compact-zones` query parameter to *false*.

Requirements class 6: Requirements Class Zone Query	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query`
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.6: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-query`
PREREQUISITE	Requirements class 1: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/core`
NORMATIVE STATEMENTS	Requirement 11: `/req/zone-query/zones-list` Requirement 12: `/req/zone-query/json-response`

Requirements class 6: Requirements Class Zone Query
Requirement 13: /req/zone-query/zone-level Requirement 14: /req/zone-query/compact-zones Requirement 15: /req/zone-query/parent-zone Requirement 16: /req/zone-query/bbox Requirement 17: /req/zone-query/bbox-crs Requirement 18: /req/zone-query/subset Requirement 19: /req/zone-query/subset-crs Requirement 20: /req/zone-query/datetime

12.2. Listing zones (…/dggs/{dggrsId}/zones)

The following requirements describe how a client can retrieve the list of zones from which data is available at the resource path …/dggs/{dggrsId}/zones.

Requirement 11	
IDENTIFIER	/req/zone-query/zones-list
INCLUDED IN	Requirements class 6: https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query
STATEMENT	For retrieving a list of DGGS zones.
A	The Implementation SHALL support an HTTP GET operation at a resource path ending with …/dggs/{dggrsId}/zones.
B	The Implementation SHALL include a link to this resource path in the “Core” …/dggs/{dggrsId} resource links using the link relation type https://www.opengis.net/def/rel/ogc/1.0/dggrs-zone-query.
C	The response of the HTTP GET operation SHALL have a status code of 200.
D	The content of the response SHALL be a list of zones fully covering where data is available (in the case where the resource is associated with a particular dataset), and matching any additional query parameters specified by the client (e.g., a filtering query parameter), without any redundancy.
E	Unless the zones are a compact list of zones (see compact-zones query parameter), the zones returned SHALL all be of the same DGGRS hierarchy level.
F	The selection of an encoding for the returned list of zones SHALL be consistent with HTTP content negotiation.
G	The Implementation SHALL support at minimum a JSON encoding (media type application/json).

See also Examples C.5 and Examples C.6.

12.3. JSON zone list encoding

Requirement 12	
IDENTIFIER	`/req/zone-query/json-response`
INCLUDED IN	Requirements class 6: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query`
STATEMENT	For returning a list of DGGS zones encoded as JSON
A	Every 200 response of the server for zone query with the media type `application/json` SHALL be a JSON document listing the textual identifiers for all zones matching the query.
B	The schema for the JSON document SHALL follow the JSON Schema for DGGS Zone Query described below, where the zone identifiers are strings within a `zones` array property within a JSON object. For example, `{ "zones": [ "1-E-14", "1-E-15", "1-F-14", "1-F-15", "1-F-16" ] }`.
C	The `links` property SHALL include an `[ogc-rel:dggrs]` link to the Discrete Global Grid Reference System description resource.
D	The `links` property SHALL include an `[ogc-rel:dggrs-definition]` link to the DGGRS definition, using the schema defined in Annex B — DGGRS Definitions or a later version.

```
{
   "$schema" : "https://json-schema.org/draft/2020-12/schema",
   "$id" : "https://schemas.opengis.net/ogcapi/dggs/part1/1.0/openapi/schemas/
dggs-core/dggs-zones.json",
   "type": "object",
   "required": [ "zones" ],
   "properties":
   {
      "zones" : { "type": "array", "items": { "type": "string" } },
      "returnedAreaMetersSquare" : { "type": "number" },
      "returnedVolumeMetersCube" : { "type": "number" },
      "returnedVolumeMetersSquareSeconds" : { "type": "number" },
      "returnedHyperVolumeMetersCubeSeconds" : { "type": "number" },
      "links" :
      {
         "description": "Links to related resources."
         "items": { "$ref" : "https://schemas.opengis.net/ogcapi/dggs/part1/
1.0/openapi/schemas/common-core/link.yaml"
      }
   }
}
```

Listing 3 — JSON Schema for a DGGS Zone Query response

The following recommendation describes how the response should include the total surface area of the zones returned from the query (or volume / hypervolume in the case of spatiotemporal and/or 3D DGGS), excluding overlapping regions when returning compact zones.

Recommendation 6	
IDENTIFIER	`/rec/zone-query/zone-total-area`
STATEMENT	For returning the total area (or volumes) occupied by zones returned by the query
A	The Implementation SHOULD indicate the total surface area (in the case of a 2D DGGS) or total (hyper)volume occupied (in the case of a 3D or 4D DGGS) occupied by the zones being returned, adding only once any overlaps which may occur for compact zone responses where zones are non-congruent (the total surface area is the same as what would be returned for the corresponding non-compacted list of zones at the refinement level of the zone query).
B	In the JSON response, for a 2D DGGS, the Implementation SHOULD include in the response a `returnedAreaMetersSquare` property indicating the total surface area occupied by the zones being returned, adding the overlaps which may occur for compact zone responses where zones are non-congruent only once.
C	In the JSON response, For a 3D DGGS, the Implementation SHOULD include in the response a `returnedVolumeMetersCube` property indicating the total volume occupied by the zones being returned, still counting overlapping regions only once.
D	In the JSON response, For a 2D + Time DGGS, the Implementation SHOULD include in the response a `returnedVolumeMetersSquareSeconds` property indicating the total volume occupied by the zones being returned, still counting overlapping regions only once.
E	In the JSON response, For a 3D + Time DGGS, the Implementation SHOULD include in the response a `returnedHyperVolumeMetersCubeSeconds` property indicating the total hypervolume occupied by the zones being returned, still counting overlapping regions only once.

Recommendation 7	
IDENTIFIER	`/rec/zone-json/additional-links`
A	If *DGGS Zone Data Retrieval* is supported, the `linkTemplates` property SHOULD include an `[ogc-rel:dggrs-zone-data]` link with a `{zoneId}` variable to retrieve data from each of these DGGRS zones.
B	The `links` property SHOULD include an `[ogc-rel:geodata]` link for zone listing pertaining to a particular collection (for Collection DGGS requirements class).

Recommendation 8	
IDENTIFIER	`/rec/zone-json/zone-order`

Recommendation 8	
A	For responses where zones of multiple hierarchy levels are returned when `compact-zones` is true, the zones SHOULD be listed with coarser refinement levels first (larger zones).

12.4. zone-level query parameter

The following requirements describe how a client can specify the DGGRS hierarchy level at which to retrieve the list of zones.

Requirement 13	
IDENTIFIER	`/req/zone-query/zone-level`
INCLUDED IN	Requirements class 6: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query`
STATEMENT	For specifying a level at which to return a list of DGGS zones using a `zone-level` query parameter.
A	The Implementation SHALL support a `zone-level` query parameter for the zone query operation (resource path ending with .../`dggs/{dggrsId}/zones`).
B	If a compact zones list is returned (which is the default, unless the `compact-zones` query parameter is set to *false*), the zones returned in the response SHALL be of the DGGRS hierarchy level specified by the `zone-level` query parameter, or of a lower hierarchy level standing in for a compact representation of multiple zones at the requested hierarchy level.
C	If a non-compact zones list is returned (if the `compact-zones` query parameter is set to *false*), the zones returned in the response SHALL be of the DGGRS hierarchy level specified by the `zone-level` query parameter.

12.5. compact-zones query parameter

By default, implementations return a compact list of zones where children zones fully covering a parent are recursively replaced by the parent zones, allowing to express large areas in a much more compact list of zones. The following requirements describe how a client can disable returning a compact list of zones.

Requirement 14	
IDENTIFIER	`/req/zone-query/compact-zones`
INCLUDED IN	Requirements class 6: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query`
STATEMENT	For specifying whether to retrieve a list of DGGS zones using a `compact-zones` query parameter.
A	The Implementation SHALL support a Boolean `compact-zones` query parameter for the zone query operation (resource path ending with …/`dggs/{dggrsId}/zones`), where a value of `true` corresponds to the default behavior when the query parameter is not specified, and a value of `false` disables the use of compact-zones in the response.
B	When the `compact-zones` query parameter is set to *false*, the zones list response SHALL NOT be a compact list, and SHALL explicitly list every individual zone at the requested or default DGGRS hierarchy level.
C	When the `compact-zones` query parameter is set to *true* (or unspecified), the zones list response SHALL be a compact list, where children zones completely covering the area of a parent zone SHALL be replaced by that parent zone, in a recursive manner all the way to the lowest DGGRS hierarchy level.

12.6. parent-zone query parameter for hierarchical exploration

The following requirement describes how a client can specify a parent zone to only return zones within this parent zone, enabling the exploration of a large list in a hierarchical manner (in combination with `zone-level`) as multiple requests and responses.

Requirement 15	
IDENTIFIER	`/req/zone-query/parent-zone`
INCLUDED IN	Requirements class 6: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query`
STATEMENT	For specifying a parent zone within which to restrict zone listing using a `parent-zone` query parameter.
A	The Implementation SHALL support a `parent-zone` query parameter accepting a textual zone identifier.
B	When specified, the response SHALL NOT contain zones which are not this parent zone itself or a sub-zone of that zone.

Recommendation 9	
IDENTIFIER	`/rec/zone-query/zone-order`
STATEMENT	Recommendation to follow the DGGRS sub-zone ordering
A	When the `parent-zone` query parameter is used, the Implementation SHOULD return a list of zones ordered according to the canonical sub-zone ordering defined by the DGGRS (the same order used for encoding values for data retrieval in formats such as DGGS-JSON relying on a shared understanding of this order by the server/producer and client/consumer).

12.7. limit query parameter for paging (recommendation)

The following recommendation describes how a client can specify a limit to the number of zones to be returned and page through large list of zones as multiple requests and responses.

Recommendation 10	
IDENTIFIER	`/rec/zone-query/limit`
STATEMENT	For specifying a paging limit for the list of zones using a `limit` query parameter.
A	The Implementation SHOULD support a `limit` integer query parameter, with a minimum value of 1.
B	The response SHOULD not contain more zones than specified by the optional `limit` query parameter (if specified).
C	If the API definition specifies a maximum value for the `limit` query parameter, the response SHOULD not contain more zones than this maximum value.
D	If the value of the `limit` query parameter is larger than the maximum value, this SHOULD NOT result in an error (but instead be replaced by the maximum as the query parameter value).
E	If using compact zones, the parent zones SHOULD count as a single zone, rather than the number of children zones they stand in for.
F	If an implementation does not return the full list of zones for the request, a link with relation type `next` SHOULD be included in a `links` array property of the response, which a client can request to resume listing the zones.

12.8. bbox query parameter

Requirement 16	
IDENTIFIER	`/req/zone-query/bbox`
INCLUDED IN	Requirements class 6: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query`
STATEMENT	For specifying a spatial bounding box for which to return a list of DGGS zones.
A	The Implementation SHALL support a `bbox` query parameter for the zone query operation (resource path ending with …`/dggs/{dggrsId}/zones`) with the characteristics defined in the OpenAPI Specification 3.0 fragment: `bbox:` `  name: bbox` `  in: query` `  description:` `    Bounding box of the rendered map. The bounding box is provided as four or six coordinates` `    * Lower left corner, coordinate axis 1` `    * Lower left corner, coordinate axis 2` `    * Minimum value, coordinate axis 3 (optional)` `    * Upper right corner, coordinate axis 1` `    * Upper right corner, coordinate axis 2` `    * Maximum value, coordinate axis 3 (optional)` ` The coordinate reference system and axis order of the values are indicated in the ``bbox-crs`` query parameter or if the query parameter is missing in https://www.opengis.net/def/crs/OGC/1.3/CRS84` ` required: false` ` schema:` ` type: array` ` oneOf:` ` - minItems: 4` ` maxItems: 4` ` - minItems: 6` ` maxItems: 6` ` items:` ` type: number` ` format: double` ` style: form` ` explode: false`
B	`bbox` SHALL be interpreted as a comma separated list of four or six floating point numbers. If the bounding box consists of six numbers, the first three numbers are the coordinates of the lower bound corner of a three-dimensional bounding box and the last three are the coordinates of the upper bound corner. The axis order is determined by the `bbox-crs` query parameter value or longitude and latitude if the query parameter is missing (https://www.opengis.net/def/crs/OGC/1.3/CRS84 axis order for a 2D bounding box, https://www.opengis.net/def/crs/OGC/1.3/CRS84h for a 3D bounding box). For example in https://www.opengis.net/def/crs/OGC/1.3/CRS84 the order is left_lon, lower_lat, right_lon, upper_lat.

Requirement 16	
C	The returned list of zone IDs SHALL only contain zones inside or intersecting with the spatial extent of the geographical area of the bounding box.

12.9. bbox-crs query parameter

Requirement 17	
IDENTIFIER	`/req/zone-query/bbox-crs`
INCLUDED IN	Requirements class 6: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query`
STATEMENT	For specifying the CRS used for the `bbox` query parameter using the `bbox-crs` query parameter.
A	The list of zones resource SHALL support a `bbox-crs` query parameter specifying the CRS used for the `bbox` query parameter.
B	For Earth centric data, the Implementation SHALL support https://www.opengis.net/def/crs/OGC/1.3/CRS84 as a value.
C	If the `bbox-crs` is not indicated https://www.opengis.net/def/crs/OGC/1.3/CRS84 SHALL be assumed.
D	The native CRS (`storageCrs`) SHALL be supported as a value. Other conformance classes may allow additional values (see `crs` query parameter definition).
E	The CRS expressed as URIs or as safe CURIEs SHALL be supported.
F	If the `bbox` query parameter is not used, the `bbox-crs` query parameter SHALL be ignored.

12.10. subset query parameter

Requirement 18	
IDENTIFIER	`/req/zone-query/subset`
INCLUDED IN	Requirements class 6: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query`

Requirement 18	
STATEMENT	For specifying a multi-dimensional subset for which to return a list of DGGS zones.

A	The Implementation SHALL support a `subset` query parameter for the zone query operation (resource path ending with .../`dggs/{dggrsId}/zones`) conforming to the following Augmented Backus Naur Form (ABNF) fragment:
	`SubsetSpec:       "subset"=axisName(intervalOrPoint)`
	`axisName:         {text}`
	`intervalOrPoint:  interval | point`
	`interval:         low : high`
	`low:              point | *`
	`high:             point | *`
	`point:            {number} | "{text}"`
	
	`Where:`
	`   \" = double quote = ASCII code 0x42,`
	`   {number} is an integer or floating-point number, and`
	`   {text} is some general ASCII text (such as a time and date notation in ISO 8601).`

B	The Implementation SHALL support as axis names `Lat` and `Lon` for geographic CRS and `E` and `N` for projected CRS, which are to be interpreted as the best matching spatial axis in the CRS definition.
C	If a third spatial dimension is supported (if the resource's spatial extent bounding box is three dimensional), the Implementation SHALL also support a `h` dimension (elevation above the ellipsoid in EPSG:4979 or CRS84h) for geographic CRS and `z` for projected CRS, which are to be interpreted as the vertical axis in the CRS definition.
D	The Implementation SHALL support as axis names `time` for a temporal dataset.
E	The Implementation SHALL support as axis names any additional dimension (beyond spatial and temporal) as described in the `extent` property of the collection or dataset description.
F	The Implementation SHALL return a 400 error status code if an axis name does not correspond to one of the axes of the Coordinate Reference System (CRS) of the data or an axis defined in the relevant `extent` property.
G	For a CRS where an axis can wrap around, such as subsetting across the dateline (anti-meridian) in a geographic CRS, a *low* value greater than *high* SHALL be supported to indicate an extent crossing that wrapping point.
H	The Implementation SHALL interpret the coordinates as values for the named axis of the CRS specified in the `subset-crs` query parameter value or in https://www.opengis.net/def/crs/OGC/1.3/CRS84 (https://www.opengis.net/def/crs/OGC/1.3/CRS84h for vertical dimension) if the `subset-crs` query parameter is missing.
I	If the `subset` query parameter including any of the dimensions corresponding to those of the map bounding box is used with a `bbox`, the server SHALL return a 400 client error.
J	The Implementation SHALL interpret multiple `subset` query parameters, as if all dimension subsetting values were provided in a single `subset` query parameter (comma separated). Example: `subset=Lat(-90:90)&subset=Lon(-180:180)` is equivalent to `subset=Lat(-90:90),Lon(-180:180)`

NOTE 1:A subset query parameter for https://www.opengis.net/def/crs/OGC/1.3/CRS84 will read as subset=Lon(left_lon:right_lon),Lat(lower_lat:upper_lat).

NOTE 2: When the *interval* values fall partially outside of the range of valid values defined by the CRS for the identified axis, the service is expected to return the non-empty portion of the resource resulting from the subset.

NOTE 3: For the operation of returning a list of zone IDs, there normally is no value in preserving dimensionality, therefore a *slicing* operation (using the *point* notation) is usually equivalent to a *trimming* operation (using the *interval* notation) when the low and high bounds of an interval are the same. Therefore, use of the point notation is encouraged in these cases.

Recommendation 11	
IDENTIFIER	`/rec/zone-query/subset-crs-axis-names`
A	The names of the axis SHOULD be the abbreviated names of the axis in the CRS definition (e.g. the ones defined in the EPSG database).
B	'e' (in lowercase), 'X' (lowercase/uppercase) or 'Easting' (lowercase/uppercase) SHOULD be interpreted as synonymous of 'E'.
C	'n' (in lowercase) or 'Y' (lowercase/uppercase) or 'Northing' (lowercase/uppercase) SHOULD be interpreted as synonymous of 'N'.
D	'Long' (lowercase/uppercase) or 'Longitude' SHOULD be interpreted as synonymous of 'Lon'.
E	'Latitude' SHOULD be interpreted as synonymous of 'Lat'.

12.11. subset-crs query parameter

Requirement 19	
IDENTIFIER	`/req/zone-query/subset-crs`
INCLUDED IN	Requirements class 6: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query`
STATEMENT	For specifying the CRS used for the `subset` query parameter using the `subset-crs` query parameter.
A	The zone listing operation SHALL support a query parameter `subset-crs` identifying the CRS in which the `subset` query parameter is specified with a URI or safe CURIE.

Requirement 19	
B	For Earth centric data, https://www.opengis.net/def/crs/OGC/1.3/CRS84 as a value SHALL be supported.
C	If the `subset-crs` is not indicated, https://www.opengis.net/def/crs/OGC/1.3/CRS84 SHALL be assumed.
D	The native CRS (`storageCrs`) SHALL be supported as a value. Other requirements classes may allow additional values (see `crs` query parameter definition).
E	CRSs expressed as URIs or as safe CURIEs SHALL be supported.
F	If no `subset` query parameter referring to an axis of the CRS is used, the query `subset-crs` SHALL be ignored.

12.12. datetime query parameter

Requirement 20	
IDENTIFIER	`/req/zone-query/datetime`
INCLUDED IN	Requirements class 6: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query`
STATEMENT	For specifying a multi-dimensional subset for which to return a list of DGGS zones.

A	The Implementation SHALL support a `datetime` query parameter expressed corresponding to either a date-time instant or a time interval, conforming to the following syntax (using ABNF):
	`interval-bounded        = instant "/" instant`
	`interval-bounded-start = [".."] "/" instant`
	`interval-bounded-end   = instant "/" [".."]`
	`interval-unbounded     = [".."] "/" [".."]`
	`interval               = interval-bounded / interval-bounded-start /`
	`interval-bounded-end / interval-unbounded`
	`datetime               = instant / interval`

B	The implementation SHALL support an `instant` defined as specified by RFC 3339, 5.6, with the exception that the server is only required to support the `Z` UTC time notation, and not required to support local time offsets.
C	Only the zones with data whose geometry intersect with the specified temporal interval SHALL be part of the zone list response.
D	Time intervals unbounded at the start or at the end SHALL be supported using a double-dot (`..`) or an empty string for the start/end.
E	If a `datetime` query parameter is specified requesting zone data where no temporal dimension applies, the Implementation SHALL either ignore the query parameter or return a 4xx client error.

13

REQUIREMENTS CLASS “FILTERING ZONE QUERIES WITH CQL2”

13 REQUIREMENTS CLASS “FILTERING ZONE QUERIES WITH CQL2”

13.1. Overview

The OGC API — DGGS “Filtering Zone Queries with CQL2” requirements class defines a `filter` query parameter where an OGC Common Query Language (CQL2) expression can be used to filter the list of returned zones. Such CQL2 queries enable filtering based on the properties / fields of the data, or on spatiotemporal intersections.

Requirements class 7: Requirements Class Filtering Zone Queries with CQL2	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query-cql2-filter`
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.7: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-query-cql2-filter`
PREREQUISITES	Requirements class 6: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query` https://www.opengis.net/spec/cql2/1.0/req/cql2-text
NORMATIVE STATEMENT	Requirement 21: `/req/zone-query-cql2-filter/filter`

13.2. filter query parameter

The following requirements describe how a client can specify a filtering CQL2 expression querying a list of zones.

Requirement 21	
IDENTIFIER	`/req/zone-query-cql2-filter/filter`
INCLUDED IN	Requirements class 7: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query-cql2-filter`

Requirement 21	
STATEMENT	For specifying a CQL2 expression filtering the zones returned from a zone query:
A	The Implementation SHALL support a `filter` query parameter specified using the CQL2-Text encoding of the OGC Common Query Language for the zone query operation (resource path ending with .../`dggs/{dggrsId}/zones`).
B	The list of returned zones SHALL only be those for which the CQL2 expression evaluates to true when considering the geometry and the data of the DGGS zones resource being queried.
C	The CQL2 expression evaluator SHALL support the queryables declared in the JSON Schema resource linked to from the origin of the DGGRS resources using the `[ogc-rel:queryables]` link relation type.

NOTE: The available queryables would typically include the fields of the data source, which may also be retrieved using the Data Retrieval requirements class.

See also Examples C.7.

14

REQUIREMENTS CLASS “ROOT DGGS”

14 REQUIREMENTS CLASS “ROOT DGGS”

14.1. Overview

The OGC API — DGGS “Root DGGS” Requirements Class defines the availability of DGGS resources applying to a whole dataset or API, as defined by *OGC API — Common — Part 1: Core*.

Requirements class 8: Requirements Class Root DGGS	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/root-dggs`
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.8: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/root-dggs`
PREREQUISITE	https://www.opengis.net/spec/ogcapi-common-1/1.0/req/core
NORMATIVE STATEMENT	Requirement 22: `/req/root-dggs/dggs`

14.2. Root DGGS (/dggs)

Requirement 22	
IDENTIFIER	`/req/root-dggs/dggs`
INCLUDED IN	Requirements class 8: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/root-dggs`
STATEMENT	For API/dataset-wide DGGS resources:
A	The Implementation SHALL support the resource paths and associated HTTP methods defined in the “Core” requirements class, as well as any additional resources defined in other requirements classes to which the Implementation declares conformance, for the root of the API as an origin. If the API allows to retrieve and/or query data, this means that the data returned or queried considers all collections comprising the dataset (unless an extension is used to select specific collections e.g., using

Requirement 22	
	a `collections` query parameter). The root DGGRS origin can also be used in API deployment not associated with any dataset, solely for the purpose of exploring and demonstrating the DGGRS.
B	The Implementation SHALL include a link to the list of available DGGRSs for the API/dataset at `/dggs` in the links of the landing page using the link relation type `[ogc-rel:dggrs-list]`.
C	If the API deployment supports querying or retrieving data from a dataset, the `/dggs` and `/dggs/{dggrsId}` resources SHALL include a link to the landing page using the link relation type `[ogc-rel:dataset]`.

15

REQUIREMENTS CLASS “COLLECTION DGGS”

15 REQUIREMENTS CLASS “COLLECTION DGGS”

15.1. Overview

The OGC API — DGGS “Collection DGGS” Requirements Class defines the availability of DGGS resources applying to one or more collections of geospatial data, as defined by *OGC API — Common — Part 2: Geospatial data*.

Requirements class 9: Requirements Class Collection DGGS	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/collection-dggs`
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.9: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/collection-dggs`
PREREQUISITE	https://www.opengis.net/spec/ogcapi-common-2/1.0/req/collections
NORMATIVE STATEMENT	Requirement 23: `/req/collection-dggs/dggs`

15.2. Collection DGGS (/collections/{collectionId}/dggs)

Requirement 23	
IDENTIFIER	`/req/collection-dggs/dggs`
INCLUDED IN	Requirements class 9: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/collection-dggs`
STATEMENT	For collection DGGS resources:
A	The Implementation SHALL support the resource paths and associated HTTP methods defined in the “Core” requirements class, as well as any additional resources defined in other requirements classes to which the Implementation declares conformance, for at least one collection of the dataset offered by the API deployment.

Requirement 23	
B	The Implementation SHALL include a link to the list of available DGGRSs for the collection at `/collections/{collectionId}/dggs` in the links of the collection using the link relation type `[ogc-rel:dggrs-list]`.
C	The `/collections/{collectionId}/dggs` and `/collections/{collectionId}/dggs/{dggrsId}` resources SHALL include a link to the collection using the link relation type `[ogc-rel:geodata]`.

See also Examples C.1.

16 REQUIREMENTS CLASSES FOR ENCODINGS OF ZONE DATA

16 REQUIREMENTS CLASSES FOR ENCODINGS OF ZONE DATA

The OGC API — DGGS Standard does not mandate any particular encoding or format in which to return the …`/dggs/{dggrsId}/zones/{zoneId}/data` resources. DGGS Zone Data can be encoded in any suitable data format. However, the Standard does define requirements classes for zone data encodings which are expected to be commonly supported in implementations of this standard. These requirements classes include:

For data rasterized to DGGRS sub-zones:

- DGGS-JSON Data
- DGGS-UBJSON Data

Depending on a selected profile, for data either rasterized to DGGRS sub-zones, or rasterized data output readily usable in non-DGGS systems (or to facilitate interoperability with other DGGHs):

- Zarr Data
- netCDF Data
- CoverageJSON Data

For 2D rasterized data output readily usable in non-DGGS systems or to facilitate interoperability with other DGGHs:

- GeoTIFF Data
- JPEG XL Data
- PNG Data

For vector features whose geometry coordinates are quantized to DGGRS sub-zones:

- DGGS-JSON-FG Data
- DGGS-UBJSON-FG Data

For vector features output readily usable in non-DGGS systems or to facilitate interoperability with other DGGHs:

- GeoJSON Data — including support for Features & Geometry JSON (JSON-FG) output as a recommendation

16.1. Media Types (for zone data)

A table of the media types used in the encoding of the Zone Data requirements classes defined in this Standard follows.

Table 3 — Media Types used for zone data encoding requirements classes

Encoding	Media type
DGGS-JSON zone data	application/json
DGGS-UBJSON zone data	application/ubjson
DGGS-JSON-FG zone data	application/geo+json (with `profile` query parameter values of `jsonfg-dggs`, `jsonfg-dggs-plus`, `jsonfg-dggs-zoneids` or `jsonfg-dggs-zoneids-plus`)
DGGS-UBJSON-FG zone data	application/geo+ubjson (with `profile` query parameter values of `jsonfg-dggs`, `jsonfg-dggs-plus`, `jsonfg-dggs-zoneids` or `jsonfg-dggs-zoneids-plus`)
GeoJSON / JSON-FG data	application/geo+json (without `profile` query parameter, or with values of `rfc7946`, `jsonfg` or `jsonfg-plus`)
GeoTIFF data	image/tiff; application=geotiff
netCDF zone data	application/netcdf (rasterized to sub-zones for `profile` query parameter values of `netcdf3-dggs`, `netcdf3-dggs-zoneids`, `netcdf4-dggs`, `netcdf4-dggs-zoneids`)
Zarr zone data	application/zarr+zip (rasterized to sub-zones for `profile` query parameter values of `zarr2-dggs` or `zarr2-dggs-zoneids`)
CoverageJSON zone data	application/prs.coverage+json (rasterized to sub-zones for `profile` query parameter values of `covjson-dggs` or `covjson-dggs-zoneids`)
PNG zone data	image/png
JPEG XL zone data	image/jxl

16.2. Requirements Class “DGGS-JSON Zone data encoding” (for data rasterized to sub-zones)

16.2.1. Overview

The DGGS-JSON Zone Data encoding requirements class defines support for encoding a DGGS Zone data response in JSON, using a schema defined below named “DGGS-JSON”.

This encoding is intended to support one-to-one mapping of sub-zones values, for one or more zone depths, regardless of zone geometry type, in a human-readable format.

Values are encoded as a one-dimensional array following a sub-zone order defined by the DGGRS.

Example encodings follow:


```
{
   "$schema" : "https://schemas.opengis.net/ogcapi/dggs/part1/1.0/openapi/
schemas/dggs-json/schema",
   "dggrs": "https://www.opengis.net/def/dggrs/OGC/1.0/ISEA3H",
   "zoneId": "C0-2B-A",
   "depths": [ 0, 1 ],
   "schema":
   {
      "$schema" : "https://json-schema.org/draft/2020-12/schema",
      "$id" : "https://example.com/ogcapi/collections/climate/schema",
      "title" : "Temperature",
      "type": "object",
      "properties":
      {
         "t": {
            "type": "number",
            "title": "air temperature in celsius",
            "x-ogc-propertySeq" : 1,
            "x-ogc-definition": "https://qudt.org/vocab/quantitykind/
Temperature",
            "x-ogc-unit": "C"
         }
      }
   },
   "values":
   {
      "t": [
         {
            "depth": 0,
            "shape": { "count": 1, "subZones": 1 },
            "data": [ 22.8 ]
         },
         {
            "depth": 1,
            "shape": { "count": 7, "subZones": 7 },
            "data": [ 25.1, 25.8, 22.0, 23.1, 22.8, 20.0, 17.1 ]
         }
      ]
```

```
   }
}
```

Listing 7 — Example encoding for DGGS-JSON

```
{
   "$schema" : "https://schemas.opengis.net/ogcapi/dggs/part1/1.0/openapi/
schemas/dggs-json/schema",
   "dggrs": "https://www.opengis.net/def/dggrs/OGC/1.0/ISEA3H",
   "zoneId": "C0-2B-A",
   "depths": [ 0, 1 ],
   "schema":
   {
      "$schema" : "https://json-schema.org/draft/2020-12/schema",
      "$id" : "https://example.com/ogcapi/collections/climate/schema",
      "title" : "Climate Variables",
      "type": "object",
      "properties":
      {
         "rh": {
            "type": "number",
            "title": "relative humidity",
            "x-ogc-propertySeq" : 1,
            "x-ogc-definition": "https://qudt.org/vocab/quantitykind/
RelativeHumidity"
         },
         "t": {
            "type": "number",
            "title": "air temperature in celsius",
            "x-ogc-propertySeq" : 2,
            "x-ogc-definition": "https://qudt.org/vocab/quantitykind/
Temperature",
            "x-ogc-unit": "C"
         },
         "ua": {
            "type": "number",
            "title": "Eastward wind",
            "x-ogc-propertySeq" : 3,
            "x-ogc-definition": "https://qudt.org/vocab/quantitykind/Speed",
            "x-ogc-unit": "m/s"
         },
         "va": {
            "type": "number",
            "title": "Northward wind",
            "x-ogc-propertySeq" : 4,
            "x-ogc-definition": "https://qudt.org/vocab/quantitykind/Speed",
            "x-ogc-unit": "m/s"
         }
      }
   },
   "dimensions" : [
      {
         "name": "time",
         "interval" : [ "2020-01-01", "2020-04-30" ],
         "grid" : { "cellsCount" : 4, "firstCoordinate" : "2020-01-01",
"resolution" : "P1M" }
      },
      {
         "name": "pressure",
         "definition": "https://qudt.org/vocab/quantitykind/
AtmosphericPressure",
         "unit" : "hPa",
         "interval" : [ 850.0, 50.0 ],
```

```
         "grid" : { "cellsCount" : 3, "coordinates" : [ 850.0, 250.0, 50.0 ] }
      }
   ],
   "values":
   {
      "rh": [
         {
            "depth": 0,
            "shape": { "count": 9, "subZones": 1, "dimensions": { "time": 4,
"pressure": 3 } },
            "data": [
               25.1, 25.8, 22.0,   23.1, 22.8, 20.0,   17.1, 12.8, 14.0,
24.1, 13.8, 12.0
            ]
         },
         {
            "depth": 1,
            "shape": { "count": 84, "subZones": 7, "dimensions": { "time": 4,
"pressure": 3 } },
            "data": [
               23.2, 25.8, 22.0,   22.8, 20.0, 22.0,   17.1, 12.8, 14.0,
23.1, 17.0, 24.0,
               22.1, 23.8, 22.0,   28.0, 23.2, 22.8,   18.0, 22.0, 17.1,
12.8, 12.0, 17.0,
               25.2, 25.8, 22.0,   27.0, 23.1, 22.8,   21.0, 20.0, 16.1,
12.8, 14.0, 12.0,
               24.1, 24.8, 21.0,   23.0, 23.3, 22.8,   23.0, 22.0, 17.1,
12.8, 13.0, 17.0,
               25.1, 25.8, 22.0,   21.0, 22.1, 22.8,   22.0, 21.0, 13.1,
12.8, 14.0, 14.0,
               26.1, 25.8, 20.0,   22.0, 23.5, 22.8,   21.0, 22.0, 17.1,
12.8, 11.0, 17.0,
               24.1, 25.4, 20.3,   23.0, 22.5, 22.9,   20.9, 21.7, 17.4,
12.9, 11.5, 17.9
            ]
         }
      ],
      "t": [ ... ], "ua": [ ... ], "va": [ ... ]
   }
}
```

Listing 8 — A more complex example encoding for DGGS-JSON with more variables and additional dimensions beyond those of the DGGRS

Requirements class 10: Requirements Class DGGS-JSON Zone Data	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-json`
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.10: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-json`
PREREQUISITES	JSON Requirements class 2: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval`

Requirements class 10: Requirements Class DGGS-JSON Zone Data	
NORMATIVE STATEMENT	Requirement 24: `/req/data-json/content`

16.2.2. DGGS-JSON Zone Data

Requirement 24	
IDENTIFIER	`/req/data-json/content`
INCLUDED IN	Requirements class 10: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-json`
STATEMENT	For encoding zone data as DGGS-JSON:
A	Every 200 response of the server for zone data with the media type `application/json` SHALL be a JSON document representing the data values for all selected fields of each included sub-zone.
B	The schema for the JSON document SHALL follow the JSON Schema for DGGS-JSON described below.
C	Every zone depth requested using the `zone-depth` query parameter SHALL be included in the response.
D	At every depth, each individual value SHALL correspond exactly to the data sampled representative of that sub-zone.
E	The list of data values SHALL follow the default zone order as specified by the Discrete Global Grid Reference System (for example based on a scanline or space-filling curved defined therein) for which the request is made.
F	Null values SHALL use the `null` JSON value.

```
{
   "$schema" : "https://json-schema.org/draft/2020-12/schema",
   "$id" : "https://schemas.opengis.net/ogcapi/dggs/part1/1.0/openapi/schemas/
dggs-json/schema",
   "type": "object",
   "properties":
   {
      "dggrs" : { "type": "string", "format": "uri" },
      "zoneId" : { "type": "string" },
      "depths" : { "type": "array", "items": { "type": "integer", "minimum": 0
} },
      "schema" : { "$ref": "https://json-schema.org/draft/2020-12/schema" },
      "dimensions" :
      {
         "type": "array",
         "items":
         {
            "type": "object",
            "properties":
```

```json
{
   "name": { "type": "string" },
   "definition": { "type": "string", "format": "uri" },
   "unit": { "type": "string" },
   "unitLang": { "type": "string", "format": "uri" },
   "grid":
   {
      "required": [ "cellsCount" ],
      "oneOf": [
          { "required": [ "resolution", "firstCoordinate" ] },
          { "required": [ "coordinates" ] }
      ],
      "type": "object",
      "properties":
      {
         "boundsCoordinates": {
            "items": {
               "type": "array",
               "items": {
                  "oneOf": [
                     { "nullable": true, "type": "number" },
                     { "nullable": true, "type": "string" }
                  ]
               },
               "maxItems": 2,
               "minItems": 2
            },
            "type": "array"
         },
         "cellsCount" : { "type": "integer" },
         "resolution" : {
            "oneOf": [
               { "type": "number" },
               { "type": "string" }
            ]
         },
         "firstCoordinate": {
            "oneOf": [
               { "type": "number" },
               { "type": "string" }
            ]
         },
         "relativeBounds": {
            "items": {
               "oneOf": [
                  { "type": "number" },
                  { "type": "string" }
               ]
            },
            "maxItems": 2,
            "minItems": 2,
            "type": "array"
         },
         "coordinates": {
            "type": "array",
            "items": {
               "oneOf": [
                  { "type": "number" },
                  { "type": "string" }
               ]
            }
         }
      }
   }
```

```
               },
               "interval": {
                  "type": "array",
                  "maxItems": 2,
                  "minItems": 2,
                  "items": {
                     "oneOf": [
                        { "type": "number", "nullable": true },
                        { "type": "string", "nullable": true }
                     ]
                  }
               }
            },
            "required": [ "name", "interval", "grid" ]
         }
      },
      "values" :
      {
         "additionalProperties":
         {
            "type": "array",
            "items":
            {
               "type": "object",
               "required": [ "data" ],
               "properties":
               {
                  "depth": { "type": "integer" },
                  "shape":
                  {
                     "type": "object",
                     "properties":
                     {
                        "count": { "type": "integer" },
                        "subZones": { "type": "integer" },
                        "dimensions":
                        {
                           "type": "object",
                           "additionalProperties": { "type": "integer" }
                        }
                     },
                     "required": [ "count", "subZones" ]
                  },
                  "data":
                  {
                     "type": "array",
                     "items":
                     {
                        "type": "number",
                        "nullable": true
                     }
                  }
               }
            }
         }
      }
   },
   "required": [ "dggrs", "zoneId", "depths", "values" ]
}
```

Listing 9 — JSON Schema for DGGS-JSON

> **IMPORTANT**
>
> *For DGGS-JSON containing additional dimensions, the one-dimensional array of `values` is ordered in a sub-zone order major manner. This is so all values of a particular field for a given sub-zone are regrouped together. Values for the first dimensions listed in the `dimensions` array are then regrouped together, followed by additional dimensions as they appear in order in the dimensions array.*

16.3. Requirements Class "DGGS-UBJSON Zone data encoding" (for data rasterized to sub-zones)

16.3.1. Overview

The DGGS-UBJSON Zone Data encoding requirements class defines support for encoding a DGGS Zone data response using the same JSON structure as the above JSON data requirements class but using the binary UBJSON encoding.

Requirements class 11: Requirements Class DGGS-UBJSON	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-ubjson`
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.11: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-ubjson`
PREREQUISITES	https://ubjson.org/ Requirements class 2: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval`
NORMATIVE STATEMENT	Requirement 25: `/req/data-ubjson/content`

16.3.2. DGGS-UBJSON Zone Data

Requirement 25	
IDENTIFIER	`/req/data-ubjson/content`

Requirement 25	
INCLUDED IN	Requirements class 11: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-ubjson`
STATEMENT	For encoding zone data as DGGS-UBJSON
A	Every 200 response of the server for zone data with the media type `application/ubjson` SHALL be a Universal Binary JSON document representing the data values for all selected fields of each included sub-zone.
B	The schema for the UBJSON document SHALL follow the same JSON Schema as for DGGS-JSON described above.
C	Every zone depth requested using the `zone-depth` query parameter SHALL be included in the response.
D	At every depth, each individual value SHALL correspond exactly to the data sampled representative of that zone geometry.
E	The list of data values SHALL follow the default zone order as specified by the Discrete Global Grid Reference System (for example based on a scanline or space-filling curved defined therein) for which the request is made.
F	Null values SHALL use the `null` JSON value.

The py-ubjson tool can be used to easily convert between JSON and UBJSON. After installing with `pip3 install py-ubjson`, a DGGS-JSON file can be converted to DGGS-UBJSON with `ubjson fromjson zoneData.dggs.json`, and a DGGS-UBJSON file can be converted to DGGS-JSON with `ubjson tojson zoneData.dggs.ubj`.

16.4. Requirements Class "DGGS-JSON-FG Zone data encoding" (for geometry coordinates quantized to sub-zones)

16.4.1. Overview

The DGGS-JSON-FG Zone Data encoding requirements class defines support for encoding a DGGS Zone data response for vector features in JSON. This is accomplished by extending OGC Features & Geometry JSON with the concepts of DGGRS, reference parent zone, relative depth and deterministic sub-zone order, allowing to encode coordinates pairs as local sub-zone indices. Alternatively, coordinates can be encoded using global zone IDs.

DGGS-JSON-FG Zone Data is intended to support efficient encoding of vector geometry at arbitrary precision using finer grids to locally address the inside of the reference parent zones. This is done in a manner agnostic of the DGGRS (which needs to define a deterministic sub-zone order), in a human-readable format sharing as much as possible with existing practice for exchanging geospatial vector data in JSON.

An Implementation may support a DGGS-JSON-FG representation where DGGS-quantized coordinates are specified either as global zone identifiers or as local indices into the deterministic zone order, and may also support GeoJSON / JSON-FG compatibility modes where coordinates are additionally specified as positions on spatiotemporal axes for interoperability with non-DGGS clients.

A `profile` query parameter can be used by clients to request a particular DGGS-JSON-FG representation from Implementations supporting multiple options:

- `jsonfg-dggs`: DGGS-JSON-FG with `dggsPlace` coordinates corresponding to local sub-zone indices
- `jsonfg-dggs-plus`: DGGS-JSON-FG with `dggsPlace` coordinates corresponding to local sub-zone indices, in addition to GeoJSON (`geometry`) and/or JSON-FG compatibility mode (`place`)
- `jsonfg-dggs-zoneids`: DGGS-JSON-FG with `dggsPlace` coordinates corresponding to global textual identifiers
- `jsonfg-dggs-zoneids-plus`: DGGS-JSON-FG with `dggsPlace` coordinates corresponding to global textual identifiers, in addition to GeoJSON (`geometry`) and/or JSON-FG compatibility mode (`place`)

Example encodings follow:


```
{
   "conformsTo": [
      "https://www.opengis.net/spec/json-fg-1/0.2/conf/core",
      "https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-dggs-fgjson"
   ],
   "links": [
      {
         "rel" : "profile",
         "href" : "https://www.opengis.net/def/profile/ogc/0/jsonfg-dggs"
      }
   ],
   "dggrs": "[ogc-dggrs:ISEA3H]",
   "zoneId": "I0-1EFA652-D",
   "depth": "15",
   "type" : "FeatureCollection",
   "features" : [ {
      "type" : "Feature",
      "id" : 2305843009218664049,
      "geometry" : null,
      "place" : null,
      "time" : null,
      "dggsPlace" : {
         "type" : "LineString",
         "coordinates" : [ 13866054, 13899376, 13953688 ]
      },
```

```
      "properties" : {
         "highway" : "service",
         "name" : "Parliament Road",
         "name:fr" : "chemin Parliament",
         "access" : "no"
      }
   }, {
      "type" : "Feature",
      "id" : 2305843009218664071,
      "geometry" : null,
      "place" : null,
      "time" : null,
      "dggsPlace" : {
         "type" : "LineString",
         "coordinates" :
   ...
```

Listing 10 — Example encoding of DGGS-JSON-FG for OpenStreetMap data

```
{
   "conformsTo": [
      "https://www.opengis.net/spec/json-fg-1/0.2/conf/core",
      "https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-dggs-fgjson"
   ],
   "links": [
      {
         "rel" : "profile",
         "href" : "https://www.opengis.net/def/profile/ogc/0/jsonfg-dggs"
      }
   ],
   "dggrs": "[ogc-dggrs:ISEA3H]",
   "zoneId": "B0-5-A",
   "depth": "15",
   "geometryDimension": 2,
   "type": "FeatureCollection",
   "features": [
      {
         "type": "Feature",
         "id": 1,
         "properties": {
            "scalerank": 0,
            "featurecla": "Bathymetry"
         },
         "geometry" : null,
         "place": null,
         "time" : null,
         "dggsPlace": {
            "type": "MultiPolygon",
            "coordinates": [
               [
                  [ 14355360, 14355250, 14355141, 14355032, 14354922, 14354813,
 ...
```

Listing 11 — Example encoding of DGGS-JSON-FG for Natural Earth bathymetry in the arctic

Requirements class 12: Requirements Class DGGS-JSON-FG Zone Data	
IDENTIFIER	https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-dggs-jsonfg

Requirements class 12: Requirements Class DGGS-JSON-FG Zone Data	
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.12: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-dggs-jsonfg`
PREREQUISITES	JSON Requirements class 2: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval` https://www.opengis.net/spec/json-fg-1/0.3/req/core
NORMATIVE STATEMENT	Requirement 26: `/req/data-dggs-jsonfg/content`

16.4.2. DGGS-JSON-FG Zone Data

Requirement 26	
IDENTIFIER	`/req/data-dggs-jsonfg/content`
INCLUDED IN	Requirements class 12: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-dggs-jsonfg`
STATEMENT	For encoding zone data as DGGS-JSON-FG
A	The Implementation SHALL support zone data requests negotiating an `application/geo+json` media type combined with a `profile` query parameter with a value of `jsonfg-dggs`, `jsonfg-dggs-plus`, `jsonfg-dggs-zoneids` or `jsonfg-dggs-zoneids-plus` allowing a client to request a DGGS-JSON-FG response.
B	Every 200 response of the server for DGGS-JSON-FG zone data requests SHALL be an extended Features & Geometry JSON (JSON-FG) document representing the vector features within the requested zone.
C	The JSON document SHALL follow the JSON Schema for JSON-FG, with geometry stored in a `dggsPlace` property mirroring the JSON-FG `place` schema and supporting all of the same feature types, but where `coordinates` use sub-zone indices (for `profile=jsonfg-dggs` or `jsonfg-dggs-plus`) or global textual zone identifiers (for `profile=jsonfg-dggs-zoneids` or `jsonfg-dggs-zoneids-plus`) as described in later parts of this requirement.
D	For `profile=jsonfg-dggs-plus` and `jsonfg-dggs-zoneids-plus`, the JSON document SHALL also include geometry stored in `geometry` or `place` using traditional coordinate reference system coordinates for compatibility.
E	The JSON document SHALL additionally contain a `dggrs` property set to the URI or CURIEs of the DGGRS of the response, which replaces the `coordRefSys` property and concept.
F	The JSON document SHALL additionally contain a `zoneId` property indicating the requested reference zone.

Requirement 26	
G	The JSON document SHALL additionally contain a `depth` property corresponding to the relative zone depth (as specified in the single depth format of the OGC API — DGGS `zone-depth` query parameter for zone data retrieval), determining how coordinates are resolved from sub-zone indices (in the case of `profile=jsonfg-dggs` and `jsonfg-dggs-plus`), and as a result, their precision.
H	For `profile=jsonfg-dggs` and `jsonfg-dggs-plus`, the JSON document SHALL contain a `dggsPlace` property specifying geometry using local sub-zone indices ranging from 1 to the number of sub-zones within the reference parent zone at the declared depth, based on the deterministic sub-zone order of the DGGRS.
I	For `profile=jsonfg-dggs-zoneids` and `jsonfg-dggs-zoneids-plus`, the JSON document SHALL contain a `dggsPlace` property specifying geometry using global textual zone identifiers (as defined in the DGGRS zone indexing reference system) at the declared depth.
J	When all spatial coordinates are dimensions of the DGGRS, the array of coordinates including the square brackets SHALL be replaced by the local index or global identifier of the sub-zone within which the coordinates are located.
K	When additional spatial coordinates that are not part of the DGGRS, such as elevation above ground for a 2D DGGRS, are needed to define the geometry coordinates, only the coordinates corresponding to DGGRS dimensions SHALL be replaced by the local index or global identifier of the sub-zone.
L	Only features fully or partly located within the reference zone SHALL be included.
M	For feature geometry intersecting the boundaries of the reference zone, the sub-zone within which the intersection is located SHALL be included, with a separate single `0` special sub-zone index (in the case of `profile=jsonfg-dggs` and `jsonfg-dggs-plus`) or a `null` value (in the case of `profile=jsonfg-dggs-zoneids` and `jsonfg-dggs-zoneids-plus`) representing all outside points before a point inside, after a point inside, or in between two inside points.
N	For geometry in two or more dimensions with an outside point (identified by a `0` sub-zone index for `profile=jsonfg-dggs` and `jsonfg-dggs-plus`, or `null` for `profile=jsonfg-dggs-zoneids` and `jsonfg-dggs-zoneids-plus`) and which is necessarily between two contour points, the portion of the zone edge(s) between the exit and entry point SHALL be considered inside of that contour.
O	The URI of the selected or default DGGS-JSON-FG profile SHALL be included in the `links` section of the JSON-FG response.

Recommendation 12	
IDENTIFIER	`/rec/data-dggs-jsonfg/profile-links`
A	The implementation SHOULD return a link to the default or selected profile as a `Link` HTTP response header, for example `Link: https://www.opengis.net/def/profile/ogc/0/jsonfg-dggs; rel="profile"`.

Recommendation 13	
IDENTIFIER	`/rec/data-dggs-jsonfg/geometry`
A	The implementation SHOULD support a `geometry` query parameter allowing a client to choose how to return feature geometry.
B	The implementation SHOULD support a value of `geometry=zone-centroid` to request a representation where each feature is a zone intersecting the data, with the geometry of each feature being a Point geometry for the centroid of that zone.
C	The implementation SHOULD support a value of `geometry=vectorized` to request a representation where each feature corresponds to a feature of the data (sharing identical property values).
D	If a `geometry` value is not specified, the Implementation SHOULD return the representation closest to the native data.
E	If a requested geometry representation is not supported, the Implementation SHOULD return a 4xx HTTP error.

Permission 2	
IDENTIFIER	`/per/data-dggs-jsonfg/supported-profiles`
A	An Implementation MAY support only a subset of the defined DGGS-JSON-FG profiles.
B	An Implementation MAY return any profile as the default when the `profile` query parameter is not specified in the request. As an example, this allows for static servers to return only a DGGS-JSON-FG representation.

16.5. Requirements Class “DGGS-UBJSON-FG Zone data encoding” (for geometry coordinates quantized to sub-zones)

16.5.1. Overview

The DGGS-UBJSON-FG Zone Data encoding requirements class defines support for encoding a DGGS Zone data response for vector features using the same JSON structure as the above DGGS-JSON-FG data requirements class, but using the binary UBJSON encoding.

Requirements class 13: Requirements Class DGGS-UBJSON-FG	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-dggs-ubjsonfg`
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.13: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-dggs-ubjsonfg`
PREREQUISITES	https://ubjson.org/ Requirements class 2: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval` https://www.opengis.net/spec/json-fg-1/0.3/req/core
NORMATIVE STATEMENT	Requirement 27: `/req/data-dggs-ubjsonfg/content`

16.5.2. DGGS-UBJSON-FG Zone Data

Requirement 27	
IDENTIFIER	`/req/data-dggs-ubjsonfg/content`
INCLUDED IN	Requirements class 13: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-dggs-ubjsonfg`
STATEMENT	For encoding zone data as DGGS-UBJSON-FG
A	The Implementation SHALL support zone data requests negotiating an `application/geo+ubjson` media type combined with a `profile` query parameter with a value of `jsonfg-dggs`, `jsonfg-dggs-plus`, `jsonfg-dggs-zoneids` or `jsonfg-dggs-zoneids-plus` allowing a client to request a DGGS-UBJSON-FG response.
B	Every 200 response of the server for DGGS-UBJSON-FG zone data requests SHALL be a Universal Binary JSON document representing the vector features within the requested zone
C	The content of the UBJSON document SHALL follow all the of same requirements as for DGGS-JSON-FG described above.

The py-ubjson tool can be used to easily convert between JSON and UBJSON. After installing with `pip3 install py-ubjson`, a DGGS-JSON-FG file can be converted to DGGS-UBJSON-FG with `ubjson fromjson zoneData.dggsjsonfg`, and a DGGS-UBJSON-FG file can be converted to DGGS-JSON-FG with `ubjson tojson zoneData.dggsfg.ubj`.

16.6. Requirements Class “GeoTIFF Zone data encoding”

16.6.1. Overview

The GeoTIFF Zone Data encoding requirements class defines support for encoding a DGGS Zone data response according to the OGC GeoTIFF standard.

GeoTIFF is a commonly used format for representing 2-dimensional gridded coverages.

For a DGGRS with non-rectangular zones (e.g., hexagonal zones), a GeoTIFF representation can be the minimal bounding rectangle of the zone shape, with NODATA values used outside the bounds of the zone.

Requirements class 14: Requirements Class GeoTIFF Data	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-geotiff`
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.14: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-geotiff`
PREREQUISITES	TIFF V6.0 GeoTIFF Requirements class 2: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval`
NORMATIVE STATEMENT	Requirement 28: `/req/data-geotiff/content`

16.6.2. GeoTIFF Zone Data

Requirement 28	
IDENTIFIER	`/req/data-geotiff/content`
INCLUDED IN	Requirements class 14: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-geotiff`
STATEMENT	For encoding zone data as GeoTIFF
A	Every 200 response of the server for zone data with the media type `image/tiff` SHALL be a TIFF image representing the data values for all selected fields within the zone for which data is requested.

Requirement 28	
B	If the TIFF encoding incorporates a GeoTIFF georeference, this information SHALL be consistent with the DGGRS Zone ID.
C	If the zone geometry is not rectangular, the closest bounding rectangle SHALL be used for referencing the image.
D	A distinct value for each sub-zone implied from the requested `zone-depth` SHALL correspond to at least one distinct cell value in the response.
E	For Implementations supporting Data Custom Depths, each depth of the requested zone depth pyramid SHALL be a separate image (overview) in the response.

Recommendation 14	
IDENTIFIER	`/rec/data-geotiff/null-values`
A	Null values SHOULD be used for cells lying outside the zone geometry.

Recommendation 15	
IDENTIFIER	`/rec/data-geotiff/crs`
A	The CRS of the response SHOULD be consistent with either the DGGRS or the underlying geographic CRS (e.g., CRS84).

16.7. Requirements Class “GeoJSON Zone data encoding”

16.7.1. Overview

The GeoJSON Zone Data encoding requirements class defines support for encoding a DGGS Zone data response according to the GeoJSON specification, and optionally according to the OGC JSON-FG candidate Standard.

GeoJSON is a commonly used format for representing features with simple geometries and related properties. GeoJSON is simple to understand and is well supported by tools and software libraries. Since most Web developers are comfortable with using JSON-based formats, supporting GeoJSON is recommended for a vector representation of DGGS Zone data.

JSON-FG is an OGC format for representing features with simple geometries and related properties extending GeoJSON for greater flexibility.

An Implementation may support JSON-FG representations in addition to GeoJSON as defined by RFC 7946.

A `profile` query parameter can be used by clients to request a particular GeoJSON / JSON-FG representation from Implementations supporting multiple options:

- `rfc7946`: GeoJSON as defined by RFC 7946
- `jsonfg`: GeoJSON as extended by JSON-FG
- `jsonfg-plus`: JSON-FG extensions in addition to GeoJSON compatibility (geometry specified in `geometry` limited to GeoJSON types, with coordinates either in `OGC:CRS84` or in a CRS explicitly requested by the client using `crs` query parameter)

Requirements class 15: Requirements Class GeoJSON	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-geojson`
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.15: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-geojson`
PREREQUISITES	GeoJSON OGC 21-045r1 (optional JSON-FG dependency) Requirements class 2: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval`
NORMATIVE STATEMENT	Requirement 29: `/req/data-geojson/content`

16.7.2. GeoJSON Zone Data

Requirement 29	
IDENTIFIER	`/req/data-geojson/content`
INCLUDED IN	Requirements class 15: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-geojson`
STATEMENT	For encoding zone data as GeoJSON (and/or JSON-FG)
A	Every 200 response of the server for zone data with the media type `application/geo+json` SHALL be a GeoJSON document representing the features, including their geometry and associated properties, within the zone for which data is requested.

Requirement 29	
B	Unless otherwise specified by a prior arrangement (for example, an output `crs` query parameter) or by negotiating a JSON-FG profile, the coordinate reference system SHALL be CRS84(h) in longitude and latitude (and optional height above the WGS84 ellipsoid).
C	Features whose geometry lie wholly outside of the zone geometry SHALL NOT be included in the response.
D	If returning a JSON-FG profile, the profile URI SHALL be included in the `links` section of the JSON-FG response.

Recommendation 16	
IDENTIFIER	`/rec/data-geojson/jsonfg-profile`
A	The Implementation SHOULD support a `profile` query parameter for zone data requests allowing to negotiate an OGC Feature & Geometry JSON representation, where a value of `rfc7946` corresponds to RFC 7936 GeoJSON, a `jsonfg` value corresponds to a response conforming to JSON-FG, and a `jsonfg-plus` value corresponds to a response conforming to JSON-FG fully backward compatible with GeoJSON (e.g., including a GeoJSON geometry in the `geometry` field if geometry is otherwise provided in the JSON-FG-specific `place` property which supports extended geometry types and alternate coordinate reference system without prior agreement).

Recommendation 17	
IDENTIFIER	`/rec/data-geojson/profile-links`
A	If returning a JSON-FG representation, the implementation SHOULD return a link to the default or selected profile as a `Link` HTTP response header, for example `Link: https://www.opengis.net/def/profile/ogc/0/jsonfg; rel="profile"`.

Permission 3	
IDENTIFIER	`/per/data-geojson/supported-profiles`
A	An Implementation MAY support only a subset of the defined GeoJSON profiles.
B	An Implementation MAY return any profile as the default when the `profile` query parameter is not specified in the request. As an example, this allows for static servers to return only a JSON-FG representation.

Recommendation 18

IDENTIFIER	`/rec/data-geojson/clipping`
A	For features partially inside of the zone geometry, the geometry included in the response SHOULD be clipped to the boundaries of the zone geometry.

Recommendation 19

IDENTIFIER	`/rec/data-geojson/generalization`
A	The geometry of the features in the response SHOULD be generalized to the scale associated with the hierarchy level implied from the `zone-depth` relative to the requested zone.

Recommendation 20

IDENTIFIER	`/rec/data-geojson/omission`
A	For datasets whose features have either a size (such as areal or linear feature) or other properties making some features irrelevant at lower hierarchy levels, such irrelevant features SHOULD be omitted from the response based on the hierarchy level implied from the `zone-depth` relative to the requested zone.

Recommendation 21

IDENTIFIER	`/rec/data-geojson/crs`
A	For an JSON-FG response for which a `crs` query parameter is not specified, the CRS of the response SHOULD be consistent with the CRS of the DGGRS or the underlying geographic CRS (e.g., CRS84).
B	For an JSON-FG response for which a `crs` query parameter is specified, the `geometry` property SHOULD be used to return coordinates if the geometry does not require an extended JSON-FG type, since this constitutes a prior arrangement regarding the CRS of GeoJSON coordinates.

Recommendation 22

IDENTIFIER	`/rec/data-geojson/geometry`
A	The implementation SHOULD support a `geometry` query parameter allowing a client to choose how to return feature geometry.

Recommendation 22	
B	The implementation SHOULD support a value of `geometry=zone-centroid` to request a representation where each feature is a zone intersecting the data, with the geometry of each feature being a Point geometry for the centroid of that zone.
C	The implementation SHOULD support a value of `geometry=zone-region` to request a representation where each feature is a zone intersecting the data, with the geometry of each feature being a (Multi)Polygon, (Multi)Polyhedron or (Multi)Prism.
D	The implementation SHOULD support a value of `geometry=vectorized` to request a representation where each feature corresponds to a feature of the data (sharing identical property values).
E	If a `geometry` value is not specified, the Implementation SHOULD return the representation closest to the native data.
F	If a requested geometry representation is not supported, the Implementation SHOULD return a 4xx HTTP error.

16.8. Requirements Class “netCDF zone data encoding”

16.8.1. Overview

The netCDF Zone Data encoding requirements class defines support for encoding a DGGS Zone data response according to the OGC netCDF standard and to any encoding extension such as the NetCDF classic and 64-bit offset format, or netCDF version 4 based on OGC HDF5. NetCDF (network Common Data Form) is a data model for array-oriented scientific data, a freely distributed collection of access libraries implementing support for that data model, and machine-independent formats.

An Implementation may support either a DGGS-optimized netCDF representation where a dimension is associated with sub-zones, with coordinates being global zone identifiers or local indices into the deterministic zone order, or a traditional representation interoperable with non-DGGS clients where spatiotemporal axes are individual dimensions.

A `profile` query parameter can be used by clients to request a particular netCDF representation from Implementations supporting multiple options:

- `netcdf3`: NetCDF classic and 64-bit offset format (not quantized to DGGH)
- `netcdf3-dggs`: NetCDF classic and 64-bit offset format where one axis corresponds to local sub-zone indices
- `netcdf3-dggs-zoneids`: NetCDF classic and 64-bit offset format where one axis corresponds to the global identifiers of sub-zones (textual or 64-bit integer)
- `netcdf4`: HFG5-based NetCDF 4 format (not quantized to DGGH)

- `netcdf4-dggs`: HDF5-based NetCDF 4 format where one axis corresponds to local sub-zone indices
- `netcdf4-dggs-zoneids`: HDF5-based NetCDF 4 format where one axis corresponds to the global identifiers of sub-zones (textual or 64-bit integer)

Requirements class 16: Requirements Class NetCDF	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-netcdf`
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.16: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-netcdf`
PREREQUISITES	OGC Network Common Data Form (NetCDF) Core Encoding Standard version 1.0 Requirements class 2: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval`
NORMATIVE STATEMENT	Requirement 30: `/req/data-netcdf/content`

16.8.2. NetCDF Zone Data

Requirement 30	
IDENTIFIER	`/req/data-netcdf/content`
INCLUDED IN	Requirements class 16: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-netcdf`
STATEMENT	For encoding zone data as netCDF
A	Every 200 response of the server for zone data with the media type `application/netcdf` SHALL be a netCDF file representing the data values for all selected fields of each included sub-zone.
B	The response SHALL be encoded using a netCDF encoding extension, such as the NetCDF classic and 64-bit offset format, or netCDF version 4 based on OGC HDF5.
C	For a DGGS-optimized profile (`netcdf3-dggs`, `netcdf3-dggs-zoneids`, `netcdf4-dggs`, `netcdf3-dggs-zoneids`), the profile URI, the global identifier of the parent zone and a URI or link to the DGGRS definition SHALL be included as metadata.
D	For a DGGS-optimized profile, coordinates along one dimension SHALL correspond to sub-zones, either as global zone identifiers (`netcdf3-dggs-zoneids`, `netcdf4-dggs-zoneids`) or local sub-zone indices following the deterministic DGGRS sub-zone order (`netcdf3-dggs`, `netcdf4-dggs`), while spatiotemporal dimensions of the DGGRS SHALL NOT be defined as separate dimensions.

Requirement 30	
E	For non-DGGS-optimized profiles, if the netCDF data incorporates georeferencing information, this information SHALL be consistent with the DGGRS Zone ID.
F	For non-DGGS-optimized profiles, if the zone geometry is not rectilinear, the closest bounding rectangle (or volume) SHALL be used for referencing the data.
G	For non-DGGS-optimized profiles, a distinct value for each sub-zone implied from the requested `zone-depth` SHALL correspond to at least one distinct cell value in the response.

Recommendation 23	
IDENTIFIER	`/rec/data-netcdf/null-values`
A	For non-DGGS-optimized profiles, null values SHOULD be used for cells lying outside the zone geometry.

Recommendation 24	
IDENTIFIER	`/rec/data-netcdf/crs`
A	For non-DGGS-optimized profiles, the spatial reference of the response, determined by the units of the coordinate variables and whether or not the `grid_mapping` attribute exists, as well as any applicable extension in use such as CF-netCDF, SHOULD be consistent with either the DGGRS or the underlying geographic CRS (e.g., CRS84).

Recommendation 25	
IDENTIFIER	`/rec/data-netcdf/profile`
A	The Implementation SHOULD support a `profile` query parameter for zone data requests allowing to negotiate a particular representation, including the selection of classic netCDF 3 and 64-bit offset format (`netcdf3`, `netcdf3-dggs` or `netcdf3-dggs-zoneids`) or netCDF 4 based on OGC HDF5 (`netcdf4`, `netcdf4-dggs` or `netcdf4-dggs-zoneids`), as well as the selection of a dimension representing global sub-zone identifiers (`netcdf3-dggs-zoneids` or `netcdf4-dggs-zoneids`) or local sub-zone indices based on the deterministic order as defined by the DGGRS (`netcdf3-dggs` or `netcdf4-dggs`), or individual spatiotemporal dimensions for interoperability with non-DGGS clients (`netcdf3` or `netcdf4`).

Recommendation 26	
IDENTIFIER	`/rec/data-netcdf/profile-links`

Recommendation 26	
A	The implementation SHOULD return a link to the default or selected profile as a `Link` HTTP response header, for example `Link: https://www.opengis.net/def/profile/ogc/0/netcdf4-dggs; rel="profile"`.

Permission 4	
IDENTIFIER	`/per/data-netcdf/supported-profiles`
A	An Implementation MAY support only a subset of the defined netCDF profiles.
B	An Implementation MAY return any profile as the default when the `profile` query parameter is not specified in the request. As an example, this allows for static servers to return only a DGGS-optimized netCDF 3 representation.

16.9. Requirements Class "Zarr zone data encoding"

16.9.1. Overview

The Zarr Zone Data encoding requirements class defines support for encoding a DGGS Zone data response according to the OGC Zarr Storage Specification 2.0 Community Standard, packed into a Zip archive.

An Implementation may support either a DGGS-optimized Zarr representation where a dimension is associated with sub-zones, with coordinates being global zone identifiers or local indices into the deterministic zone order, or a traditional representation interoperable with non-DGGS clients where spatiotemporal axes are individual dimensions.

A `profile` query parameter can be used by clients to request a particular Zarr representation from Implementations supporting multiple options:

- `zarr2`: Zipped Zarr 2.0 (not quantized to DGGH)
- `zarr2-dggs`: Zipped Zarr 2.0 where one axis corresponds to local sub-zone indices
- `zarr2-dggs-zoneids`: Zipped Zarr 2.0 where one axis corresponds to the global identifiers of sub-zones (textual or 64-bit integer)

Requirements class 17: Requirements Class Zarr	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-zarr`
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.17: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-zarr`
PREREQUISITES	OGC Zarr Storage Specification 2.0 Community Standard ISO/IEC 21320-1:2015 Information technology — Document Container File (Zip) Requirements class 2: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval`
NORMATIVE STATEMENT	Requirement 31: `/req/data-zarr/content`

16.9.2. Zarr Zone Data

Requirement 31	
IDENTIFIER	`/req/data-zarr/content`
INCLUDED IN	Requirements class 17: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-zarr`
STATEMENT	For encoding zone data as zipped Zarr 2
A	Every 200 response of the server for zone data with the media type `application/zarr+zip` SHALL be a Zip file containing the data values for all selected fields of each included sub-zone encoded following the Zarr Storage Specification Version 2 file.
B	For a DGGS-optimized profile (`zarr2-dggs`, `zarr2-dggs-zoneids`), the profile URI, the global identifier of the parent zone and a URI or link to the DGGRS definition SHALL be included as metadata.
C	For a DGGS-optimized profile, coordinates along one dimension SHALL correspond to sub-zones, either as global zone identifiers (`zarr2-dggs-zoneids`) or local sub-zone indices following the deterministic DGGRS sub-zone order (`zarr2-dggs`), while spatiotemporal dimensions of the DGGRS SHALL NOT be defined as separate dimensions.
D	For a DGGS-optimized profile, at every depth, each individual value SHALL correspond exactly to the data sampled representative of that sub-zone.
E	For non-DGGS-optimized profiles, if the Zarr data incorporates georeferencing information (such as defined by the candidate OGC GeoZarr Standard), this information SHALL be consistent with the DGGRS Zone ID.
F	For non-DGGS-optimized profiles, if the zone geometry is not rectilinear, the closest bounding rectangle (or volume) SHALL be used for referencing the data.

Requirement 31	
G	For non-DGGS-optimized profiles, a distinct value for each sub-zone implied from the requested `zone-depth` SHALL correspond to at least one distinct cell value in the response.
H	If multiple relative zone depths are being returned, each depth SHALL be encoded as a separate Zarr dataset within the data packet.
I	Every zone depth requested using the `zone-depth` query parameter SHALL be included in the response.

Recommendation 27	
IDENTIFIER	`/rec/data-zarr/no-extra-chunking`
A	Since the zone data packet are for a limited set of sub-zones within a single parent zone, the zone data response SHOULD be encoded as a single chunk.

Recommendation 28	
IDENTIFIER	`/rec/data-zarr/null-values`
A	For non-DGGS-optimized profiles, null values SHOULD be used for cells lying outside the zone geometry.

Recommendation 29	
IDENTIFIER	`/rec/data-zarr/crs`
A	For non-DGGS-optimized profiles, if the respone contains georferencing information (such as defined by the candidate OGC GeoZarr Standard) the spatial reference of the response SHOULD be consistent with either the DGGRS or the underlying geographic CRS (e.g., CRS84).

Recommendation 30	
IDENTIFIER	`/rec/data-zarr/profile`
A	The Implementation SHOULD support a `profile` query parameter for zone data requests allowing to negotiate a particular representation, including the selection of a dimension representing global sub-zone identifiers (`zarr-dggs-zoneids`) or local sub-zone indices based on the deterministic order as defined by the DGGRS (`zarr2-dggs`), or individual spatiotemporal dimensions for interoperability with non-DGGS clients (`zarr2`).

Recommendation 31	
IDENTIFIER	`/rec/data-zarr/profile-links`
A	The implementation SHOULD return a link to the default or selected profile as a `Link` HTTP response header, for example `Link: https://www.opengis.net/def/profile/ogc/0/zarr2-dggs; rel="profile"`.

Permission 5	
IDENTIFIER	`/per/data-zarr/supported-profiles`
A	An Implementation MAY support only a subset of the defined Zarr profiles.
B	An Implementation MAY return any profile as the default when the `profile` query parameter is not specified in the request. As an example, this allows for static servers to return only a DGGS-optimized Zarr 2 representation.

16.10. Requirements Class “CoverageJSON zone data encoding”

16.10.1. Overview

The CoverageJSON requirements class defines support for encoding a DGGS Zone data response according to the OGC CoverageJSON Community Standard.

An Implementation may support either a DGGS-optimized CoverageJSON representation where a dimension is associated with sub-zones, with coordinates being global zone identifiers or local indices into the deterministic zone order, or a traditional representation interoperable with non-DGGS clients where spatiotemporal axes are individual dimensions.

A `profile` query parameter can be used by clients to request a particular CoverageJSON representation from Implementations supporting multiple options:

- `covjson`: CoverageJSON (not quantized to DGGH)
- `covjson-dggs`: CoverageJSON where one axis corresponds to local sub-zone indices
- `covjson-dggs-zoneids`: CoverageJSON where one axis corresponds to the global identifiers of sub-zones (textual or 64-bit integer)

Requirements class 18: Requirements Class CoverageJSON Data	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-coveragejson`
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.18: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-coveragejson`
PREREQUISITES	OGC 21-069r2 Requirements class 2: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval`
NORMATIVE STATEMENT	Requirement 32: `/req/data-coveragejson/content`

16.10.2. CoverageJSON Zone Data

Requirement 32	
IDENTIFIER	`/req/data-coveragejson/content`
INCLUDED IN	Requirements class 18: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-coveragejson`
STATEMENT	For encoding zone data as CoverageJSON
A	Every 200 response of the server for zone data with the media type `application/prs.coverage+json` SHALL be a CoverageJSON file representing the data values for all selected fields of each included sub-zone.
B	For a DGGS-optimized profile (`covjson-dggs`, `covjson-dggs-zoneids`), the profile URI, the global identifier of the parent zone and a URI or link to the DGGRS definition SHALL be included as metadata.
C	For a DGGS-optimized profile, coordinates along one dimension SHALL correspond to sub-zones, either as global zone identifiers (`covjson-dggs-zoneids`) or local sub-zone indices following the deterministic DGGRS sub-zone order (`covjson-dggs`), while spatiotemporal dimensions of the DGGRS SHALL NOT be defined as separate dimensions.
D	For non-DGGS-optimized profiles, if the CoverageJSON encoding incorporates georeferencing information, this information SHALL be consistent with the DGGRS Zone ID.
E	For non-DGGS-optimized profiles, if the zone geometry is not rectilinear, the closest bounding rectangle (or volume) SHALL be used for referencing the data.
F	For non-DGGS-optimized profiles, a distinct value for each sub-zone implied from the requested `zone-depth` SHALL correspond to at least one distinct cell value in the response.

Recommendation 32	
IDENTIFIER	`/rec/data-coveragejson/null-values`
A	For non-DGGS-optimized profiles, null values SHOULD be used for cells lying outside the zone geometry.

Recommendation 33	
IDENTIFIER	`/rec/data-coveragejson/crs`
A	For non-DGGS-optimized profiles, the CRS of the response SHOULD be consistent with either the DGGRS or the underlying geographic CRS (e.g., CRS84).

Recommendation 34	
IDENTIFIER	`/rec/data-coveragejson/profile`
A	The Implementation SHOULD support a `profile` query parameter for zone data requests allowing to negotiate a particular representation, including the selection of a dimension representing global sub-zone identifiers (`covjson-dggs-zoneids`) or local sub-zone indices based on the deterministic order as defined by the DGGRS (`covjson-dggs`), or individual spatiotemporal dimensions for interoperability with non-DGGS clients (`covjson`).

Recommendation 35	
IDENTIFIER	`/rec/data-coveragejson/profile-links`
A	The implementation SHOULD return a link to the default or selected profile as a `Link` HTTP response header, for example `Link: https://www.opengis.net/def/profile/ogc/0/covjson-dggs; rel="profile"`.

Permission 6	
IDENTIFIER	`/per/data-coveragejson/supported-profiles`
A	An Implementation MAY support only a subset of the defined CoverageJSON profiles.
B	An Implementation MAY return any profile as the default when the `profile` query parameter is not specified in the request. As an example, this allows for static servers to return only a DGGS-optimized CoverageJSON representation.

16.11. Requirements Class “JPEG-XL Zone data encoding”

16.11.1. Overview

The JPEG XL Zone Data encoding requirements class defines support for encoding a DGGS Zone data response according to the JPEG XL ISO/IEC 18181 Standard.

JPEG XL supports a large number of bands, image dimensions, and floating point values.

Requirements class 19: Requirements Class JPEG XL Data	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-jpegxl`
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.19: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-jpegxl`
PREREQUISITES	ISO/IEC 18181-1 ISO/IEC 18181-2 Requirements class 2: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval`
NORMATIVE STATEMENT	Requirement 33: `/req/data-jpegxl/content`

16.11.2. JPEG XL Zone Data

Requirement 33	
IDENTIFIER	`/req/data-jpegxl/content`
INCLUDED IN	Requirements class 19: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-jpegxl`
STATEMENT	For encoding zone data as JPEG XL
A	Every 200 response of the server for zone data with the media type `image/jxl` SHALL be a JPEG XL image representing the data values for all selected fields within the zone for which data is requested.

Requirement 33	
B	If the zone geometry is not rectangular, the closest bounding rectangle SHALL be used for referencing the image.
C	A distinct value for each sub-zone implied from the requested `zone-depth` SHALL correspond to at least one distinct cell value in the response.
D	For Implementations supporting Data Custom Depths, each depth of the requested zone depth pyramid SHALL be a separate image (overview) in the response.

Recommendation 36	
IDENTIFIER	`/rec/data-jpegxl/null-values`
A	Null values SHOULD be used for cells lying outside the zone geometry.

Recommendation 37	
IDENTIFIER	`/rec/data-jpegxl/crs`
A	The CRS of the response SHOULD be consistent with either the DGGRS or the underlying geographic CRS (e.g., CRS84).

16.12. Requirements Class “PNG Zone data encoding”

16.12.1. Overview

The PNG Zone Data encoding requirements class defines support for encoding a DGGS Zone data response according to the W3C Portable Network Graphics (PNG) Specification (ISO/IEC 15948:2003).

Because PNG encoding is limited to an integer data values, this requirements class defines additional query parameters and response headers allowing a client to request a specific scale factor and offset to be used to quantize data values, and allowing the implementation to inform the client of the scale factor and offset used for that quantization.

Requirements class 20: Requirements Class PNG Data	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-png`
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.20: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-png`
PREREQUISITES	ISO/IEC 15948 Requirements class 2: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval`
NORMATIVE STATEMENT	Requirement 34: `/req/data-png/content`

16.12.2. PNG Zone Data

Requirement 34	
IDENTIFIER	`/req/data-png/content`
INCLUDED IN	Requirements class 20: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-png`
STATEMENT	For encoding zone data as PNG
A	Every 200 response of the server for zone data with the media type `image/png` SHALL be a PNG image representing the data values for all selected fields within the zone for which data is requested.
B	If the zone geometry is not rectangular, the closest bounding rectangle SHALL be used for referencing the image.
C	A distinct value for each sub-zone implied from the requested `zone-depth` SHALL correspond to at least one distinct cell value in the response.
D	For Implementations supporting Data Custom Depths, each depth of the requested zone depth pyramid SHALL be a separate image (overview) in the response.
E	A `Values-Scale:` response header with a numeric real value SHALL be returned indicating the factor by which values were multiplied before an offset was added to result in the encoded 8-bit or 16-bit PNG unsigned integer values.
F	A `Values-Offset:` response header with a numeric real value SHALL be returned indicating the offset which was added after multiplying values by the scale factor to result in the encoded 8-bit or 16-bit PNG unsigned integer values.

Recommendation 38	
IDENTIFIER	`/rec/data-png/null-values`
A	Null values SHOULD be used for cells lying outside the zone geometry.

Recommendation 39	
IDENTIFIER	`/rec/data-png/crs`
A	The CRS of the response SHOULD be consistent with either the DGGRS or the underlying geographic CRS (e.g., CRS84).

Recommendation 40	
IDENTIFIER	`/rec/data-png/scale-offset`
A	The Implementation SHOULD support a `values-scale` query parameter on Zone data resources when requesting a PNG output to indicate the value scale factor, as described in the `Values-Scale` response header part of the content requirement.
B	The Implementation SHOULD support a `values-offset` query parameter on Zone data resources when requesting a PNG output to indicate the value offset, as described in the `Values-Offset` response header part of the content requirement.

17

REQUIREMENTS CLASSES FOR ENCODINGS OF ZONE LIST

17 REQUIREMENTS CLASSES FOR ENCODINGS OF ZONE LIST

The OGC API — DGGS Standard requires supporting a JSON encoding for returning the `../dggs/{dggrsId}/zones` resources, as described in the Zone Query requirement class. In addition to this JSON encoding, DGGRS Zone Lists can be encoded in any suitable data format, as negotiated by erver and client. A number of requirements classes are defined for zone list encodings which are expected to be commonly supported in implementations of this Standard. These requirements classes include:

- HTML Zone List
- 64-bit Binary Zone List
- GeoJSON Zone List — including support for Features & Geometry JSON (JSON-FG) output as a recommendation
- GeoTIFF Zone List

17.1. Media Types (for zone list)

A table of the media types used in the zone list encoding requirements classes defined in this standard follows.

Table 4 — Media Types used for zone list encoding requirements classes

Encoding	Media type
JSON zone list	application/json
HTML zone list	text/html
Binary 64-bit zone list	application/x-binary
GeoJSON / JSON-FG zone list	application/geo+json
GeoTIFF zone list	image/tiff; application=geotiff

17.2. Requirements Class “HTML zone list encoding”

17.2.1. Overview

The HTML zone list encoding requirements class defines at a high level the ability to retrieve an HTML response for the DGGRS zone list resource intended primarily for users accessing the API from a Web browser.

The exact content of the HTML response is not prescribed, leaving the flexibility for implementations to choose a preferred approach.

HTML is the core language of the World Wide Web. An API that supports HTML will support browsing the spatial resources with a web browser and will also enable search engines to crawl and index those resources.

Requirements class 21: Requirements Class HTML Zone List	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-html`
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.21: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-html`
PREREQUISITES	https://html.spec.whatwg.org/: Requirements class 6: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query`
NORMATIVE STATEMENT	Requirement 35: `/req/zone-html/content`

17.2.2. HTML Zone List

Requirement 35	
IDENTIFIER	`/req/zone-html/content`
INCLUDED IN	Requirements class 21: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-html`
STATEMENT	For returning zone lists encoded as HTML:

Requirement 35	
A	Every 200 response of the server for zone query with the media type `text/html` SHALL be an HTML document listing the textual identifiers for all zones matching the query.

Recommendation 41	
IDENTIFIER	`/rec/zone-html/zone-information`
A	The implementation SHOULD include additional information for each zone ID, such as the area, bounding box and/or a preview of the data available for this zone.

Recommendation 42	
IDENTIFIER	`/rec/zone-html/zone-data`
A	If the Implementation also supports Zone Data Retrieval, the Implementation SHOULD include a link for each zone to download the associated data .

17.3. Requirements Class “Binary 64-bit integer zone list encoding”

17.3.1. Overview

The binary 64-bit integer zone list encoding requirements class defines the ability to retrieve a binary response for the DGGRS zone list resource for DGGRS whose zone identifiers can be expressed as a single 64-bit integer.

The response consists of a 64-bit integer count of zones, followed by that count of zones, also 64-bit integers. With compact zones and additional HTTP content encoding compression, this provides an optimal way to exchange DGGRS zone lists.

Requirements class 22: Requirements Class 64-bit Binary Zone List	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-uint64`
TARGET TYPE	Web API

Requirements class 22: Requirements Class 64-bit Binary Zone List	
CONFORMANCE CLASS	Conformance class A.22: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-uint64`
PREREQUISITE	Requirements class 6: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query`
NORMATIVE STATEMENT	Requirement 36: `/req/zone-uint64/content`

17.3.2. 64 bit Binary Zone List

Requirement 36	
IDENTIFIER	`/req/zone-uint64/content`
INCLUDED IN	Requirements class 22: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-uint64`
STATEMENT	For returning zone lists encoded as binary 64-bit unsigned integers:
A	Every 200 response of the server for zone query with the media type `application/x-binary` SHALL be a binary response consisting of a first 64-bit integer count defining the number of zones returned, followed by one 64-bit integer for each zone matching the query.
B	The 64-bit integer identifiers SHALL be the one defined by the DGGRS.
C	The endianness of the returned count and zones integers SHALL be little endian.
D	If the DGGRS does not define a 64-bit integer identifier, a 406 “Not Acceptable” response SHALL be returned.

17.4. Requirements Class “GeoJSON zone list encoding”

17.4.1. Overview

The GeoJSON zone list encoding requirements class defines support for encoding a DGGRS Zone list response according to the GeoJSON specification, and optionally according to the OGC Features & Geometry JSON (JSON-FG) Standard.

The response is a feature collection where each feature represents a zone in the list. The geometry of the zone should be polygons or multipolygons (e.g., in the case of zone geometry which must be split on the anti-meridian or pole), or in the case of a 3D DGGRS and an JSON-

FG response polyhedrons, multipolyhedrons, prisms or multiprisms. The zone identifier is represented in the `id` field of each feature.

While not intended to efficiently exchange zone list, as the response carries the geometry of the zones which could easily be computed client-side which takes considerable bandwidth and is not necessary for a DGGS client working natively with the same DGGRS, this requirements class provides an easy way to readily visualize the response in a variety of tools. This requirements class is therefore intended for convenience, demonstration and educational purposes.

Requirements class 23: Requirements Class GeoJSON Zone List	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-geojson`
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.23: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-geojson`
PREREQUISITES	GeoJSON OGC 21-045r1 (optional JSON-FG dependency) Requirements class 6: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query`
NORMATIVE STATEMENT	Requirement 37: `/req/zone-geojson/content`

17.4.2. GeoJSON Zone List

Requirement 37	
IDENTIFIER	`/req/zone-geojson/content`
INCLUDED IN	Requirements class 23: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-geojson`
STATEMENT	For returning zone lists encoded as GeoJSON:
A	Every 200 response of the server for zone query with the media type `application/geo+json` SHALL be a GeoJSON document consisting of a FeatureCollection, where every Feature represents a single zone.
B	Unless otherwise specified by a prior arrangement (for example, an output `crs` query parameter) or by negotiating a JSON-FG profile, the coordinate reference system SHALL be CRS84(h) in longitude and latitude (and optional height above the WGS84 ellipsoid).
C	Every feature SHALL have a `zoneID` property corresponding to the textual identifier of the zone.

Requirement 37

D	The geometry of each feature SHALL be the geometry of the zone.
E	If returning a JSON-FG profile, the profile URI SHALL be included in the `links` section of the JSON-FG response.

Recommendation 43

IDENTIFIER	`/rec/zone-geojson/jsonfg-profile`
A	The Implementation SHOULD support a `profile` query parameter for zone query requests allowing to negotiate an OGC Feature & Geometry JSON representation, where a value of `rfc7946` corresponds to RFC 7936 GeoJSON, a `jsonfg` value corresponds to a response conforming to JSON-FG, and a `jsonfg-plus` value corresponds to a response conforming to JSON-FG fully backward compatible with GeoJSON (e.g., including a GeoJSON geometry in the `geometry` field if geometry is otherwise provided in the JSON-FG-specific `place` property which supports extended geometry types and alternate coordinate reference system without prior agreement).

Recommendation 44

IDENTIFIER	`/rec/zone-geojson/profile-links`
A	If returning a JSON-FG representation, the implementation SHOULD also return a link to the default or selected profile as a `Link` HTTP response header, for example `Link: https://www.opengis.net/def/profile/ogc/0/jsonfg; rel="profile"`.

Permission 7

IDENTIFIER	`/per/zone-geojson/supported-rofiles`
A	An Implementation MAY support only a subset of the defined GeoJSON profiles.
B	An Implementation MAY return any profile as the default when the `profile` query parameter is not specified in the request.

Recommendation 45

IDENTIFIER	`/rec/zone-geojson/id`
A	The Feature `id` SHOULD be a sequential identifier starting at 1, have the same text value as the `zoneID` property, or correspond to the 64-bit integer identifier for the zone.

Recommendation 46

IDENTIFIER	`/rec/zone-geojson/crs`
A	For an JSON-FG response for which a `crs` query parameter is not specified, the CRS of the response SHOULD be consistent with the CRS of the DGGRS or the underlying geographic CRS (e.g., CRS84).
B	For an JSON-FG response for which a `crs` query parameter is specified, the `geometry` property SHOULD be used to return coordinates if the geometry does not require an extended JSON-FG type, since this constitutes a prior arrangement regarding the CRS of GeoJSON coordinates.

Recommendation 47

IDENTIFIER	`/rec/zone-geojson/mid-points`
A	For DGGRSs not in CRS84, the geometry SHOULD include intermediate points between the vertices of the zone geometry so as to accurately represent the shape of the zones.

Recommendation 48

IDENTIFIER	`/rec/zone-geojson/geometry`
A	The implementation SHOULD support a `geometry` query parameter allowing a client to choose how to return the geometry of each zone.
B	The implementation SHOULD support a value of `geometry=zone-centroid` to request a representation where the geometry of each a zone is a Point geometry for the centroid of that zone.
C	The implementation SHOULD support a value of `geometry=zone-region` to request a representation where the geometry of each zone is a (Multi)Polygon, (Multi)Polyhedron, or (Multi)Prism.
D	The implementation SHOULD support a value of `geometry=none` to request a representation with a null geometry.
E	If a requested geometry representation is not supported, the Implementation SHOULD return a 4xx HTTP error.

17.5. Requirements Class “GeoTIFF zone list encoding”

17.5.1. Overview

The GeoTIFF zone list encoding requirements class defines support for encoding a DGGRS Zone list response according to the OGC GeoTIFF standard, intended primarily for DGGRSs with rectangular zones.

The response is a 2D gridded coverage where each cell is a zone in the list, in the coordinate reference system of that DGGRS.

For DGGRSs with non-rectangular zones, the resolution would need to be higher than a single pixel per the most detailed zone to be returned in order to be able to recognize the zone geometry, and would therefore be very sub-optimal. While for rectangular zones it would be easy to identify zones with a one-to-one correspondence, recognizing non-rectangular zones would be significantly more difficult.

While not intended to efficiently exchange zone list, this requirements class provides an easy way to readily visualize the response in a variety of tools. This requirements class is therefore intended for convenience, demonstration and educational purposes.

Requirements class 24: Requirements Class GeoTIFF Zone List	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-geotiff`
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.24: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-geotiff`
PREREQUISITES	TIFF V6.0 GeoTIFF Requirements class 6: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query`
NORMATIVE STATEMENT	Requirement 38: `/req/zone-geotiff/content`

17.5.2. GeoTIFF Zone List

Requirement 38	
IDENTIFIER	`/req/zone-geotiff/content`
INCLUDED IN	Requirements class 24: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-geotiff`
STATEMENT	For returning zone lists encoded as GeoTIFF:
A	Every 200 response of the server for zone query with the media type `image/tiff` SHALL be a Geo TIFF document representing the zones matching the query in a geo-referenced image where each zone corresponds to at least one pixel.
B	The GeoTIFF SHALL be encoded as `Pixel-Is-Area`.

Recommendation 49	
IDENTIFIER	`/rec/zone-geotiff/crs`
A	The CRS of the response SHOULD be consistent with either the DGGRS or the underlying geographic CRS (e.g., CRS84).

Recommendation 50	
IDENTIFIER	`/rec/zone-geotiff/values`
A	The value of the pixel SHOULD be a 64-bit integer identifier representing the zone, if the DGGRS provides for such an identifier.

Recommendation 51	
IDENTIFIER	`/rec/zone-geotiff/non-rectilinear`
A	For DGGRS where the zones do not correspond to a rectilinear structure, the resolution of the response image SHOULD be high enough so that the shape of the zone geometry is recognizable.

18

REQUIREMENTS CLASS “API DEFINITION OPERATION IDS”

18 REQUIREMENTS CLASS “API DEFINITION OPERATION IDS”

The OGC API — DGGS Standard does not mandate any particular API definition language. However, if the API is described using a definition language supporting operation identifiers, such as OpenAPI 3.0, being able to associate the functionality described in these requirements with an operation defined in the language is useful. This requirements class defines how this is achieved.

18.1. Overview

Requirements class 25: Requirements Class API Definition Operation IDs	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/operation-ids`
TARGET TYPE	Web API
CONFORMANCE CLASS	Conformance class A.25: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/operation-ids`
PREREQUISITE	https://www.opengis.net/spec/ogcapi-common-1/1.0/req/landing-page
NORMATIVE STATEMENT	Requirement 39: `/req/operation-ids/operation-ids`

18.2. Operation IDs

Requirement 39	
IDENTIFIER	`/req/operation-ids/operation-ids`
INCLUDED IN	Requirements class 25: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/operation-ids`
STATEMENT	For specifying the operation identifiers associated with the capabilities defined in OGC API — DGGS

Requirement 39	
A	The API definition SHALL identify the supported operations defined in this Standard using the identifier suffixes defined in table Table 5.

Table 5 — API operation identifier suffixes

ORIGIN	RESOURCE	OPERATION ID SUFFIXES
With the origins described in this document		
DataSet[4]	DGGRS List[1]	`.dataset.getDGGRSList`
DataSet[4]	DGGRS Description[1]	`.dataset.getDGGRS`
DataSet[4]	DGGRS Zones[2]	`.dataset.getDGGRSZones`
DataSet[4]	DGGRS Zone Information[1]	`.dataset.getDGGRSZoneInfo`
DataSet[4]	DGGRS Zone Data[3]	`.dataset.getDGGRSZoneData`
Collection[5]	DGGRS List[1]	`.collection.getDGGRSList`
Collection[5]	DGGRS Description[1]	`.collection.getDGGRS`
Collection[5]	DGGRS Zones[2]	`.collection.getDGGRSZones`
Collection[5]	DGGRS Zone Information[1]	`.collection.getDGGRSZoneInfo`
Collection[5]	DGGRS Zone Data[3]	`.collection.getDGGRSZoneData`
With other potential origins[6]		
other	DGGRS List[1]	`#.getDGGRSList`
other	DGGRS Description[1]	`#.getDGGRS`
other	DGGRS Zones[2]	`#.getDGGRSZones`
other	DGGRS Zone Information[1]	`#.getDGGRSZoneInfo`
other	DGGRS Zone Data[3]	`#.getDGGRSZoneData`
[1] The *DGGRS List* (…/`dggs`), *DGGRS Description* (…/`dggs/{dggrsId}`) and *DGGRS Zone Information* (…/`dggs/{dggrsId}/zones/{zoneId}`) resources are defined in requirements class “Core”.		

ORIGIN	RESOURCE	OPERATION ID SUFFIXES
[2] The *DGGRS Zones* resource (.../`dggs/{dggrsId}/zones`) is defined in the “Zone Query” requirements class.		
[3] The *DGGRS Zone Data* resource (.../`dggs/{dggrsId}/zones/{zoneId}/data`) is defined in the “Data Retrieval” requirements class.		
[4] The *DataSet* origin is defined in requirements class “Root DGGS” and depends on *OGC API — Common — Part 1: Core*.		
[5] The *Collection* origin is defined in requirements class “Collection DGGS” and depends on the *Collections* requirements class defined in *OGC API — Common — Part 2: Geospatial data*.		
[6] ‘#’ represents an optional *origin* that could be defined in another relevant standard.		

A

ANNEX A (NORMATIVE) CONFORMANCE CLASS ABSTRACT TEST SUITE

ANNEX A (NORMATIVE) CONFORMANCE CLASS ABSTRACT TEST SUITE

A.1. Conformance Class “Core”

Conformance class A.1	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/core`
REQUIREMENTS CLASS	Requirements class 1: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/core`
TARGET TYPE	Web API
CONFORMANCE TESTS	Abstract test A.1: `/conf/core/dggrs-list` Abstract test A.2: `/conf/core/dggrs-description` Abstract test A.3: `/conf/core/zone-info`

A.1.1. Abstract Test for Requirement Listing available DGGRS

Abstract test A.1	
IDENTIFIER	`/conf/core/dggrs-list`
REQUIREMENT	Requirement 1: `/req/core/dggrs-list`
TEST PURPOSE	Verify that the Implementation supports listing the available DGGRSs
TEST METHOD	**Given:** a geospatial data resource conforming to the DGGS API Standard, with an API path including …/`dggs` or discovered following a link with relation type `[ogc-rel:dggrs-list]`

Abstract test A.1	
	When: performing a GET operation on the …/`dggs` resource with an `application/json` media type specified in the `Accept:` header (e.g., `Accept: application/json`) **Then:** - assert that the Implementation supports an HTTP GET operation at a resource path ending with …/`dggs`. - assert that the Implementation supports a JSON representation of this …/`dggs` resource. - assert that the …/`dggs` resource includes a `dggrs` array property listing all Discrete Global Grid Reference Systems (DGGRSs) supported by the Implementation. - assert that each element of the `dggrs` array includes a summary description (a subset of the information available in the individual resources describing each DGGRS as tested for Abstract test A.2: `/conf/core/dggrs-description`), including at minimum the `id`, `title`, `uri` (if applicable), and `links`. - assert that the `links` property within each DGGRS element includes at minimum a link to the discrete global grid reference system using the `self` link relation type, as well as a link to the discrete global grid reference system definition using the `[ogc-rel:dggrs-definition]` relation type. - assert that the link relation type to use for linking from a particular resource of origin (such as a collection at `/collections/{collectionId}`, or landing page) to the list of available DGGRSs for that resource (for example at `/collections/{collectionId}/dggs` or `/dggs`) is `[ogc-rel:dggrs-list]`.

A.1.2. Abstract Test for Requirement Discrete global grid reference system description

Abstract test A.2	
IDENTIFIER	`/conf/core/dggrs-description`
REQUIREMENT	Requirement 2: `/req/core/dggrs-description`
TEST PURPOSE	Verify that the Implementation supports retrieving information for a particular DGGRS
TEST METHOD	**Given:** a DGGRS resource identified from listing available DGGRS or discovered by following a link with relation type `[ogc-rel:dggrs]` **When:** performing a GET operation on the …/`dggs/{dggrsId}` resource with an `application/json` media type specified in the `Accept:` header (e.g., `Accept: application/json`) **Then:** - assert that the Implementation supports an HTTP GET operation at a resource path ending with …/`dggs/{dggrsId}`. - assert that the Implementation supports a JSON representation of this …/`dggs/{dggrsId}` resource. - assert that the …/`dggs/{dggrsId}` resource includes an `id` property consistent with the `{dggrsId}` resource path parameter.

Abstract test A.2	
	- assert that the …/`dggs/{dggrsId}` resource includes a link to the resource itself using the `self` link relation type. - assert that the …/`dggs/{dggrsId}` resource includes a link to a definition of the discrete global grid reference system using the link relation type `[ogc-rel:dggrs-definition]` (this definition should be based on the draft schema in Annex B — Discrete Global Grid Reference System Definitions, although in this first version of the OGC API — DGGS Standard the schema is informative). - assert that the …/`dggs/{dggrsId}` resource includes a templated link in the `linkTemplates` array to request information for a particular zone using the link relation type `[ogc-rel:dggrs-zone-info]` and the template variable `{zoneId}`. - assert that if the discrete global grid reference system (the combination of the discrete global grid and indexing system) is registered with an authority, the resource includes a `uri` property corresponding to that registered discrete global grid reference system. - assert that if the discrete global grid reference system is based on a particular coordinate reference system, the resource specifies that CRS in a `crs` property, preferably as a URI (if one is available). - assert that the Implementation includes a short `title` property identifying the discrete global grid reference system intended for display to a human. - assert that the Implementation includes a `description` property providing a summary description of the discrete global grid reference system.

A.1.3. Abstract Test for Requirement Retrieving zone information

Abstract test A.3	
IDENTIFIER	`/conf/core/zone-info`
REQUIREMENT	Requirement 3: `/req/core/zone-info`
TEST PURPOSE	Verify that the Implementation supports retrieving information for a particular DGGRS zone
TEST METHOD	**Given:** a DGGRS zone information resource identified from a `[ogc-rel:dggrs-zone-info]` link template within a DGGRS description resource, replacing the `{zoneId}` variable with a known valid identifier for the DGGRS, not beyond the `maxRefinementLevel` if one is specified **When:** performing a GET operation on the …/`dggs/{dggrsId}/zones/{zoneID}` resource with an `application/json` media type specified in the `Accept:` header (e.g., `Accept: application/json`) **Then:** - assert that the Implementation supports an HTTP GET operation at a resource path ending with …/`dggs/{dggrsId}/zones/{zoneId}` providing information for valid individual zones of the discrete global grid reference system. - assert that the zone information resource supports a JSON representation. - assert that the zone information resource includes an `id` property corresponding to the `{zoneId}` resource path parameter.

Abstract test A.3	
	- assert that the zone information resource includes a link back to the corresponding DGGRS resource (.../`dggs/{dggrsId}`) using the `[ogc-rel:dggrs]` link relation type.

A.2. Conformance Class “Data Retrieval”

Conformance class A.2	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-retrieval`
REQUIREMENTS CLASS	Requirements class 2: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-retrieval`
TARGET TYPE	Web API
CONFORMANCE TEST	Abstract test A.4: `/conf/data-retrieval/zone-data`

A.2.1. Abstract Test for Requirement Retrieving data from a zone

Abstract test A.4	
IDENTIFIER	`/conf/data-retrieval/zone-data`
REQUIREMENT	Requirement 4: `/req/data-retrieval/zone-data`
TEST PURPOSE	Verify that the Implementation supports retrieving data for a particular DGGRS zone
TEST METHOD	**Given:** a DGGRS zone data resource identified from a `[ogc-rel:dggrs-zone-data]` link template within a DGGRS description resource, replacing the `{zoneId}` variable with a known valid identifier for the DGGRS, not beyond the `maxRefinementLevel` if one is specified **When:** performing a GET operation on the .../`dggs/{dggrsId}/zones/{zoneID}/data` resource with supported media type specified in the `Accept:` header (e.g., `Accept: application/json` for DGGS-JSON) **Then:** - assert that the Implementation supports an HTTP GET operation at a resource path ending with .../`dggs/{dggrsId}/zones/{zoneId}/data`. - assert that the Implementation includes a templated link to this resource path in the “Core” .../`dggs/{dggrsId}` resource link templates, and regular link in the .../`dggs/{dggrsId}/zones/`

Abstract test A.4	
	`{zoneId}` resource links using the link relation type `[ogc-rel:dggrs-zone-data]` for all zones for which data is available. - assert that the response of the HTTP GET operation has a status code of 200. - assert that the content of the response is a data packet corresponding precisely to the area covered by the DGGS zone. - assert that the selection of an encoding for the response is consistent with HTTP content negotiation. - assert that the …`/dggs/{dggrsId}` resource includes a `defaultDepth` property indicating the Implementation's default depth for when the `zone-depth` query parameter is omitted. This default value could be any valid value and/or form as defined in the `/req/data-custom-depths/zone-depth-parameter` requirement (single depth, range of depths, or list of depths, relative to the `{zoneId}` hierarchy level). - assert that unless a `zone-depth` query parameter is specified, the response returns a data packet consistent with this `defaultDepth` property, in accordance with the capabilities of the negotiated data packet encoding.

A.3. Conformance Class "Data Subsetting"

Conformance class A.3	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-subsetting`
REQUIREMENTS CLASS	Requirements class 3: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-subsetting`
TARGET TYPE	Web API
CONFORMANCE TESTS	Abstract test A.5: `/conf/data-subsetting/subset` Abstract test A.6: `/conf/data-subsetting/datetime` Abstract test A.7: `/conf/data-subsetting/properties` Abstract test A.8: `/conf/data-subsetting/exclude-properties`

A.3.1. Abstract Test for Requirement `subset` query parameter

Abstract test A.5	
IDENTIFIER	`/conf/data-subsetting/subset`
REQUIREMENT	Requirement 5: `/req/data-subsetting/subset`

Abstract test A.5	
TEST PURPOSE	Verify that the Implementation supports specifying a multi-dimensional subset for the zone data being retrieved
TEST METHOD	**Given:** a DGGRS zone data resource identified from a `[ogc-rel:dggrs-zone-data]` link template within a DGGRS description resource, replacing the `{zoneId}` variable with a known valid identifier for the DGGRS which successfully passed the `/conf/data-retrieval` abstract test **When:** performing a GET operation on the …`/dggs/{dggrsId}/zones/{zoneID}/data` resource with a supported media type and a `subset` query parameter identifying an additional dimension not part of the DGGRS supported by the collection **Then:** - assert that the Implementation supports a `subset` query parameter for the zone data retrieval operation (resource path ending with …`/dggs/{dggrsId}/zones/{zoneId}/data`) conforming to the Augmented Backus Naur Form (ABNF) fragment in Requirement 5: `/req/data-subsetting/subset` - assert that the Implementation supports as an axis names`time` for a temporal dataset, unless this temporal axis is an axis of the discrete global grid reference system. - assert that if a third vertical spatial dimension is supported (if the resource's spatial extent bounding box is three dimensional) and that dimension is not part of the discrete global grid system definition, the Implementation also supports a `h` dimension (elevation above the ellipsoid in EPSG:4979 or CRS84h) for geographic CRS and `z` for projected CRS, which are to be interpreted as the vertical axis in the CRS definition. - assert that the Implementation supports as axis names any additional dimension (beyond spatial and temporal) as described in the `extent` property of the collection or dataset description. - assert that the Implementation returns a 400 error status code if an axis name does not correspond to one of the axes of the Coordinate Reference System (CRS) of the data or an axis defined in the relevant `extent` property. - assert that if a `subset` query parameter including any of the dimensions corresponding to the axes of the discrete global grid reference system is used, the server returns a 400 client error. - assert that the Implementation interprets multiple `subset` query parameters, as if all dimension subsetting values were provided in a single `subset` query parameter (comma separated).

A.3.2. Abstract Test for Requirement `datetime` query parameter

Abstract test A.6	
IDENTIFIER	`/conf/data-subsetting/datetime`
REQUIREMENT	Requirement 6: `/req/data-subsetting/datetime`
TEST PURPOSE	Verify that the Implementation supports specifying a time instant or interval for which to retrieve data from a zone for a non-temporal DGGS
TEST METHOD	**Given:** a DGGRS zone data resource identified from a `[ogc-rel:dggrs-zone-data]` link template within a DGGRS description resource, replacing the `{zoneId}` variable with a known

Abstract test A.6	
	valid identifier for the DGGRS which successfully passed the `/conf/data-retrieval` abstract test **When:** performing a GET operation on the …/`dggs/{dggrsId}/zones/{zoneID}/data` resource with a supported media type and a `datetime` query parameter identifying a temporal dimension not part of the DGGRS supported by the collection **Then:** - assert that the Implementation supports a `datetime` query parameter expressed corresponding to either a date-time instant or a time interval, conforming to the ABNF in Requirement 6: `/req/data-subsetting/datetime` - assert that the implementation supports an `instant` defined as specified by RFC 3339, 5.6, with the exception that the server is only required to support the `Z` UTC time notation, and not required to support local time offsets. - assert that only the portions of the data within the specified interval is part of the zone data response, performing a trim operation for an interval or a slicing operation for an instant (in the case of a gridded coverage), or a filtering operation for feature data. - assert that time intervals unbounded at the start or at the end is supported using a double-dot (`..`) or an empty string for the start/end. - assert that if a `datetime` query parameter is specified requesting zone data where no temporal dimension applies, the Implementation either ignores the query parameter or returns a 4xx client error.

A.3.3. Abstract Test for Requirement `properties` query parameter

Abstract test A.7	
IDENTIFIER	`/conf/data-subsetting/properties`
REQUIREMENT	Requirement 7: `/req/data-subsetting/properties`
TEST PURPOSE	Verify that the Implementation supports specifying fields to include when retrieving zone data
TEST METHOD	**Given:** a DGGRS zone data resource identified from a `[ogc-rel:dggrs-zone-data]` link template within a DGGRS description resource, replacing the `{zoneId}` variable with a known valid identifier for the DGGRS which successfully passed the `/conf/data-retrieval` abstract test **When:** performing a GET operation on the …/`dggs/{dggrsId}/zones/{zoneID}/data` resource with a supported media type and a `properties` query parameter identifying one or more field defined in the collection's logical schema **Then:** - assert that the zone data retrieval operation supports a query parameter `properties` where the value is a comma-separated list of fields to be returned. - assert that the Implementation supports selecting a field using the identifier corresponding to the top-level property keys of the logical schema of the resource associated with the DGGRS zone data request, and return 400 status code for an unrecognized selected field.

Abstract test A.7	
	- assert that only the selected fields is returned from the zone data request. - assert that if the zone data encodings response can self-describe its list of fields (as with the `schema` property of the DGGS-JSON encoding), the field description corresponds to the requested list of fields. - assert that if the negotiated format of the response has a concept of field order, then the fields are in the same order as the requested list of selected fields.

A.3.4. Abstract Test for Requirement `exclude-properties` query parameter

Abstract test A.8	
IDENTIFIER	`/conf/data-subsetting/exclude-properties`
REQUIREMENT	Requirement 8: `/req/data-subsetting/exclude-properties`
TEST PURPOSE	Verify that the Implementation supports including all but specific fields when retrieving zone data
TEST METHOD	**Given:** a DGGRS zone data resource identified from a `[ogc-rel:dggrs-zone-data]` link template within a DGGRS description resource, replacing the `{zoneId}` variable with a known valid identifier for the DGGRS which successfully passed the `/conf/data-retrieval` abstract test **When:** performing a GET operation on the …`/dggs/{dggrsId}/zones/{zoneID}/data` resource with a supported media type and a `exclude-properties` query parameter identifying one or more field defined in the collection's logical schema **Then:** - assert that the zone data retrieval operation supports a query parameter `exclude-properties` where the value is a comma-separated list of fields not to be returned. - assert that the Implementation supports selecting fields not to return using the identifier corresponding to the top-level property keys of the logical schema of the resource associated with the DGGRS zone data request, and returns 400 status code for an unrecognized selected field. - assert that all but the selected fields are returned from the zone data request. - assert that the Implementation returns a 4xx error when using the both the `properties` and `exclude-properties` query parameters in the same request.

A.4. Conformance Class “Data Custom Depths”

Conformance class A.4	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-custom-depths`
REQUIREMENTS CLASS	Requirements class 4: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-custom-depths`
TARGET TYPE	Web API
CONFORMANCE TEST	Abstract test A.9: `/conf/data-custom-depths/zone-depth`

A.4.1. Abstract Test for Requirement `zone-depth` query parameter

Abstract test A.9	
IDENTIFIER	`/conf/data-custom-depths/zone-depth`
REQUIREMENT	Requirement 9: `/req/data-custom-depths/zone-depth`
TEST PURPOSE	Verify that the Implementation supports a query parameter to specify the DGGS refinement levels beyond the specified DGGS zone's refinement level to include in the response, when retrieving data for that zone
TEST METHOD	**Given:** a DGGRS zone data resource identified from a `[ogc-rel:dggrs-zone-data]` link template within a DGGRS description resource, replacing the `{zoneId}` variable with a known valid identifier for the DGGRS which successfully passed the `/conf/data-retrieval` abstract test **When:** performing a GET operation on the …`/dggs/{dggrsId}/zones/{zoneID}/data` resource with a supported media type and a `zone-depth` query parameter identifying the relative depth at which to retrieve the data **Then:** - assert that the Implementation supports a `zone-depth` query parameter for the HTTP GET operation on a resource path ending with …`/dggs/{dggrsId}/zones/{zoneId}/data`. - assert that the Implementation accepts the different types of values for the `zone-depth` query parameter defined in Requirement 9: `/req/data-custom-depths/zone-depth` (single positive integer value, a range of positive integer values in the form "{low}-{high}", a comma separated list of at least two (2) positive integer values), keeping in mind that some or all of these forms of the `zone-depth` query parameter may not be supported with particular data packet encodings. - assert that for each zone depth to be included in the response, the interpretation of a selected depth is: 0 corresponding to a single set of field value(s) for the requested zone, 1 corresponding to all zones of the next deeper hierarchy level associated with the requested zone by the indexing scheme, *n* corresponding to all zones for the *n*'th deeper level in the hierarchy level associated with the requested zone by the indexing scheme.

Abstract test A.9	
	- assert that the association of zones of deeper hierarchy levels with the requested zone is based on the DGGRS, which takes into consideration both the grid definition as well as the indexing system in use for the DGGS resource. - assert that if a `zone-depth` is specified, the operation returns the data at the refinement level(s) specified.

A.5. Conformance Class "Filtering Zone Data with CQL2"

Conformance class A.5	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-cql2-filter`
REQUIREMENTS CLASS	Requirements class 5: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-cql2-filter`
TARGET TYPE	Web API
CONFORMANCE TEST	Abstract test A.10: `/conf/data-cql2-filter/filter`

A.5.1. Abstract Test for Requirement `filter` query parameter (for zone data)

Abstract test A.10	
IDENTIFIER	`/conf/data-cql2-filter/filter`
REQUIREMENT	Requirement 10: `/req/data-cql2-filter/filter`
TEST PURPOSE	Verify that the Implementation supports specifying a CQL2 expression filtering the data returned from a zone data request
TEST METHOD	**Given:** a DGGRS zone data resource identified from a `[ogc-rel:dggrs-zone-data]` link template within a DGGRS description resource, replacing the `{zoneId}` variable with a known valid identifier for the DGGRS which successfully passed the `/conf/data-retrieval` abstract test **When:** performing a GET operation on the …`/dggs/{dggrsId}/zones/{zoneID}/data` resource with a supported media type and a `filter` query parameter identifying a CQL2 expression to filter the data being retrieved **Then:**

Abstract test A.10	
	- assert that the Implementation supports a `filter` query parameter specified using the CQL2-Text encoding of the OGC Common Query Language for the zone data retrieval operation (resource path ending with .../`dggs/{dggrsId}/zones/data`). - assert that for a rasterized representation, the data returned has null or NODATA values for sub-zones where the CQL2 expression evaluates to false when considering the geometry and the data of that sub-zone. - assert that or a vector representation, the data returned only includes features where the CQL2 expression evaluates to true when considering the geometry and the data of that sub-zone. - assert that the CQL2 expression evaluator supports the queryables declared in the JSON Schema resource linked to from the origin of the DGGRS resources using the `[ogc-rel:queryables]` link relation type.

A.6. Conformance Class “Zone Query”

Conformance class A.6	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-query`
REQUIREMENTS CLASS	Requirements class 6: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query`
TARGET TYPE	Web API
CONFORMANCE TESTS	Abstract test A.11: `/conf/zone-query/zones-list` Abstract test A.12: `/conf/zone-query/json-response` Abstract test A.13: `/conf/zone-query/zone-level` Abstract test A.14: `/conf/zone-query/compact-zones` Abstract test A.15: `/conf/zone-query/parent-zone` Abstract test A.16: `/conf/zone-query/bbox` Abstract test A.17: `/conf/zone-query/bbox-crs` Abstract test A.18: `/conf/zone-query/subset` Abstract test A.19: `/conf/zone-query/subset-crs` Abstract test A.20: `/conf/zone-query/datetime`

A.6.1. Abstract Test for Requirement Listing zones

Abstract test A.11	
IDENTIFIER	`/conf/zone-query/zones-list`

Abstract test A.11	
REQUIREMENT	Requirement 11: `/req/zone-query/zones-list`
TEST PURPOSE	Verify that the Implementation supports querying DGGRS zones
TEST METHOD	**Given:** a DGGRS zones query resource identified from a `[ogc-rel:dggrs-zone-query]` link **When:** performing a GET operation on the …`/dggs/{dggrsId}/zones` resource with supported media type specified in the `Accept:` header (e.g., `Accept: application/json` for a JSON zone list) **Then:** - assert that the Implementation supports an HTTP GET operation at a resource path ending with …`/dggs/{dggrsId}/zones`. - assert that the Implementation includes a link to this resource path in the “Core” …`/dggs/{dggrsId}` resource links using the link relation type https://www.opengis.net/def/rel/ogc/1.0/dggrs-zone-query. - assert that the response of the HTTP GET operation has a status code of 200. - assert that the content of the response is a list of zones fully covering where data is available (in the case where the resource is associated with a particular dataset), and matching any additional query parameters specified by the client (e.g., a filtering query parameter), without any redundancy. - assert that unless the zones are a compact list of zones (see `compact-zones` query parameter), the zones returned all are of the same DGGRS hierarchy level. - assert that the selection of an encoding for the returned list of zones is consistent with HTTP content negotiation. - assert that the Implementation supports at minimum a JSON encoding (media type `application/json`).

A.6.2. Abstract Test for Requirement JSON zone list encoding

Abstract test A.12	
IDENTIFIER	`/conf/zone-query/json-response`
REQUIREMENT	Requirement 12: `/req/zone-query/json-response`
TEST PURPOSE	Verify that the Implementation supports returning a list of DGGS zones encoded as JSON
TEST METHOD	**Given:** a DGGRS zones query resource identified from a `[ogc-rel:dggrs-zone-query]` link **When:** performing a GET operation on the …`/dggs/{dggrsId}/zones` resource with an `application/json` media type specified in the `Accept:` header **Then:** - assert that every 200 response of the server for zone query with the media type `application/json` is a JSON document listing the textual identifiers for all zones matching the query.

Abstract test A.12	
	- assert that the schema for the JSON document follows the JSON Schema for DGGS Zone Query described in Requirement 12: `/req/zone-query/json-response`, where the zone identifiers are strings within a `zones` array property within a JSON object. - assert that the `links` property includes an `[ogc-rel:dggrs]` link to the Discrete Global Grid Reference System description resource. - assert that the `links` property includes an `[ogc-rel:dggrs-definition]` link to the DGGRS definition, using the schema defined in Annex B — DGGRS Definitions or a later version.

A.6.3. Abstract Test for Requirement `zone-level` query parameter

Abstract test A.13	
IDENTIFIER	`/conf/zone-query/zone-level`
REQUIREMENT	Requirement 13: `/req/zone-query/zone-level`
TEST PURPOSE	Verify that the Implementation supports specifying a level at which to return a list of DGGS zones using a `zone-level` query parameter
TEST METHOD	**Given:** a DGGRS zones query resource identified from a `[ogc-rel:dggrs-zone-query]` link which passed the `/zone-query/zones-list` abstract test **When:** performing a GET operation on the …`/dggs/{dggrsId}/zones` resource with a `zone-level` query parameter **Then:** - assert that the Implementation supports a `zone-level` query parameter for the zone query operation (resource path ending with …`/dggs/{dggrsId}/zones`). - assert that if a compact zones list is returned (which is the default, unless the `compact-zones` parameter is set to *false*), the zones returned in the response are of the DGGRS hierarchy level specified by the `zone-level` query parameter, or of a lower hierarchy level standing in for a compact representation of multiple zones at the requested hierarchy level. - assert that if a non-compact zones list is returned (if the `compact-zones` query parameter is set to *false*), the zones returned in the response are of the DGGRS hierarchy level specified by the `zone-level` query parameter.

A.6.4. Abstract Test for Requirement `compact-zones` query parameter

Abstract test A.14	
IDENTIFIER	`/conf/zone-query/compact-zones`
REQUIREMENT	Requirement 14: `/req/zone-query/compact-zones`

Abstract test A.14	
TEST PURPOSE	Verify that the Implementation supports specifying whether to retrieve a list of DGGS zones using a compact-zones query parameter
TEST METHOD	**Given:** a DGGRS zones query resource identified from a `[ogc-rel:dggrs-zone-query]` link which passed the `/zone-query/zones-list` abstract test **When:** performing GET operations on the …/`dggs/{dggrsId}/zones` resource with a `compact-zones` query parameter set to true and false **Then:** - assert that the Implementation supports a Boolean `compact-zones` query parameter for the zone query operation (resource path ending with …/`dggs/{dggrsId}/zones`), where a value of `true` corresponds to the default behavior when the query parameter is not specified, and a value of `false` disables the use of compact-zones in the response. - assert that when the `compact-zones` query parameter is set to *false*, the zones list response is not a compact list, and explicitly lists every individual zone at the requested or default DGGRS hierarchy level. - assert that when the `compact-zones` query parameter is set to *true* (or unspecified), the zones list response is a compact list, where children zones completely covering the area of a parent zone are replaced by that parent zone, in a recursive manner all the way to the lowest DGGRS hierarchy level.

A.6.5. Abstract Test for Requirement `parent-zone` query parameter

Abstract test A.15	
IDENTIFIER	`/conf/zone-query/parent-zone`
REQUIREMENT	Requirement 15: `/req/zone-query/parent-zone`
TEST PURPOSE	Verify that the Implementation supports specifying a parent zone within which to restrict zone listing using a `parent-zone` query parameter
TEST METHOD	**Given:** a DGGRS zones query resource identified from a `[ogc-rel:dggrs-zone-query]` link which passed the `/zone-query/zones-list` abstract test **When:** performing GET operations on the …/`dggs/{dggrsId}/zones` resource with a `parent-zone` query parameter set to a valid zone identifier for the selected DGGRS **Then:** - assert that the Implementation supports a `parent-zone` query parameter accepting a textual zone identifier. - assert that when specified, the response does not contain zones which are not this parent zone itself or a sub-zone of that zone.

A.6.6. Abstract Test for Requirement `bbox` query parameter

Abstract test A.16	
IDENTIFIER	`/conf/zone-query/bbox`
REQUIREMENT	Requirement 16: `/req/zone-query/bbox`
TEST PURPOSE	Verify that the Implementation supports specifying a spatial bounding box for which to return a list of DGGS zones
TEST METHOD	**Given:** a DGGRS zones query resource identified from a `[ogc-rel:dggrs-zone-query]` link which passed the `/zone-query/zones-list` abstract test **When:** performing GET operations on the …`/dggs/{dggrsId}/zones` resource with a `bbox` query parameter set to a geospatial extent in OGC:CRS84 within the extent of the dataset or collection being tested **Then:** - assert that the Implementation supports a `bbox` query parameter for the zone query operation (resource path ending with …`/dggs/{dggrsId}/zones`) as defined in Requirement 16: `/req/zone-query/bbox` - assert that the `bbox` query parameter is interpreted as a comma separated list of four or six floating point numbers, that if the bounding box consists of six numbers, the first three numbers are interpreted as the coordinates of the lower bound corner of a three-dimensional bounding box and the last three are interpreted as the coordinates of the upper bound corner. - assert that the axis order is determined by the `bbox-crs` query parameter value or longitude and latitude if the query parameter is omitted (https://www.opengis.net/def/crs/OGC/1.3/CRS84 axis order for a 2D bounding box, https://www.opengis.net/def/crs/OGC/1.3/CRS84h for a 3D bounding box). - assert that the returned list of zone IDs only contain zones inside or intersecting with the spatial extent of the geographical area of the bounding box.

A.6.7. Abstract Test for Requirement `bbox-crs` query parameter

Abstract test A.17	
IDENTIFIER	`/conf/zone-query/bbox-crs`
REQUIREMENT	Requirement 17: `/req/zone-query/bbox-crs`
TEST PURPOSE	Verify that the Implementation supports specifying the CRS used for the `bbox` query parameter using the `bbox-crs` parameter
TEST METHOD	**Given:** a DGGRS zones query resource identified from a `[ogc-rel:dggrs-zone-query]` link which passed the `/zone-query/zones-list` abstract test and a collection **When:** performing GET operations on the …`/dggs/{dggrsId}/zones` resource with a `bbox` query parameter set to a geospatial extent in either OGC:CRS84 or in the `storageCrs` of the dataset or collection within the extent of the dataset or collection being tested and a `bbox-crs` set to that CRS in which the extent is specified

Abstract test A.17	
	Then: - assert that the list of zones resource supports a `bbox-crs` query parameter specifying the CRS used for the `bbox` query parameter. - assert that for Earth centric data, the Implementation supports https://www.opengis.net/def/crs/OGC/1.3/CRS84 as a value. - assert that if the `bbox-crs` is not indicated https://www.opengis.net/def/crs/OGC/1.3/CRS84 is assumed. - assert that the native CRS (`storageCrs`) is supported as a value. - assert that both CRS expressed as URIs and as safe CURIEs are supported. - assert that if the `bbox` query parameter is not used, the `bbox-crs` is ignored.

A.6.8. Abstract Test for Requirement `subset` query parameter

Abstract test A.18	
IDENTIFIER	`/conf/zone-query/subset`
REQUIREMENT	Requirement 18: `/req/zone-query/subset`
TEST PURPOSE	Verify that the Implementation supports specifying a spatial bounding box for which to return a list of DGGS zones
TEST METHOD	**Given:** a DGGRS zones query resource identified from a `[ogc-rel:dggrs-zone-query]` link which passed the `/zone-query/zones-list` abstract test **When:** performing GET operations on the …`/dggs/{dggrsId}/zones` resource with a `subset` query parameter set to a geospatial extent in OGC:CRS84 within the extent of the dataset or collection being tested, time for temporal dataset or collection, and another dimension (e.g., atmospheric pressure level) if supported by the dataset or collection **Then:** - assert that the Implementation supports a `subset` query parameter for the zone query operation (resource path ending with …`/dggs/{dggrsId}/zones`) conforming to the ABNF defined in Requirement 18: `/req/zone-query/subset`. - assert that the Implementation supports as axis names `Lat` and `Lon` for geographic CRS and `E` and `N` for projected CRS, which are to be interpreted as the best matching spatial axis in the CRS definition. - assert that if a third spatial dimension is supported (if the resource's spatial extent bounding box is three dimensional), the Implementation also supports a `h` dimension (elevation above the ellipsoid in EPSG:4979 or CRS84h) for geographic CRS and `z` for projected CRS, which are to be interpreted as the vertical axis in the CRS definition. - assert that the Implementation supports as axis names `time` for a temporal dataset. - assert that the Implementation supports as axis names any additional dimension (beyond spatial and temporal) as described in the `extent` property of the collection or dataset description.

Abstract test A.18	
	- assert that the Implementation returns a 400 error status code if an axis name does not correspond to one of the axes of the Coordinate Reference System (CRS) of the data or an axis defined in the relevant `extent` property. - assert that for a CRS where an axis can wrap around, such as subsetting across the dateline (anti-meridian) in a geographic CRS, a *low* value greater than *high* is supported to indicate an extent crossing that wrapping point. - assert that the Implementation interprets the coordinates as values for the named axis of the CRS specified in the `subset-crs` query parameter value or in https://www.opengis.net/def/crs/OGC/1.3/CRS84 (https://www.opengis.net/def/crs/OGC/1.3/CRS84h for vertical dimension) if the `subset-crs` query parameter is missing. - assert that if the `subset` query parameter including any of the dimensions corresponding to those of the map bounding box is used with a `bbox`, the server returns a 400 client error. - assert that the Implementation interprets multiple `subset` query parameters, as if all dimension subsetting values were provided in a single `subset` query parameter (comma separated).

A.6.9. Abstract Test for Requirement `subset-crs` query parameter

Abstract test A.19	
IDENTIFIER	`/conf/zone-query/subset-crs`
REQUIREMENT	Requirement 19: `/req/zone-query/subset-crs`
TEST PURPOSE	Verify that the Implementation supports specifying the CRS used for the `bbox` query parameter using the `bbox-crs` query parameter
TEST METHOD	**Given:** a DGGRS zones query resource identified from a `[ogc-rel:dggrs-zone-query]` link which passed the `/zone-query/zones-list` abstract test and a collection **When:** performing GET operations on the …`/dggs/{dggrsId}/zones` resource with a `subset` query parameter set to a geospatial extent in either OGC:CRS84 or in the `storageCrs` of the dataset or collection within the extent of the dataset or collection being tested and a `subset-crs` set to that CRS in which the extent is specified **Then:** - assert that the zone listing operation supports a query parameter `subset-crs` identifying the CRS in which the `subset` query parameter is specified with a URI or safe CURIE. - assert that for Earth centric data, https://www.opengis.net/def/crs/OGC/1.3/CRS84 as a value is supported. - assert that if the `subset-crs` is not indicated, https://www.opengis.net/def/crs/OGC/1.3/CRS84 is assumed. - assert that the native CRS (`storageCrs`) are supported as a value. Other requirements classes may allow additional values (see `crs` query parameter definition). - assert that CRSs expressed both as URIs or as safe CURIEs are supported. - assert that if no `subset` query parameter referring to an axis of the CRS is used, the `subset-crs` is ignored.

A.6.10. Abstract Test for Requirement `datetime` query parameter

Abstract test A.20	
IDENTIFIER	`/conf/zone-query/datetime`
REQUIREMENT	Requirement 20: `/req/zone-query/datetime`
TEST PURPOSE	Verify that the Implementation supports specifying a multi-dimensional subset for which to return a list of DGGS zones
TEST METHOD	**Given:** a DGGRS zones query resource identified from a `[ogc-rel:dggrs-zone-query]` link which passed the `/zone-query/zones-list` abstract test **When:** performing GET operations on the …`/dggs/{dggrsId}/zones` resource with a `datetime` query parameter set to a temporal extent within the dataset or collection's temporal extent **Then:** - assert that the Implementation supports a `datetime` query parameter expressed corresponding to either a date-time instant or a time interval, conforming to the ABNF in Requirement 20: `/req/zone-query/datetime`. - assert that the implementation supports an `instant` defined as specified by RFC 3339, 5.6, with the exception that the server is only required to support the `Z` UTC time notation, and not required to support local time offsets. - assert that only the zones with data whose geometry intersect with the specified temporal interval are part of the zone list response. - assert that time intervals unbounded at the start or at the end are supported using a double-dot (`..`) or an empty string for the start/end. - assert that if a `datetime` query parameter is specified requesting zone data where no temporal dimension applies, the Implementation either ignores the query parameter or returns a 4xx client error.

A.7. Conformance Class "Filtering Zone Queries with CQL2"

Conformance class A.7	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-query-cql2-filter`
REQUIREMENTS CLASS	Requirements class 7: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-query-cql2-filter`

Conformance class A.7	
TARGET TYPE	Web API
CONFORMANCE TEST	Abstract test A.21: `/conf/zone-query-cql2-filter/filter`

A.7.1. Abstract Test for Requirement `filter` query parameter (for zone queries)

Abstract test A.21	
IDENTIFIER	`/conf/zone-query-cql2-filter/filter`
REQUIREMENT	Requirement 21: `/req/zone-query-cql2-filter/filter`
TEST PURPOSE	Verify that the Implementation supports specifying a CQL2 expression filtering the zones returned from a zone query
TEST METHOD	**Given:** a DGGRS zones query resource identified from a `[ogc-rel:dggrs-zone-query]` link which passed the `/zone-query/zones-list` abstract test **When:** performing GET operations on the …`/dggs/{dggrsId}/zones` resource with a `filter` query parameter set to a CQL2 expression filtering zones to return **Then:** - assert that the Implementation supports a `filter` query parameter specified using the CQL2-Text encoding of the OGC Common Query Language for the zone query operation (resource path ending with …`/dggs/{dggrsId}/zones`). - assert that the list of returned zones are only those for which the CQL2 expression evaluates to true when considering the geometry and the data of the DGGS zones resource being queried. - assert that the CQL2 expression evaluator supports the queryables declared in the JSON Schema resource linked to from the origin of the DGGRS resources using the `[ogc-rel:queryables]` link relation type.

A.8. Conformance Class “Root DGGS”

Conformance class A.8	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/root-dggs`
REQUIREMENTS CLASS	Requirements class 8: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/root-dggs`

Conformance class A.8	
TARGET TYPE	Web API
CONFORMANCE TEST	Abstract test A.22: `/conf/root-dggs/dggs`

A.8.1. Abstract Test for Requirement Root DGGS

Abstract test A.22	
IDENTIFIER	`/conf/root-dggs/dggs`
REQUIREMENT	Requirement 22: `/req/root-dggs/dggs`
TEST PURPOSE	Verify that the Implementation supports API/dataset-wide DGGS resources
TEST METHOD	**Given:** an implementation declaring conformance to Root DGGS **When:** testing the `/conf/core` conformance class as well the `/conf/zone-query` and/or the `/conf/data-retrieval` conformance classes for the DGGS resources originating from the root origin at `/dggs` **Then:** - assert that the Implementation supports the resource paths and associated HTTP methods defined in the "Core" requirements class, as well as any additional resources defined in other requirements classes to which the Implementation declares conformance, for the root of the API as an origin. If the API allows to retrieve and/or query data, this means that the data returned or queried considers all collections comprising the dataset (unless an extension is used to select specific collections e.g., using a `collections` query parameter). Note that the root DGGRS origin could also be used in API deployment not associated with any dataset, solely for the purpose of exploring and demonstrating the DGGRS. - assert that the Implementation includes a link to the list of available DGGRSs for the API/dataset at `/dggs` in the links of the landing page using the link relation type `[ogc-rel:dggrs-list]`. - assert that if the API deployment supports querying or retrieving data from a dataset, the `/dggs` and `/dggs/{dggrsId}` resources include a link to the landing page using the link relation type `[ogc-rel:dataset]`.

A.9. Conformance Class "Collection DGGS"

Conformance class A.9	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/collection-dggs`
REQUIREMENTS CLASS	Requirements class 9: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/collection-dggs`
TARGET TYPE	Web API
CONFORMANCE TEST	Abstract test A.23: `/conf/collection-dggs/dggs`

A.9.1. Abstract Test for Requirement Collection DGGS

Abstract test A.23	
IDENTIFIER	`/conf/collection-dggs/dggs`
REQUIREMENT	Requirement 23: `/req/collection-dggs/dggs`
TEST PURPOSE	Verify that the Implementation supports collection DGGS resources
TEST METHOD	**Given:** an implementation declaring conformance to Collection DGGS **When:** testing the `/conf/core` conformance class as well the `/conf/zone-query` and/or the `/conf/data-retrieval` conformance classes for the DGGS resources originating from collections at `/collections/{collectoinId}/dggs` **Then:** - assert that the Implementation supports the resource paths and associated HTTP methods defined in the “Core” requirements class, as well as any additional resources defined in other requirements classes to which the Implementation declares conformance, for at least one collection of the dataset offered by the API deployment. - assert that the Implementation includes a link to the list of available DGGRSs for the collection at `/collections/{collectionId}/dggs` in the links of the collection using the link relation type `[ogc-rel:dggrs-list]`. - assert that the `/collections/{collectionId}/dggs` and `/collections/{collectionId}/dggs/{dggrsId}` resources includes a link to the collection using the link relation type `[ogc-rel:geodata]`.

A.10. Conformance Class “DGGS-JSON zone data encoding”

Conformance class A.10	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-json`
REQUIREMENTS CLASS	Requirements class 10: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-json`
TARGET TYPE	Web API
CONFORMANCE TEST	Abstract test A.24: `/conf/data-json/content`

A.10.1. Abstract Test for Requirement DGGS-JSON Zone data encoding

Abstract test A.24	
IDENTIFIER	`/conf/data-json/content`
REQUIREMENT	Requirement 24: `/req/data-json/content`
TEST PURPOSE	Verify that the Implementation supports encoding zone data as DGGS-JSON
TEST METHOD	**Given:** a DGGRS zone data resource identified from a `[ogc-rel:dggrs-zone-data]` link template within a DGGRS description resource, replacing the `{zoneId}` variable with a known valid identifier for the DGGRS, not beyond the `maxRefinementLevel` if one is specified **When:** performing a GET operation on the …`/dggs/{dggrsId}/zones/{zoneID}/data` resource with media type `application/json` for DGGS-JSON **Then:** - assert that every 200 response of the server for zone data with the media type `application/json` is a JSON document representing the data values for all selected fields of each included sub-zone. - assert that the schema for the JSON document follows the JSON Schema for DGGS-JSON described in Requirement 24: `/req/data-json/content` - assert that every zone depth requested is using the `zone-depth` query parameter included in the response. - assert thatt at every depth, each individual value corresponds exactly to the data sampled representative of that sub-zone.

Abstract test A.24	
	- assert that the list of data values follows the default zone order as specified by the Discrete Global Grid Reference System (for example based on a scanline or space-filling curved defined therein) for which the request is made. - assert that Null values use the `null` JSON value.

A.11. Conformance Class “DGGS-UBJSON zone data encoding”

Conformance class A.11	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-ubjson`
REQUIREMENTS CLASS	Requirements class 11: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-ubjson`
TARGET TYPE	Web API
CONFORMANCE TEST	Abstract test A.25: `/conf/data-ubjson/content`

A.11.1. Abstract Test for Requirement DGGS-UBJSON Zone data encoding

Abstract test A.25	
IDENTIFIER	`/conf/data-ubjson/content`
REQUIREMENT	Requirement 25: `/req/data-ubjson/content`
TEST PURPOSE	Verify that the Implementation supports encoding zone data as DGGS-UBJSON
TEST METHOD	**Given:** a DGGRS zone data resource identified from a `[ogc-rel:dggrs-zone-data]` link template within a DGGRS description resource, replacing the `{zoneId}` variable with a known valid identifier for the DGGRS, not beyond the `maxRefinementLevel` if one is specified **When:** performing a GET operation on the …`/dggs/{dggrsId}/zones/{zoneID}/data` resource with media type `application/ubjson` for DGGS-JSON **Then:**

Abstract test A.25	
	- assert that every 200 response of the server for zone data with the media type `application/ubjson` is a Universal Binary JSON document representing the data values for all selected fields of each included sub-zone. - assert that the schema for the UBJSON document follows the same JSON Schema as for DGGS-JSON described in Requirement 24: `/req/data-json/content`. - assert that every zone depth requested using the `zone-depth` query parameter is included in the response. - assert that at every depth, each individual value corresponds exactly to the data sampled representative of that zone geometry. - assert that the list of data values follows the default zone order as specified by the Discrete Global Grid Reference System (for example based on a scanline or space-filling curved defined therein) for which the request is made. - assert that Null values use the `null` JSON value.

A.12. Conformance Class "DGGS-JSON-FG zone data encoding"

Conformance class A.12	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-dggs-jsonfg`
REQUIREMENTS CLASS	Requirements class 12: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-dggs-jsonfg`
TARGET TYPE	Web API
CONFORMANCE TEST	Abstract test A.26: `/conf/data-dggs-jsonfg/content`

A.12.1. Abstract Test for Requirement DGGS-JSON-FG Zone data encoding

Abstract test A.26	
IDENTIFIER	`/conf/data-dggs-jsonfg/content`
REQUIREMENT	Requirement 26: `/req/data-dggs-jsonfg/content`

Abstract test A.26	
TEST PURPOSE	Verify that the Implementation supports encoding zone data as DGGS-JSON-FG
TEST METHOD	**Given:** a DGGRS zone data resource identified from a `[ogc-rel:dggrs-zone-data]` link template within a DGGRS description resource, replacing the `{zoneId}` variable with a known valid identifier for the DGGRS, not beyond the `maxRefinementLevel` if one is specified **When:** performing a GET operation on the …`/dggs/{dggrsId}/zones/{zoneID}/data` resource with media type `application/geo+json` and a `profile` query parameter value of `jsonfg-dggs`, `jsonfg-dggs-plus`, `jsonfg-dggs-zoneids` or `jsonfg-dggs-zoneids-plus` for DGGS-JSON-FG **Then:** - assert that the implementation supports zone data requests negotiating an `application/geo+json` media type combined with a `profile` query parameter with a value of `jsonfg-dggs`, `jsonfg-dggs-plus`, `jsonfg-dggs-zoneids` or `jsonfg-dggs-zoneids-plus` allowing a client to request a DGGS-JSON-FG response. - assert that every 200 response of the server for DGGS-JSON-FG zone data requests is an extended Features & Geometry JSON (JSON-FG) document representing the vector features within the requested zone. - assert that the JSON document follows the JSON Schema for JSON-FG, with geometry stored in a `dggsPlace` property mirroring the JSON-FG `place` schema and supporting all of the same feature types, but where `coordinates` use sub-zone indices (for `profile=jsonfg-dggs` or `jsonfg-dggs-plus`) or global textual zone identifiers (for `profile=jsonfg-dggs-zoneids` or `jsonfg-dggs-zoneids-plus`) as described in later parts of this requirement. - assert that for `profile=jsonfg-dggs-plus` and `jsonfg-dggs-zoneids-plus`, the JSON document also includes geometry stored in `geometry` or `place` using traditional coordinate reference system coordinates for compatibility. - assert that the JSON document additionally contains a `dggrs` property set to the URI or CURIEs of the DGGRS of the response, which replaces the `coordRefSys` property and concept. - assert that the JSON document additionally contains a `zoneId` property indicating the requested reference zone. - assert that the JSON document additionally contains a `depth` property corresponding to the relative zone depth (as specified in the single depth format of the OGC API — DGGS `zone-depth` query parameter for zone data retrieval), determining how coordinates are resolved from sub-zone indices (in the case of `profile=jsonfg-dggs` and `jsonfg-dggs-plus`), and as a result, their precision. - assert that for `profile=jsonfg-dggs` and `jsonfg-dggs-plus`, the JSON document contains a `dggsPlace` property specifying geometry using local sub-zone indices ranging from 1 to the number of sub-zones within the reference parent zone at the declared depth, based on the deterministic sub-zone order of the DGGRS. - assert that for `profile=jsonfg-dggs-zoneids` and `jsonfg-dggs-zoneids-plus`, the JSON document contains a `dggsPlace` property specifying geometry using global textual zone identifiers (as defined in the DGGRS zone indexing reference system) at the declared depth. - assert that when all spatial coordinates are dimensions of the DGGRS, the array of coordinates including the square brackets are replaced by the local index or global identifier of the sub-zone within which the coordinates are located. - assert that when additional spatial coordinates that are not part of the DGGRS, such as elevation above ground for a 2D DGGRS, are needed to define the geometry coordinates, only

Abstract test A.26	
	the coordinates corresponding to DGGRS dimensions are replaced by the local index or global identifier of the sub-zone. - assert that only features fully or partly located within the reference zone are included. - assert that for feature geometry intersecting the boundaries of the reference zone, the sub-zone within which the intersection is located are included, with a separate single `0` special sub-zone index (in the case of `profile=jsonfg-dggs` and `jsonfg-dggs-plus`) or a `null` value (in the case of `profile=jsonfg-dggs-zoneids` and `jsonfg-dggs-zoneids-plus`) representing all outside points before a point inside, after a point inside, or in between two inside points. - assert that for geometry in two or more dimensions with an outside point (identified by a `0` sub-zone index for `profile=jsonfg-dggs` and `jsonfg-dggs-plus`, or `null` for `profile=jsonfg-dggs-zoneids` and `jsonfg-dggs-zoneids-plus`) and which is necessarily between two contour points, the portion of the zone edge(s) between the exit and entry point are considered inside of that contour. - assert that the URI of the selected or default DGGS-JSON-FG profile are included in the `links` section of the JSON-FG response.

A.13. Conformance Class "DGGS-UBJSON-FG zone data encoding"

Conformance class A.13	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-dggs-ubjsonfg`
REQUIREMENTS CLASS	Requirements class 13: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-dggs-ubjsonfg`
TARGET TYPE	Web API
CONFORMANCE TEST	Abstract test A.27: `/conf/data-dggs-ubjsonfg/content`

A.13.1. Abstract Test for Requirement DGGS-UBJSON-FG Zone data encoding

Abstract test A.27	
IDENTIFIER	`/conf/data-dggs-ubjsonfg/content`

Abstract test A.27	
REQUIREMENT	Requirement 27: `/req/data-dggs-ubjsonfg/content`
TEST PURPOSE	Verify that the Implementation supports encoding zone data as DGGS-UBJSON-FG
TEST METHOD	**Given:** a DGGRS zone data resource identified from a `[ogc-rel:dggrs-zone-data]` link template within a DGGRS description resource, replacing the `{zoneId}` variable with a known valid identifier for the DGGRS, not beyond the `maxRefinementLevel` if one is specified **When:** performing a GET operation on the …`/dggs/{dggrsId}/zones/{zoneID}/data` resource with media type `application/geo+ubjson` and a `profile` query parameter value of `jsonfg-dggs`, `jsonfg-dggs-plus`, `jsonfg-dggs-zoneids` or `jsonfg-dggs-zoneids-plus` for DGGS-UBJSON-FG **Then:** - assert that the Implementation supports zone data requests negotiating an `application/geo+ubjson` media type combined with a `profile` query parameter with a value of `jsonfg-dggs`, `jsonfg-dggs-plus`, `jsonfg-dggs-zoneids` or `jsonfg-dggs-zoneids-plus` allowing a client to request a DGGS-UBJSON-FG response. - assert that every 200 response of the server for DGGS-UBJSON-FG zone data requests is a Universal Binary JSON document representing the vector features within the requested zone. - assert that the content of the UBJSON document passes all of the tests described in Requirement 24: `/req/data-json/content` adapted for a UBJSON encoding of the JSON data.

A.14. Conformance Class “GeoTIFF zone data encoding”

Conformance class A.14	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-geotiff`
REQUIREMENTS CLASS	Requirements class 14: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-geotiff`
TARGET TYPE	Web API
CONFORMANCE TEST	Abstract test A.28: `/conf/data-geotiff/content`

A.14.1. Abstract Test for Requirement GeoTIFF Zone data encoding

Abstract test A.28	
IDENTIFIER	`/conf/data-geotiff/content`
REQUIREMENT	Requirement 28: `/req/data-geotiff/content`
TEST PURPOSE	Verify that the Implementation supports encoding zone data as GeoTIFF
TEST METHOD	**Given:** a DGGRS zone data resource identified from a `[ogc-rel:dggrs-zone-data]` link template within a DGGRS description resource, replacing the `{zoneId}` variable with a known valid identifier for the DGGRS, not beyond the `maxRefinementLevel` if one is specified **When:** performing a GET operation on the …`/dggs/{dggrsId}/zones/{zoneID}/data` resource with media type `image/tiff; application=geotiff` for GeoTIFF **Then:** - assert that every 200 response of the server for zone data with the media type `image/tiff` is a TIFF image representing the data values for all selected fields within the zone for which data is requested. - assert that if the TIFF encoding incorporates a GeoTIFF georeference, this information is consistent with the DGGRS Zone ID. - assert that if the zone geometry is not rectangular, the closest bounding rectangle is used for referencing the image. - assert that there is a distinct value for each sub-zone implied from the requested `zone-depth` has at least one distinct cell value in the response. - assert that for implementations supporting Data Custom Depths, each depth of the requested zone depth pyramid is a separate image (overview) in the response.

A.15. Conformance Class “GeoJSON zone data encoding”

Conformance class A.15	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-geojson`
REQUIREMENTS CLASS	Requirements class 15: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-geojson`
TARGET TYPE	Web API
CONFORMANCE TEST	Abstract test A.29: `/conf/data-geojson/content`

A.15.1. Abstract Test for Requirement GeoJSON Zone data encoding

Abstract test A.29	
IDENTIFIER	`/conf/data-geojson/content`
REQUIREMENT	Requirement 29: `/req/data-geojson/content`
TEST PURPOSE	Verify that the Implementation supports encoding zone data as GeoJSON
TEST METHOD	**Given:** a DGGRS zone data resource identified from a `[ogc-rel:dggrs-zone-data]` link template within a DGGRS description resource, replacing the `{zoneId}` variable with a known valid identifier for the DGGRS, not beyond the `maxRefinementLevel` if one is specified **When:** performing a GET operation on the …`/dggs/{dggrsId}/zones/{zoneID}/data` resource with media type `application/geo+json` for GeoJSON **Then:** - assert that every 200 response of the server for zone data with the media type `application/geo+json` is a GeoJSON document representing the features, including their geometry and associated properties, within the zone for which data is requested. - assert that unless otherwise specified by a prior arrangement (for example, an output `crs` query parameter) or by negotiating a JSON-FG profile, the coordinate reference system is CRS84(h) in longitude and latitude (and optional height above the WGS84 ellipsoid). - assert that features whose geometry lie wholly outside of the zone geometry are not included in the response. - assert that if returning a JSON-FG profile, the profile URI is included in the `links` section of the JSON-FG response.

A.16. Conformance Class “netCDF zone data encoding”

Conformance class A.16	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-netcdf`
REQUIREMENTS CLASS	Requirements class 16: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-netcdf`
TARGET TYPE	Web API
CONFORMANCE TEST	Abstract test A.30: `/conf/data-netcdf/content`

A.16.1. Abstract Test for Requirement netCDF Zone data encoding

Abstract test A.30	
IDENTIFIER	`/conf/data-netcdf/content`
REQUIREMENT	Requirement 30: `/req/data-netcdf/content`
TEST PURPOSE	Verify that the Implementation supports encoding zone data as netCDF
TEST METHOD	**Given:** a DGGRS zone data resource identified from a `[ogc-rel:dggrs-zone-data]` link template within a DGGRS description resource, replacing the `{zoneId}` variable with a known valid identifier for the DGGRS, not beyond the `maxRefinementLevel` if one is specified **When:** performing a GET operation on the …`/dggs/{dggrsId}/zones/{zoneID}/data` resource with media type `application/netcdf` for netCDF **Then:** - assert that every 200 response of the server for zone data with the media type `application/netcdf` is a netCDF file representing the data values for all selected fields of each included sub-zone. - assert that the response is encoded using a netCDF encoding extension, such as the NetCDF classic and 64-bit offset format, or netCDF version 4 based on OGC HDF5. - assert that for a DGGS-optimized profile (`netcdf3-dggs`, `netcdf3-dggs-zoneids`, `netcdf4-dggs`, `netcdf3-dggs-zoneids`), the profile URI, the global identifier of the parent zone and a URI or link to the DGGRS definition is included as metadata. - assert that for a DGGS-optimized profile, coordinates along one dimension corresponds to sub-zones, either as global zone identifiers (`netcdf3-dggs-zoneids`, `netcdf4-dggs-zoneids`) or local sub-zone indices following the deterministic DGGRS sub-zone order (`netcdf3-dggs`, `netcdf4-dggs`), while spatiotemporal dimensions of the DGGRS is not defined as separate dimensions. - assert that for non-DGGS-optimized profiles, if the netCDF data incorporates georeferencing information, this information is consistent with the DGGRS Zone ID. - assert that for non-DGGS-optimized profiles, if the zone geometry is not rectilinear, the closest bounding rectangle (or volume) is used for referencing the data. - assert that for non-DGGS-optimized profiles, a distinct value for each sub-zone implied from the requested `zone-depth` corresponds to at least one distinct cell value in the response.

A.17. Conformance Class “Zarr zone data encoding”

Conformance class A.17	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-zarr`
REQUIREMENTS CLASS	Requirements class 17: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-zarr`
TARGET TYPE	Web API
CONFORMANCE TEST	Abstract test A.31: `/conf/data-zarr/content`

A.17.1. Abstract Test for Requirement Zarr Zone data encoding

Abstract test A.31	
IDENTIFIER	`/conf/data-zarr/content`
REQUIREMENT	Requirement 31: `/req/data-zarr/content`
TEST PURPOSE	Verify that the Implementation supports encoding zone data as zipped Zarr 2
TEST METHOD	**Given:** a DGGRS zone data resource identified from a `[ogc-rel:dggrs-zone-data]` link template within a DGGRS description resource, replacing the `{zoneId}` variable with a known valid identifier for the DGGRS, not beyond the `maxRefinementLevel` if one is specified **When:** performing a GET operation on the …`/dggs/{dggrsId}/zones/{zoneID}/data` resource with media type `application/zarr+zip` for zipped Zarr 2 **Then:** - assert that every 200 response of the server for zone data with the media type `application/zarr+zip` is a Zip file containing the data values for all selected fields of each included sub-zone encoded following the Zarr Storage Specification Version 2 file. - assert that for a DGGS-optimized profile (`zarr2-dggs`, `zarr2-dggs-zoneids`), the profile URI, the global identifier of the parent zone and a URI or link to the DGGRS definition is included as metadata. - assert that for a DGGS-optimized profile, coordinates along one dimension corresponds to sub-zones, either as global zone identifiers (`zarr2-dggs-zoneids`) or local sub-zone indices following the deterministic DGGRS sub-zone order (`zarr2-dggs`), while spatiotemporal dimensions of the DGGRS is not defined as separate dimensions. - assert that for a DGGS-optimized profile, at every depth, each individual value corresponds exactly to the data sampled representative of that sub-zone. - assert that for non-DGGS-optimized profiles, if the Zarr data incorporates georeferencing information (such as defined by the candidate OGC GeoZarr Standard), this information is consistent with the DGGRS Zone ID. - assert that for non-DGGS-optimized profiles, if the zone geometry is not rectilinear, the closest bounding rectangle (or volume) is used for referencing the data.

Abstract test A.31	
	- assert that for non-DGGS-optimized profiles, a distinct value for each sub-zone implied from the requested `zone-depth` corresponds to at least one distinct cell value in the response. - assert that if multiple relative zone depths are being returned, each depth is encoded as a separate Zarr dataset within the data packet. - assert that every zone depth requested using the `zone-depth` query parameter is included in the response.

A.18. Conformance Class "CoverageJSON zone data encoding"

Conformance class A.18	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-coveragejson`
REQUIREMENTS CLASS	Requirements class 18: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-coveragejson`
TARGET TYPE	Web API
CONFORMANCE TEST	Abstract test A.32: `/conf/data-coveragejson/content`

A.18.1. Abstract Test for Requirement CoverageJSON Zone data encoding

Abstract test A.32	
IDENTIFIER	`/conf/data-coveragejson/content`
REQUIREMENT	Requirement 32: `/req/data-coveragejson/content`
TEST PURPOSE	Verify that the Implementation supports encoding zone data as CoverageJSON
TEST METHOD	**Given:** a DGGRS zone data resource identified from a `[ogc-rel:dggrs-zone-data]` link template within a DGGRS description resource, replacing the `{zoneId}` variable with a known valid identifier for the DGGRS, not beyond the `maxRefinementLevel` if one is specified **When:** performing a GET operation on the …`/dggs/{dggrsId}/zones/{zoneID}/data` resource with media type `application/prs.coverage+json` for CoverageJSON **Then:**

Abstract test A.32	
	- assert that every 200 response of the server for zone data with the media type `application/prs.coverage+json` is a CoverageJSON file representing the data values for all selected fields of each included sub-zone. - assert that for a DGGS-optimized profile (`covjson-dggs`, `covjson-dggs-zoneids`), the profile URI, the global identifier of the parent zone and a URI or link to the DGGRS definition is included as metadata. - assert that for a DGGS-optimized profile, coordinates along one dimension corresponds to sub-zones, either as global zone identifiers (`covjson-dggs-zoneids`) or local sub-zone indices following the deterministic DGGRS sub-zone order (`covjson-dggs`), while spatiotemporal dimensions of the DGGRS is not be defined as separate dimensions. - assert that for non-DGGS-optimized profiles, if the CoverageJSON encoding incorporates georeferencing information, this information is consistent with the DGGRS Zone ID. - assert that for non-DGGS-optimized profiles, if the zone geometry is not rectilinear, the closest bounding rectangle (or volume) is used for referencing the data. - assert that for non-DGGS-optimized profiles, a distinct value for each sub-zone implied from the requested `zone-depth` corresponds to at least one distinct cell value in the response.

A.19. Conformance Class “JPEG XL zone data encoding”

Conformance class A.19	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-jpegxl`
REQUIREMENTS CLASS	Requirements class 19: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-jpegxl`
TARGET TYPE	Web API
CONFORMANCE TEST	Abstract test A.33: `/conf/data-jpegxl/content`

A.19.1. Abstract Test for Requirement JPEG XL Zone data encoding

Abstract test A.33	
IDENTIFIER	`/conf/data-jpegxl/content`
REQUIREMENT	Requirement 33: `/req/data-jpegxl/content`

Abstract test A.33	
TEST PURPOSE	Verify that the Implementation supports encoding zone data as JPEG XL
TEST METHOD	**Given:** a DGGRS zone data resource identified from a `[ogc-rel:dggrs-zone-data]` link template within a DGGRS description resource, replacing the `{zoneId}` variable with a known valid identifier for the DGGRS, not beyond the `maxRefinementLevel` if one is specified **When:** performing a GET operation on the …`/dggs/{dggrsId}/zones/{zoneID}/data` resource with media type `image/jxl` for JPEG XL **Then:** - assert that every 200 response of the server for zone data with the media type `image/jxl` is a JPEG XL image representing the data values for all selected fields within the zone for which data is requested. - assert that if the zone geometry is not rectangular, the closest bounding rectangle is used for referencing the image. - assert that a distinct value for each sub-zone implied from the requested `zone-depth` corresponds to at least one distinct cell value in the response. - assert that for Implementations supporting Data Custom Depths, each depth of the requested zone depth pyramid is a separate image (overview) in the response.

A.20. Conformance Class “PNG zone data encoding “

Conformance class A.20	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-png`
REQUIREMENTS CLASS	Requirements class 20: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/data-png`
TARGET TYPE	Web API
CONFORMANCE TEST	Abstract test A.34: `/conf/data-png/content`

A.20.1. Abstract Test for Requirement PNG Zone data encoding

Abstract test A.34	
IDENTIFIER	`/conf/data-png/content`

Abstract test A.34	
REQUIREMENT	Requirement 34: `/req/data-png/content`
TEST PURPOSE	Verify that the Implementation supports encoding zone data as PNG
TEST METHOD	**Given:** a DGGRS zone data resource identified from a `[ogc-rel:dggrs-zone-data]` link template within a DGGRS description resource, replacing the `{zoneId}` variable with a known valid identifier for the DGGRS, not beyond the `maxRefinementLevel` if one is specified **When:** performing a GET operation on the …`/dggs/{dggrsId}/zones/{zoneID}/data` resource with media type `image/png` for PNG **Then:** - assert that every 200 response of the server for zone data with the media type `image/png` is a PNG image representing the data values for all selected fields within the zone for which data is requested. - assert that if the zone geometry is not rectangular, the closest bounding rectangle is used for referencing the image. - assert that a distinct value for each sub-zone implied from the requested `zone-depth` corresponds to at least one distinct cell value in the response. - assert that for Implementations supporting Data Custom Depths, each depth of the requested zone depth pyramid is a separate image (overview) in the response. - assert that a `Values-Scale:` response header with a numeric real value is returned indicating the factor by which values were multiplied before an offset was added to result in the encoded 8-bit or 16-bit PNG unsigned integer values. - assert that a `Values-Offset:` response header with a numeric real value is returned indicating the offset which was added after multiplying values by the scale factor to result in the encoded 8-bit or 16-bit PNG unsigned integer values.

A.21. Conformance Class “HTML zone list encoding”

Conformance class A.21	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-html`
REQUIREMENTS CLASS	Requirements class 21: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-html`
TARGET TYPE	Web API
CONFORMANCE TEST	Abstract test A.35: `/conf/zone-html/content`

A.21.1. Abstract Test for Requirement HTML zone list encoding

Abstract test A.35	
IDENTIFIER	`/conf/zone-html/content`
REQUIREMENT	Requirement 35: `/req/zone-html/content`
TEST PURPOSE	Verify that the Implementation supports encoding zone lists as HTML
TEST METHOD	**Given:** a DGGRS zones query resource identified from a `[ogc-rel:dggrs-zone-query]` link **When:** performing a GET operation on the …`/dggs/{dggrsId}/zones` resource with `text/html` media type for HTML **Then:** - assert that every 200 response of the server for zone query with the media type `text/html` is a HTML document listing the textual identifiers for all zones matching the query.

A.22. Conformance Class “Binary 64-bit integer zone list encoding”

Conformance class A.22	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-uint64`
REQUIREMENTS CLASS	Requirements class 22: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-uint64`
TARGET TYPE	Web API
CONFORMANCE TEST	Abstract test A.36: `/conf/zone-uint64/content`

A.22.1. Abstract Test for Requirement Binary 64-bit integer zone list encoding

Abstract test A.36	
IDENTIFIER	`/conf/zone-uint64/content`
REQUIREMENT	Requirement 36: `/req/zone-uint64/content`
TEST PURPOSE	Verify that the Implementation supports encoding zone lists as 64-bit unsigned integer
TEST METHOD	**Given:** a DGGRS zones query resource identified from a `[ogc-rel:dggrs-zone-query]` link **When:** performing a GET operation on the …`/dggs/{dggrsId}/zones` resource with `application/x-binary` media type for 64-bit unsigned integers **Then:** - assert that every 200 response of the server for zone query with the media type `application/x-binary` is a binary response consisting of a first 64-bit integer count defining the number of zones returned, followed by one 64-bit integer for each zone matching the query. - assert that the 64-bit integer identifiers are the ones defined by the DGGRS. - assert that the endianness of the returned count and zones integers are little endian. - assert that if the DGGRS does not define a 64-bit integer identifier, a 406 "Not Acceptable" response is returned.

A.23. Conformance Class "GeoJSON zone list encoding "

Conformance class A.23	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-geojson`
REQUIREMENTS CLASS	Requirements class 23: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-geojson`
TARGET TYPE	Web API
CONFORMANCE TEST	Abstract test A.37: `/conf/zone-geojson/content`

A.23.1. Abstract Test for Requirement GeoJSON zone list encoding

Abstract test A.37	
IDENTIFIER	`/conf/zone-geojson/content`

Abstract test A.37	
REQUIREMENT	Requirement 37: `/req/zone-geojson/content`
TEST PURPOSE	Verify that the Implementation supports encoding zone lists as GeoJSON
TEST METHOD	**Given:** a DGGRS zones query resource identified from a `[ogc-rel:dggrs-zone-query]` link **When:** performing a GET operation on the …`/dggs/{dggrsId}/zones` resource with `application/geo+json` media type for GeoJSON **Then:** - assert that every 200 response of the server for zone query with the media type `application/geo+json` is a GeoJSON document consisting of a FeatureCollection, where every Feature represents a single zone. - assert that unless otherwise specified by a prior arrangement (for example, an output `crs` query parameter) or by negotiating a JSON-FG profile, the coordinate reference system is CRS84(h) in longitude and latitude (and optional height above the WGS84 ellipsoid). - assert that every feature has a `zoneID` property corresponding to the textual identifier of the zone. - assert that the geometry of each feature is the geometry of the zone. - assert that if returning a JSON-FG profile, the profile URI is included in the `links` section of the JSON-FG response.

A.24. Conformance Class “GeoTIFF zone list encoding”

Conformance class A.24	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/zone-geotiff`
REQUIREMENTS CLASS	Requirements class 24: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/zone-geotiff`
TARGET TYPE	Web API
CONFORMANCE TEST	Abstract test A.38: `/conf/zone-geotiff/content`

A.24.1. Abstract Test for Requirement GeoTIFF zone list encoding

Abstract test A.38	
IDENTIFIER	`/conf/zone-geotiff/content`
REQUIREMENT	Requirement 38: `/req/zone-geotiff/content`
TEST PURPOSE	Verify that the Implementation supports encoding zone lists as GeoTIFF
TEST METHOD	**Given:** a DGGRS zones query resource identified from a `[ogc-rel:dggrs-zone-query]` link **When:** performing a GET operation on the …`/dggs/{dggrsId}/zones` resource with `image/tiff; application=geotiff` media type for GeoTIFF **Then:** - assert that every 200 response of the server for zone query with the media type `image/tiff` is a GeoTIFF document representing the zones matching the query in a geo-referenced image where each zone corresponds to at least one pixel. - assert that the GeoTIFF is encoded as `Pixel-Is-Area`.

A.25. Conformance Class “Operation IDs”

Conformance class A.25	
IDENTIFIER	`https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/operation-ids`
REQUIREMENTS CLASS	Requirements class 25: `https://www.opengis.net/spec/ogcapi-dggs-1/1.0/req/operation-ids`
TARGET TYPE	Web API
CONFORMANCE TEST	Abstract test A.39: `/conf/operation-ids/operation-ids`

A.25.1. Abstract Test for Operation IDs

Abstract test A.39	
IDENTIFIER	`/conf/operation-ids/operation-ids`
REQUIREMENT	Requirement 39: `/req/operation-ids/operation-ids`
TEST PURPOSE	Verify that the correct operation IDs suffixed are used in the API definition to identify DGGS resources

Abstract test A.39	
TEST METHOD	**Given:** an DGGS API implementation providing an API definition with a concept of operation IDs (such as OpenAPI 3.0) **When:** inspecting the operations in the API definition for DGGS resources **Then:** - assert that the operation IDs for DGGS resources correspond to those specified in Table 5

B

ANNEX B (INFORMATIVE) DISCRETE GLOBAL GRID REFERENCE SYSTEM DEFINITIONS

B

ANNEX B (INFORMATIVE) DISCRETE GLOBAL GRID REFERENCE SYSTEM DEFINITIONS

The following informative schema describes the expected response of the resource linked to by using the `[ogc-rel:dggrs-definition]` link relation from the DGGRS description. This behavior is specified by the Core requirements class. An important objective for this schema is to support establishing authoritative registries of DGGRSs providing a URI for a DGGRS corresponding to a specific definition.

Note that this schema is preliminary, and a normative version may be defined in a later version of the OGC API — DGGS Standard or by a future part of OGC Abstract Specification Topic 21. The schema may change to support a better description of existing and new classes of DGGH.

The DGGRS definition consists of three main components:

- The definition of the hierarchy of discrete global grids (`dggh`), including any parameterization such as the spheroid or orientation, establishing both the topology as well as the geometry of every zone,
- The Zonal Identifier Reference System (`zirs`), with textual identifiers being mandatory while 64-bit integer identifiers (which support the 64-bit integer zone list encoding requirements class) are optional,
- The sub-zone ordering (`subZoneOrder`), which allows for retrieving zone data using e.g., DGGS-JSON at a `zone-depth` other than 0 (the values of sub-zones can be provided as a 1D array of values with a pre-established association to sub-zones).

Whereas the ZIRS provides a mechanism to globally identify a single zone using a textual identifier (and alternatively using a 64-bit integer), the deterministic sub-zone order provides a local indexing mechanism to address smaller regions within a particular zone. This local indexing mechanism enables the DGGS-optimized data encodings defined by the DGGS-JSON and DGGS-JSON-FG requirements classes. The two mechanisms may potentially be related to each other, for example using the same space filling curve such as a Morton code (Z-order curve), but do not need to be. If defining a more optimal sub-zone order for a particular DGGRS is not practical, the ascending sort order of the sub-zone identifiers can be specified as a fallback, based on either the ASCII character strings of the textual identifiers, or the 64-bit integer identifiers.

B.1. JSON Schema for defining a DGGRS

```
{
  "$schema": "https://json-schema.org/draft/2019-09/schema",
  "description": "Discrete Global Grid System Reference System (DGGRS)
Definition.",
  "type": "object",
  "required": [
    "dggh",
    "zirs",
    "subZoneOrder"
  ],
  "properties": {
    "title": {
      "type": "string"
    },
    "description": {
      "type": "string"
    },
    "uri": {
      "type": "string",
      "format": "uri",
      "description": "The authoritative URI associated with this DGGRS
definition"
    },
    "dggh": {
      "type": "object",
      "description": "The hierarchical series of Discrete Global Grid upon
which this DGGRS is based, including any parameters.",
      "properties": {
        "definition": {
          "description": "The base definition of the hierarchical series of
Discrete Global Grid, which may be parameterized.",
          "type": "object",
          "required": [
            "spatialDimensions",
            "temporalDimensions"
          ],
          "properties": {
            "crs": {
              "allOf": [
                {
                  "description": "The native Coordinate Reference System (CRS)
in which the geometry of the zones for this DGG hierarchy is defined."
                },
                {
                  "anyOf": [
                    {
                      "type": "string",
                      "format": "uri",
                      "example": "EPSG:7789"
                    }
                  ]
                }
              ]
            },
            "basePolyhedron": {
              "type": "string",
```

```
              "description": "The Type/Class of Polyhedron used to construct
the Discrete Global Grid System - if it is constructued using a Base
Polyhedron.",
              "example": "icosahedron"
            },
            "refinementRatio": {
              "description": "The ratio of the area of zones between two
consecutive hierarchy level (the ratio of child zones to parent zones, also
called the aperture).",
              "example": 9,
              "type": "integer"
            },
            "constraints": {
              "type": "object",
              "properties": {
                "cellAxisAligned": {
                  "description": "Set to true if all edges of the geometry of
all zones are aligned with one of the axis of the `crs`.",
                  "type": "boolean",
                  "default": false
                },
                "cellConformal": {
                  "type": "boolean",
                  "default": false
                },
                "cellEquiAngular": {
                  "type": "boolean",
                  "default": false
                },
                "cellEquiDistant": {
                  "type": "boolean",
                  "default": false
                },
                "cellEqualSized": {
                  "description": "Set to true if the area of all zones is the
same for a particular zone geometry type of any specifc discrete global grid
of the DGG hierarchy.",
                  "type": "boolean",
                  "default": false
                }
              }
            },
            "spatialDimensions": {
              "description": "Number of Spatial Dimensions defined by the
Discrete Global Grid System.",
              "example": 2,
              "type": "integer"
            },
            "temporalDimensions": {
              "description": "Number of Temporal Dimensions defined by the
Discrete Global Grid System.",
              "example": 0,
              "type": "integer"
            },
            "zoneTypes": {
              "type": "array",
              "items": {
                "anyOf": [
                   { "type": "string" },
                   {
                      "type": "string",
                      "enum": [
                        "triangle",
```

```
                        "square",
                        "hexagon",
                        "pentagon",
                        "rhombus"
                      ]
                    }
                  ]
                },
                "example": [
                  "hexagon",
                  "pentagon"
                ]
              },
              "refinementStrategy": {
                "description": "The refinement strategy used by the Discrete
Global Grid System",
                "type": "array",
                "items": {
                  "type": "string",
                  "enum": [
                    "centredChildCell",
                    "nestedChildCell",
                    "nodeCentredChildCell",
                    "edgeCentredChildCell",
                    "faceCentredChildCell",
                    "solidCentredChildCell"
                  ]
                },
                "example": [ "nestedChildCell" ]
              }
            }
          },
          "parameters": {
            "description": "The optional parameters establishing a very specific
Discrete Global Grid System, where each zone has a well-defined geometry.",
            "type": "object",
            "properties": {
              "ellipsoid": {
                "type": "string",
                "format": "uri",
                "description": "Globe Reference System Identifier/Specification",
                "example": [
                  "EPGS:7019"
                ]
              },
              "orientation": {
                "type": "object",
                "properties": {
                  "latitude": {
                    "description": "Reference geodetic Latitude in decimal
degrees to fix the orientation of the polyhedron.",
                    "example": 58.397145907431,
                    "type": "number"
                  },
                  "longitude": {
                    "description": "Reference Longitude in decimal degrees to
fix the orientation of the polyhedron.",
                    "example": 11.20,
                    "type": "number"
                  },
                  "azimuth": {
                    "description": "Azimuth in decimal degrees of second vertex
relative to the first vertex.",
```

```
                  "example": 0.0,
                  "type": "number",
                  "default": 0.0
                },
                "description": {
                  "type": "string"
                }
              },
              "required": [
                "longitude",
                "latitude"
              ]
            }
          }
        }
      }
    },
    "zirs": {
      "description": "The Zone Identifier Reference System used for this
Discrete Global Grid System Reference System.",
      "type": "object",
      "required": [
        "textZIRS"
      ],
      "properties": {
        "textZIRS": {
          "description": "textual zone identifier indexing scheme",
          "type": "object",
          "required": [
            "description"
          ],
          "properties": {
            "description": {
              "type": "string",
              "examples": [
                "Topological Morton Code (Space Filling Curve) Identifier.
With a Level 0 alphabetical identifier from 'A' to 'S' followed by a numeric
reference from 1 to 9 for all subsequent levels representing the arrangement
of child zones following a Morton (Z-code) subdivision of the parent zone. The
length of the identifier represents the level of refinement.",
                "Universal Morton Code (Space Filling Curve) Identifier.
An ellipsoidal bitwise concatenation of longitude and latitude direct
position to form a direct Morton code. This can be used across multiple DGGS
infrastructures to provide a common index structure that is fixed in nature
regardless of the topological zone index used by each DGGS.",
                "An identifier comprised of three separate level, row and
column components corresponding to an OGC 2D Tile Matrix Set tile identifier,
where the individual component are represented by uppercase hexadecimal
numbers separated by a hyphen.",
                "An identifier comprised of an uppercase letter representing
a level (A for level 0, B for level 1...), followed by a number from 0 to 9
representing the root rhombus corresponding to the faces of an icosahedron,
in a staircase order from the top-left to the bottom right of an ISEA planar
projection after a 60 degrees clockwise rotation, followed by a hyphen, then
by an uppercase hexadecimal number identifying a sub-zone starting to count
from 0 in row-major order.",
                "An identifier comprised of an uppercase letter representing
the ISEA9R level corresponding to the current even or immediate lower odd
ISEA3H level (A for level 0, B for level 1...), followed by a number from 0 to
9 representing the root rhombus corresponding to the faces of an icosahedron,
in a staircase order from the top-left to the bottom right of an ISEA planar
projection after a 60 degrees clockwise rotation, followed by a hyphen, then
by an uppercase hexadecimal number identifying a sub-zone starting to count
```

```
from 0 in row-major order, followed by a second hyphen, then by a letter
identifying a hexagonal or pentagonal zone intersecting the sub-rhombus: A:
even level regular zone B: even level North pole zone C: even level South pole
zone D: odd level zone centered on rhombus vertex E: odd level zone top-right
of rhombus vertex F: odd level zone bottom-right of rhombus vertex G: odd
level North pole zone H: odd level South pole zone\n"
              ]
            },
            "type": {
              "anyOf": [
                {
                  "type": "string"
                },
                {
                  "type": "string",
                  "enum": [
                    "hierarchicalConcatenation",
                    "ogc2DTMSHexLevelRowCol",
                    "levelRootFaceHexRowMajorSubZone"
                  ]
                }
              ]
            }
          }
        },
        "uint64ZIRS": {
          "description": "64-bit unsigned integer zone indexing scheme",
          "type": "object",
          "required": [
            "description"
          ],
          "properties": {
            "description": {
              "type": "string",
              "examples": [
                "A 64-bit integer with the 30 least significant bits
corresponds to a column, the next 29 bits corresponding to a row, and the
5 most significant bits correspond to a level of an OGC 2D Tile Matrix Set
identifier, with individual components using little endian.",
                "A 64-bit integer with the 3 least significant bits
corresponding to a hexagonal or pentagonal zone intersecting a sub-rhombus:
0: even level regular zone 1: even level North pole zone 2: even level South
pole zone 3: odd level zone centered on rhombus vertex 4: odd level zone
top-right of rhombus vertex 5: odd level zone bottom-right of rhombus vertex
6: odd level North pole zone 7: odd level South pole zone, the next 51 bits
corresponding to a sub-rhombus index, the next 4 bits corresponding to a root
rhombus, the next 5 bits corresponding to an associated ISEA9R level, and the
last most significant bit always 0 for a valid zone identifier.\n"
              ]
            },
            "type": {
              "anyOf": [
                {
                  "type": "string"
                },
                {
                  "type": "string",
                  "enum": [
                    "ogc2DTMSHexLevelRowCol"
                  ]
                }
              ]
            }
```

```json
          }
        }
      }
    },
    "subZoneOrder": {
      "description": "The ordering used for this Discrete Global Grid System
Reference System when encoding the values associated with sub-zones at any
given depth relative to a parent zone.",
      "type": "object",
      "required": [
        "description"
      ],
      "properties": {
        "description": {
          "type": "string",
          "examples": [
            "The zones are ordered in tightly packed scanlines. For an
odd relative depth, scanlines start along a hexagon edge, whereas for even
relative depths, scanlines start on a hexagon vertex. Scanlines run in a
clockwise direction. For sub-zones at an even refinement level, zones are
ordered left to right along a scanline and scanlines are ordered from top
to bottom (considering the ISEA planar projection). For sub-zones at an
odd refinement level, zones are ordered top to bottom along a scanline and
scanlines are ordered from left to right. Sub-zones are at an even refinement
level when the parity (even/odd) of the parent zone level corresponds to the
parity of the relative depth, and at an odd refinement level when the parity
of the parent zone level and relative depth do not match. For polar pentagons,
the orientation of the scanlines varies in the ISEA planar projection but is
always clockwise, the vertex/edge of the first sub-zone is on the left side of
the planar ISEA projection for the north pole (in root rhombus 0), and on the
right side for the south pole (in root rhombus 9)."
          ]
        },
        "type": {
          "anyOf": [
            {
              "type": "string"
            },
            {
              "type": "string",
              "enum": [
                "scanline",
                "spiralFromCenter",
                "mortonCurve",
                "hilbertCurve"
              ]
            }
          ]
        }
      }
    }
  }
}
```

Listing B.1 — JSON Schema for the DGGRS Definition

The following example JSON DGGRS definitions conform to this schema.

B.2. ISEA9R DGGRS definition

The following DGGRS is for an axis-aligned and equal-area DGGH based on the Icosahedral Snyder Equal-Area (ISEA) planar projection (see PROJ and geogrid for open-source implementations) using rhombuses with a refinement ratio of 9. When the ISEA projection is rotated clockwise 60 degrees, sheared horizontally by a -60 degrees angle, and units are mapped to a 5×6 space, the zones become square and their edges are aligned with the CRS axes, while remaining equal area. A 2D Tile Matrix Set can also be defined in this transformed CRS, allowing for compatibility with OGC API — Tiles, WMTS and traditional client and visualization software with no built-in DGGS functionality. This transformed CRS distorts angles and distances more than the planar ISEA projection, but the topology, indexing, and surface geometry on the Earth model remains the same as in that planar ISEA projection.

While the ISEA projection is defined for an authalic sphere, this DGGRS assumes that when importing or exporting data referenced to the WGS84 ellipsoid a conversion between geodetic and authalic latitude is performed. This guarantees that all zones are exactly the same area on the surface of the ellipsoid.

NOTE: The authalic (meaning "area preserving") sphere is a sphere that has the same surface area as the oblate ellipsoid being used.


```
{
   "title": "ISEA9R",
   "description": "An ISEA9R Discrete Global Grid Reference System using an
ISEA orientation symmetric about the equator with a single vertex on land, and
WGS84 authalic sphere (a WGS84 geodetic latitude is mapped to the sphere by
converting it to an authalic latitude), a {level}{rootRhombus}-{hexSubZone}
indexing system, and scanline-based sub-zone ordering. The ten root rhombuses
are formed by combining two icosahedron triangles at their base. Depending
on whether the ISEA planar CRS is rotated, sheared and scaled or not, the
geometries of the zones are either squares or rhombuses, and the zone edges
are either axis-aligned, or the angles and distances are preserved. In either
case, the topology, indexing and surface geometry on the Earth model remains
the same. The ISEA9R DGGH is a dual of the ISEA3H DGGH even levels (a vertex/
node in ISEA9R is the centroid of an ISEA3H zone).",
   "uri": "https://www.opengis.net/def/dggrs/OGC/1.0/ISEA9R",
   "links": [ { "rel": "describedby", "href": "https://portal.ogc.org/files/?
artifact_id=107432" } ],
   "dggh": {
      "definition": {
         "spatialDimensions": 2,
         "temporalDimensions": 0,
         "crs": "https://www.opengis.net/def/crs/OGC/0/153456",
         "projString": "+proj=pipeline +step +proj=isea +R=6371007.18091847
+x_0=19186144.8709340879 +y_0=-3323137.7717834860 +step +proj=affine +inv
+s11=3837228.974186818 +s12=3837228.974186818 +s21=6646275.543566972 +s22=-
6646275.543566972",
         "links": [
            { "rel": "related", "href": "https://doi.org/10.3138/27H7-8K88-
4882-1752" },
            { "rel": "related", "href": "https://proj.org/en/9.3/operations/
projections/isea.html" },
         ],
         "basePolyhedron": "icosahedron",
         "refinementRatio": 9,
```

```
         "refinementStrategy": [ "nestedChildCell" ],
         "constraints": {
            "cellAxisAligned": true,
            "cellEqualSized": true
         },
         "zoneTypes": [ "square" ],
      },
      "parameters": {
         "ellipsoid": "Authalic sphere based on [EPSG:7030] (WGS84) -- NOTE:
No identifier defined yet? [ESPG:7048] exists for Authalic sphere based on
GRS80 [EPSG:7019]",
         "orientation":
         {
            "description": "ISEA orientation symmetric about the equator
with a single icosahedron vertex falling on land, with first icosahedron
vertex at authalic latitude equal to the arctangent of the golden ratio
(~58.397145907431° N geodetic latitude), 11.20°E longitude, and second vertex
due North from that first vertex.",
            "latitude": 58.397145907431,
            "longitude": 11.20,
            "azimuth": 0
         }
      }
   },
   "zirs": {
      "textZIRS": {
         "description": "An identifier comprised of an uppercase letter
representing a level (A for level 0, B for level 1...), followed by a number
from 0 to 9 representing the root rhombus corresponding to the faces of an
icosahedron, in a staircase order from the top-left to the bottom right of
an ISEA planar projection after a 60 degrees clockwise rotation, followed
by a hyphen, then by an uppercase hexadecimal number identifying a sub-zone
starting to count from 0 in row-major order.",
         "type": "levelRootFaceHexRowMajorSubZone"
      },
      "uint64ZIRS": {
         "description": "A 64-bit integer with the 30 least significant bits
corresponds to a column, the next 29 bits corresponding to a row, and the
5 most significant bits correspond to a level of an OGC 2D Tile Matrix Set
identifier, with individual components using little endian.",
         "type": "ogc2DTMSHexLevelRowCol"
      }
   },
   "subZoneOrder": {
      "description": "The zones are ordered as scanlines in row-major order,
left to right, then top to bottom, based on a CRS derived from the planar ISEA
projection after rotating it 60° clockwise (-60° rotation) and applying a -60°
degrees horizontal shear.",
      "type": "scanline"
   }
}
```

Listing B.2 — DGGRS Definition for ISEA9R

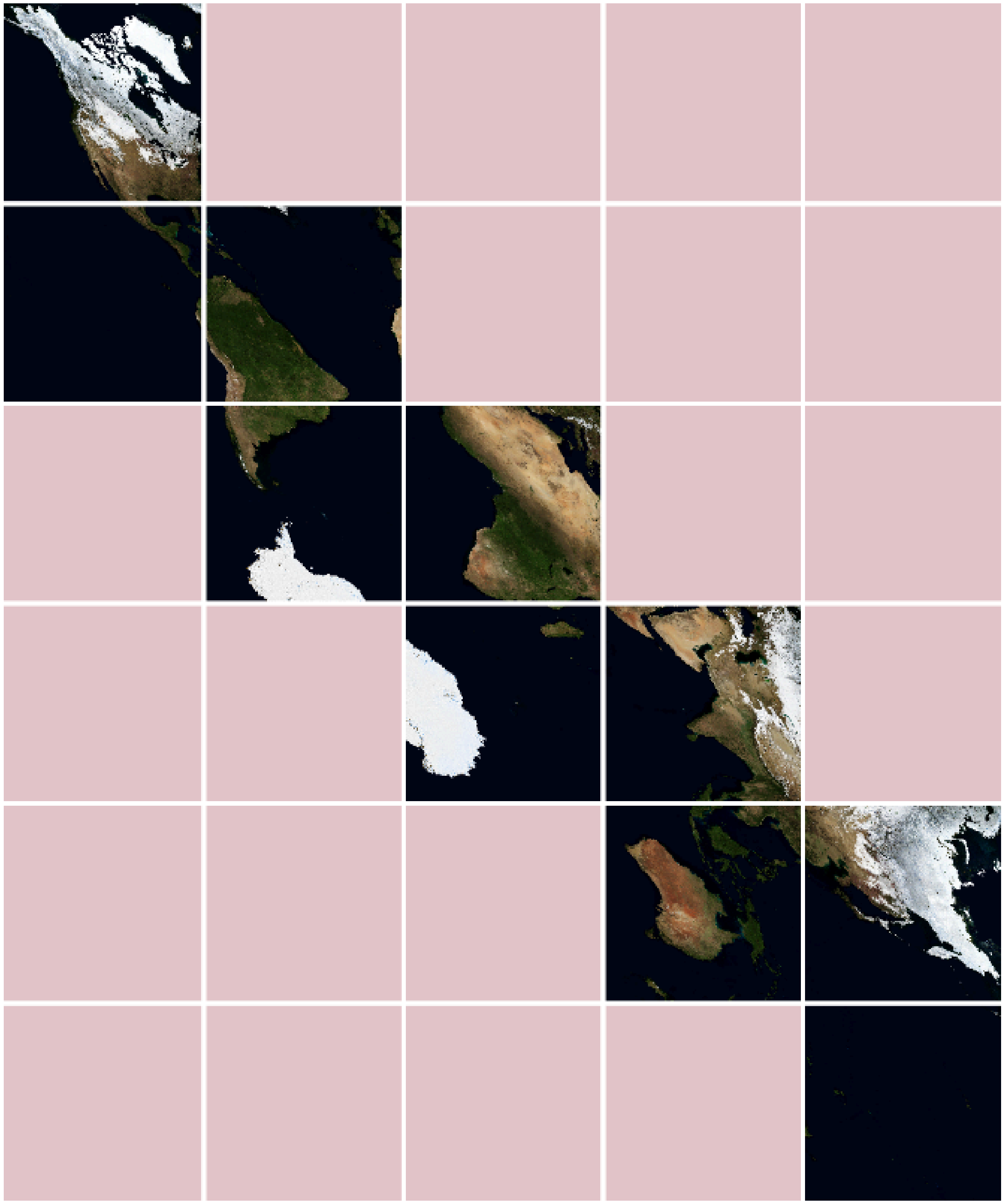


Figure B.1 — The ISEA9R squared root rhombuses (level 0) in the rotated and sheared ISEA planar CRS mapped to a 5x6 space whose axes are aligned with the zone edges, compatible with a 2DTMS definition (imagery from NASA Earth Observatory's Blue Marble: Next Generation)

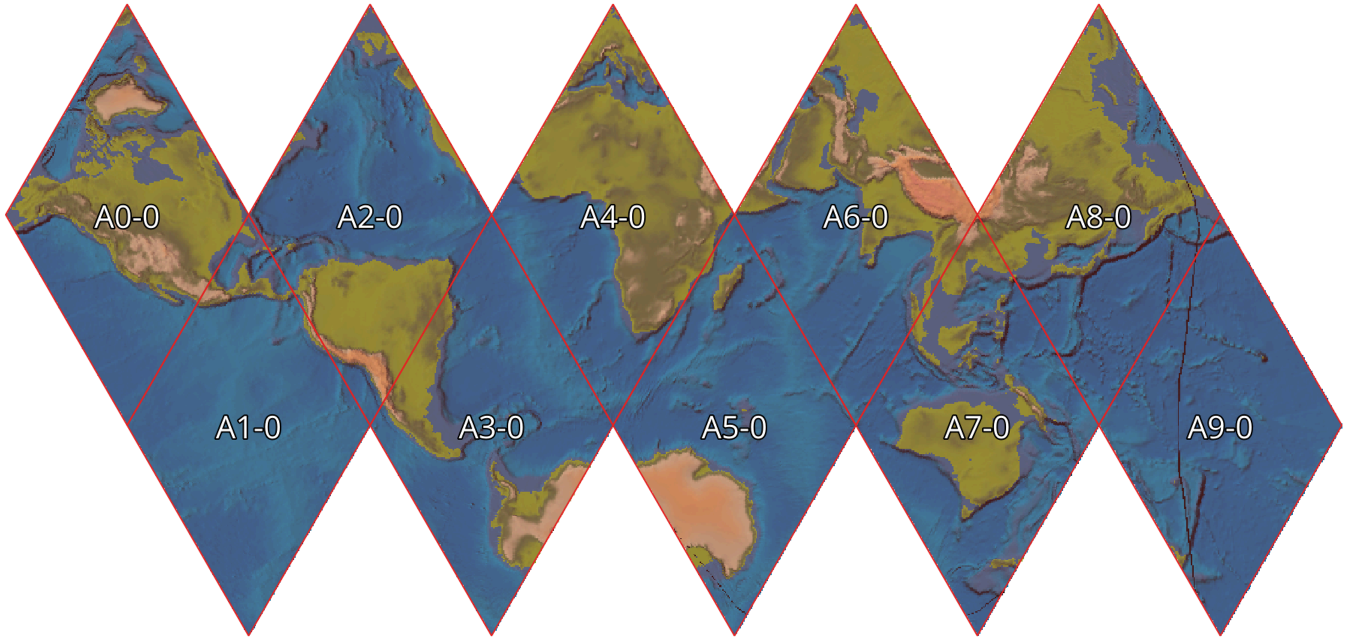


Figure B.2 — The ISEA9R level 0 zone identifiers, shown in the original ISEA planar CRS (GEBCO 2014 bathymetry)

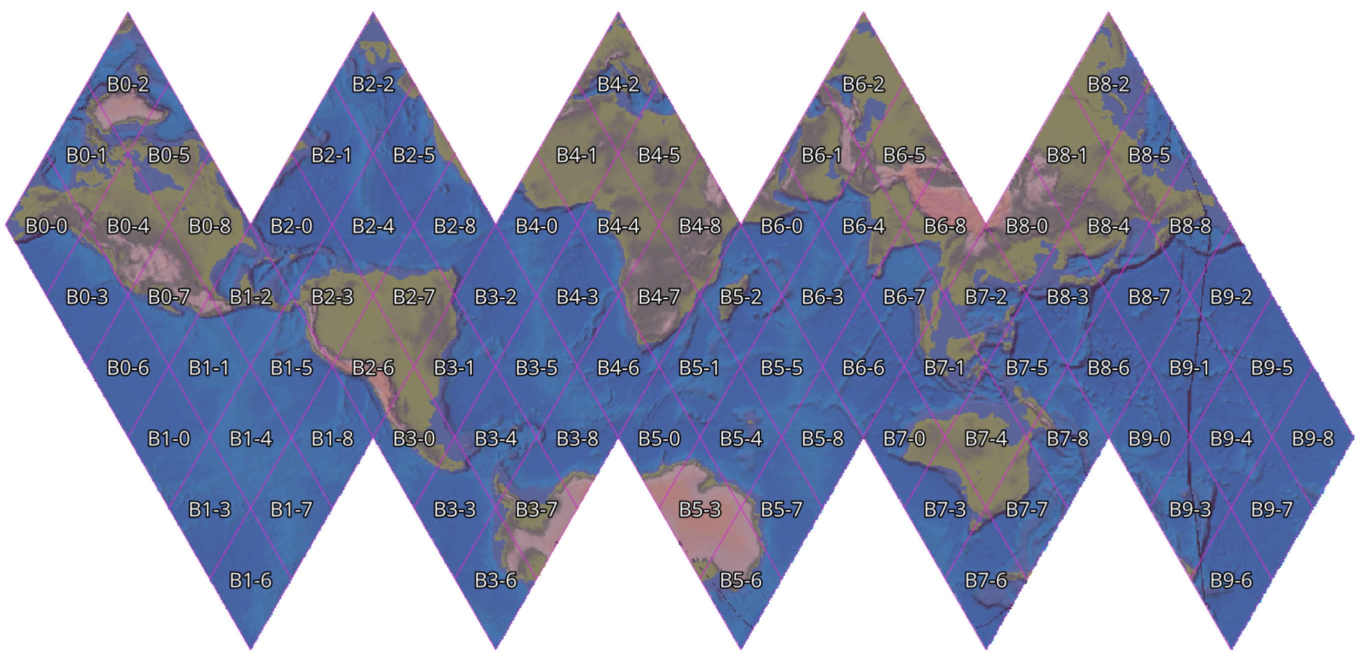


Figure B.3 — The ISEA9R level 1 zone identifiers, shown in the original ISEA planar CRS (GEBCO 2014 bathymetry)

B.3. ISEA3H DGGRS definition

The following DGGRS is for an ISEA3H DGGH, also based on the ISEA planar projection, with hexagonal and pentagonal zones, and a refinement ratio of 3, using a zone reference system based in part on its dual relationship with the ISEA9R DGGH. The DGGH is considered equal area, although each of the 12 pentagonal zones at any level occupies 5/6th of the area of a hexagonal zone at the same level.

```
{
   "title": "ISEA3H",
   "description": "An ISEA3H Discrete Global Grid Reference System using an
ISEA orientation symmetric about the equator with a single vertex on land,
and WGS84 authalic sphere (a WGS84 geodetic latitude is mapped to the sphere
by converting it to an authalic latitude), a zone indexing system based on an
ISEA9R, and a scanline-based sub-zone ordering",
   "uri": "https://www.opengis.net/def/dggrs/OGC/1.0/ISEA3H",
   "links": [ { "rel": "describedby", "href": "https://portal.ogc.org/files/?
artifact_id=107432" } ],
   "dggh": {
      "definition": {
         "spatialDimensions": 2,
         "temporalDimensions": 0,
         "crs": "https://www.opengis.net/def/crs/OGC/0/1534",
         "projString": "+proj=pipeline +step +proj=isea +R=6371007.18091847 +x_
0=19186144.8709340879 +y_0=-3323137.7717834860",
         "links": [
            { "rel": "related", "href": "https://doi.org/10.3138/27H7-8K88-
4882-1752" },
            { "rel": "related", "href": "https://proj.org/en/9.3/operations/
projections/isea.html" },
         ],
         "basePolyhedron": "icosahedron",
         "refinementRatio": 3,
         "refinementStrategy": [ "nestedChildCell", "nodeCentredChildCell" ],
         "constraints": {
            "cellEqualSized": true
         },
         "zoneTypes": [ "hexagon", "pentagon" ],
      },
      "parameters": {
         "ellipsoid": "Authalic sphere based on [EPSG:7030] (WGS84) -- NOTE:
No identifier defined yet? [ESPG:7048] exists for Authalic sphere based on
GRS80 [EPSG:7019]",
         "orientation":
         {
            "description": "ISEA orientation symmetric about the equator
with a single icosahedron vertex falling on land, with first icosahedron
vertex at authalic latitude equal to the arctangent of the golden ratio
(~58.397145907431° N geodetic latitude), 11.20°E longitude, and second vertex
due North from that first vertex.",
            "latitude": 58.397145907431,
            "longitude": 11.20,
            "azimuth": 0
         }
      }
   },
   "zirs": {
      "textZIRS": {
```

```
         "description": "An identifier comprised of an uppercase letter
representing the ISEA9R level corresponding to the current even or immediate
lower odd ISEA3H level (A for level 0, B for level 1...), followed by a number
from 0 to 9 representing the root rhombus corresponding to the faces of an
icosahedron, in a staircase order from the top-left to the bottom right of
an ISEA planar projection after a 60 degrees clockwise rotation, followed
by a hyphen, then by an uppercase hexadecimal number identifying a sub-zone
starting to count from 0 in row-major order, followed by a second hyphen,
then by a letter identifying a hexagonal or pentagonal zone intersecting
the sub-rhombus: A: even level regular zone B: even level North pole zone C:
even level South pole zone D: odd level zone centered on rhombus vertex E:
odd level zone top-right of rhombus vertex F: odd level zone bottom-right of
rhombus vertex G: odd level North pole zone H: odd level South pole zone\n",
         "type": "levelRootFaceHexRowMajorSubZone"
      },
      "uint64ZIRS": {
         "description": "A 64-bit integer with the 3 least significant bits
corresponding to a hexagonal or pentagonal zone intersecting a sub-rhombus:
0: even level regular zone 1: even level North pole zone 2: even level South
pole zone 3: odd level zone centered on rhombus vertex 4: odd level zone
top-right of rhombus vertex 5: odd level zone bottom-right of rhombus vertex
6: odd level North pole zone 7: odd level South pole zone, the next 51 bits
corresponding to a sub-rhombus index, the next 4 bits corresponding to a root
rhombus, the next 5 bits corresponding to an associated ISEA9R level, and the
last most significant bit always 0 for a valid zone identifier."
      }
   },
   "subZoneOrder": {
      "description": "The zones are ordered in tightly packed scanlines.
For an odd relative depth, scanlines start along a hexagon edge, whereas
for even relative depths, scanlines start on a hexagon vertex. Scanlines run
in a clockwise direction. For sub-zones at an even refinement level, zones
are ordered left to right along a scanline and scanlines are ordered from
top to bottom (considering the ISEA planar projection). For sub-zones at an
odd refinement level, zones are ordered top to bottom along a scanline and
scanlines are ordered from left to right. Sub-zones are at an even refinement
level when the parity (even/odd) of the parent zone level corresponds to the
parity of the relative depth, and at an odd refinement level when the parity
of the parent zone level and relative depth do not match. For polar pentagons,
the orientation of the scanlines varies in the ISEA planar projection but is
always clockwise, the vertex/edge of the first sub-zone is on the left side of
the planar ISEA projection for the north pole (in root rhombus 0), and on the
right side for the south pole (in root rhombus 9).",
      "type": "scanline"
   }
}
```

Listing B.3 — DGGRS Definition for ISEA3H using indexing based on ISEA9R

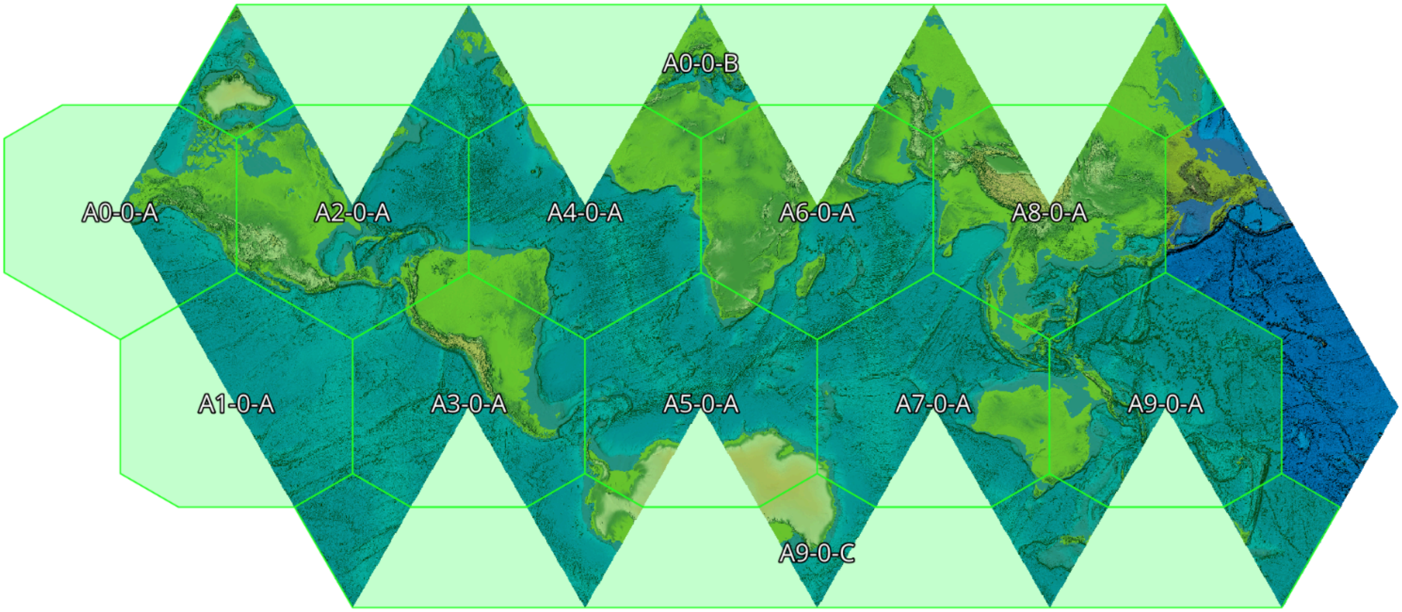


Figure B.4 — The ISEA3H level 0 zones and associated identifiers (derived from ISEA9R level 0), shown on an ISEA planar projection (GEBCO 2014 bathymetry)

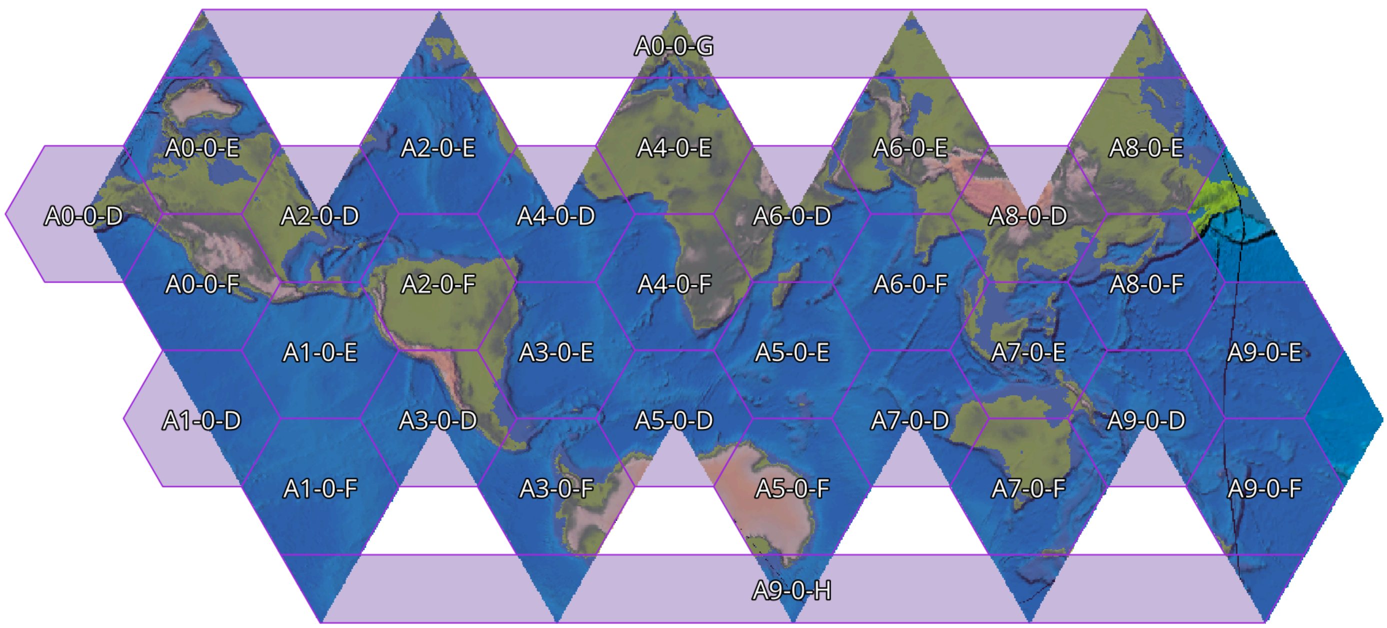


Figure B.5 — The ISEA3H level 1 zones and associated identifiers (derived from ISEA9R level 0), shown on an ISEA planar projection (GEBCO 2014 bathymetry)

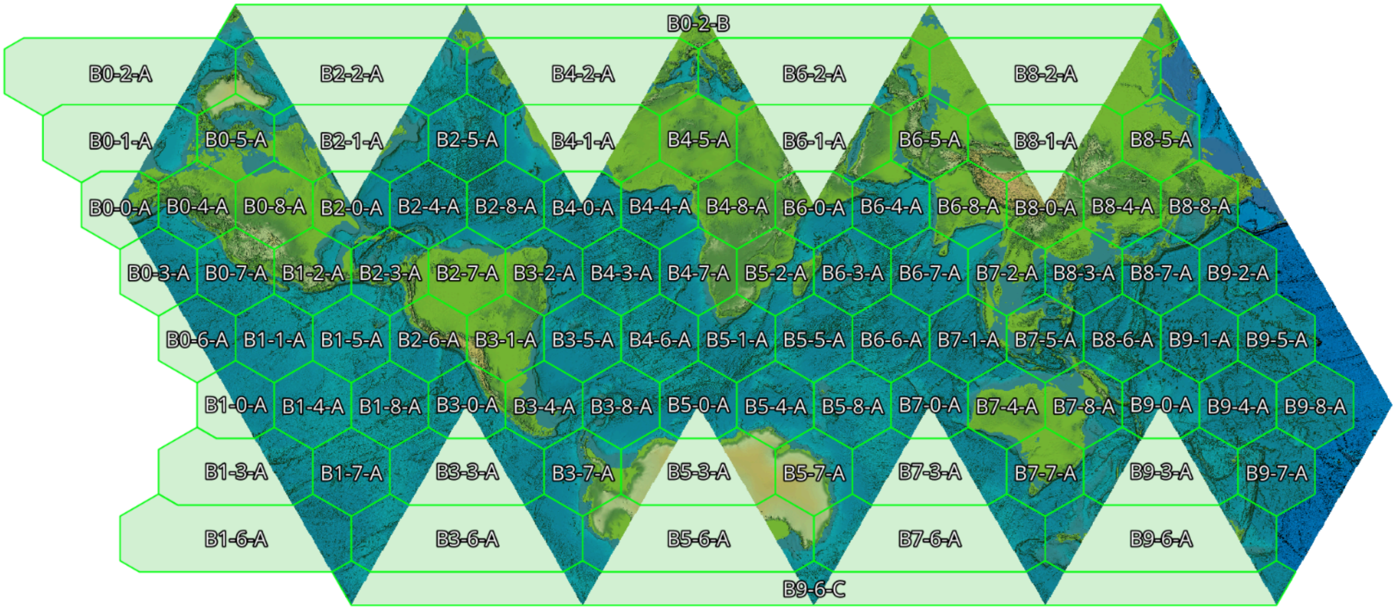


Figure B.6 — The ISEA3H level 2 zones and associated identifiers (derived from ISEA9R level 1), shown on an ISEA planar projection (GEBCO 2014 bathymetry)

The sub-zone ordering for this DGGRS is based on scanlines of tightly packed hexagons, sharing whole edges with the previous and next zone within a scanline, while each zone shares two edges with the previous scanline, and another two edges with the next scanline. The scanlines start on a vertex of the parent zone for an even relative depth, while they start on an edge for an odd relative depth.

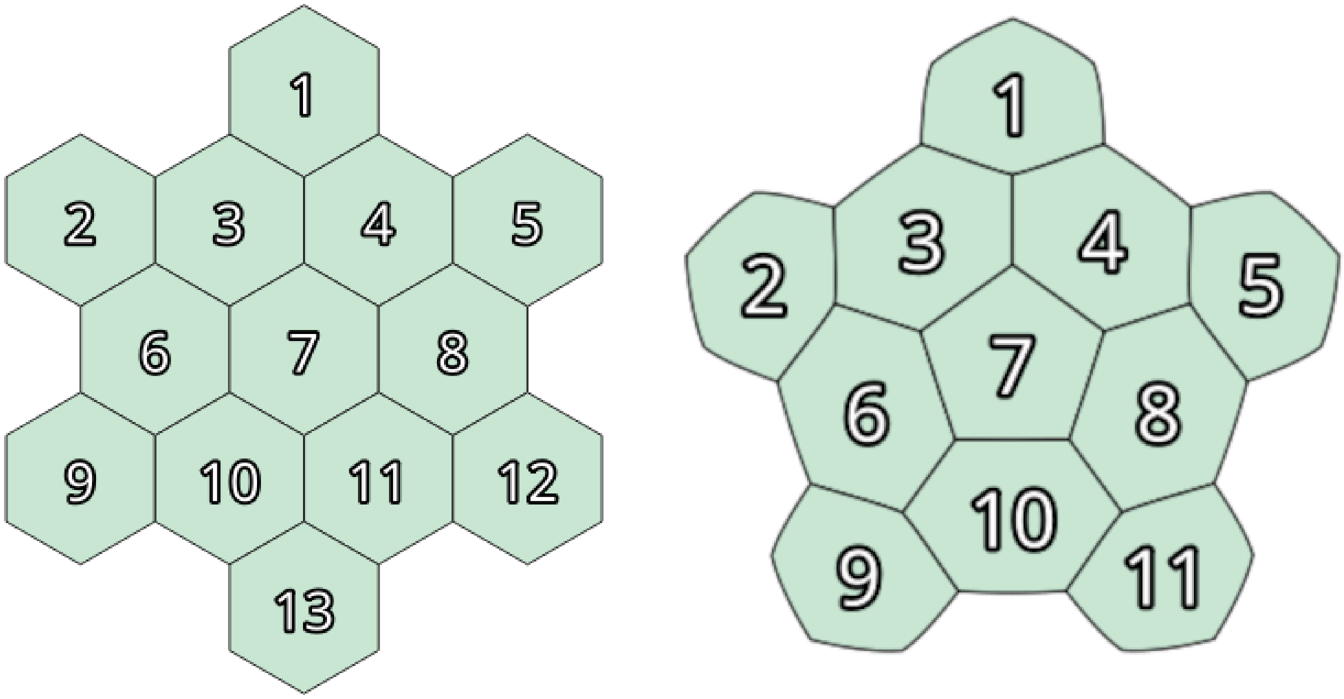


Figure B.7 — Sub-zone order for even relative depth starting on a vertex (hexagonal and pentagonal parents, depth 2)

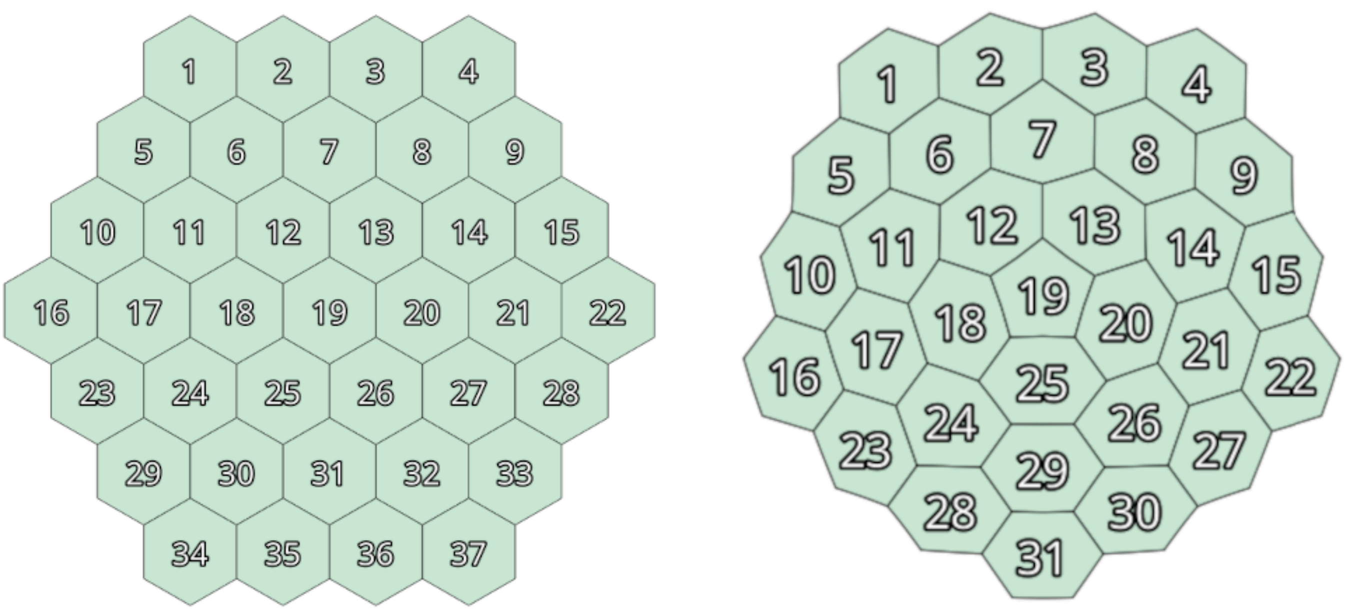


Figure B.8 — Sub-zone order for odd relative depth starting on an edge (hexagonal and pentagonal parents, depth 3)

B.4. GNOSIS Global Grid DGGRS definition

The following DGGRS is based on the variable width 2D Tile Matrix Set of the same name. The GNOSIS Global Grid is also axis-aligned, with the axes being the geographic latitude and longitude, and already corresponds to a registered 2DTMS. Through the variable width capability of the tile matrices, this DGGH approximates equal area. For example, at any refinement level, only 4 zones touch each pole. Since the error budget still exceeds the 1%, it cannot be considered equal area per the OGC Topic 21 DGGS Equal Area Earth Reference System Package. However, this divergence from the mean zone area is the same along a particular row of zones following latitude lines, and can easily be calculated and taken into consideration. Since its native axes are latitude and longitude, no additional reprojection is required to import data to or export data from this DGGH.


```json
{
   "title": "GNOSISGlobalGrid",
   "description": "A Discrete Global Grid Reference System based on the GNOSIS
Global Grid 2D Tile Matrix Set, a {hexLevel}-{hexRow}-{hexCol} indexing
system, and scanline-based sub-zone ordering. Making use of variable matrix
widths, zones are divided by 4, unless they touch the pole, in which case the
half touching the pole is not split longitude-wise. The error budget is ~-25.8/
+66.0% from mean zone size (+/- 45.9% from median) up to level 25 (~849 cm²
mean zones).",
   "uri": "https://www.opengis.net/def/dggrs/OGC/1.0/GNOSISGlobalGrid",
   "links": [ { "rel": "describedby", "href": "https://docs.ogc.org/is/17-
083r4/17-083r4.html#toc58" } ],
   "dggh": {
      "definition": {
         "spatialDimensions": 2,
         "temporalDimensions": 0,
         "crs": "https://www.opengis.net/def/crs/EPSG/0/4326",
         "links": [
            { "rel": "related", "href": "https://www.opengis.net/def/
tilematrixset/OGC/1.0/GNOSISGlobalGrid" }
         ],
         "basePolyhedron": "octahedron",
         "refinementRatio": 4,
         "refinementStrategy": [ "nestedChildCell" ],
         "constraints": {
            "cellAxisAligned": true
         },
         "zoneTypes": [ "rectangle" ],
      },
      "parameters": {
         "ellipsoid": "[EPSG:7030]"
         "orientation":
         {
            "latitude": 0,
            "longitude": 0
         }
      }
   },
   "zirs": {
      "textZIRS": {
         "description": "An identifier comprised of three uppercase
hexadecimal numbers separated by hyphens representing the level, row and
column of the GNOSIS Global Grid 2D Tile Matrix Set.",
         "type": "ogc2DTMSHexLevelRowCol"
```

```
        },
        "uint64ZIRS": {
           "description": "A 64-bit integer with the 30 least significant bits
corresponds to a column, the next 29 bits corresponding to a row, and the
5 most significant bits correspond to a level of an OGC 2D Tile Matrix Set
identifier, with individual components using little endian.",
           "type": "ogc2DTMSHexLevelRowCol"
        }
     },
     "subZoneOrder": {
        "description": "The zones are ordered as scanlines in row-major order,
left to right, then top to bottom, based on the EPSG:4326 CRS.",
        "type": "scanline"
     }
}
```

Listing B.4 — DGGRS Definition for the GNOSIS Global Grid

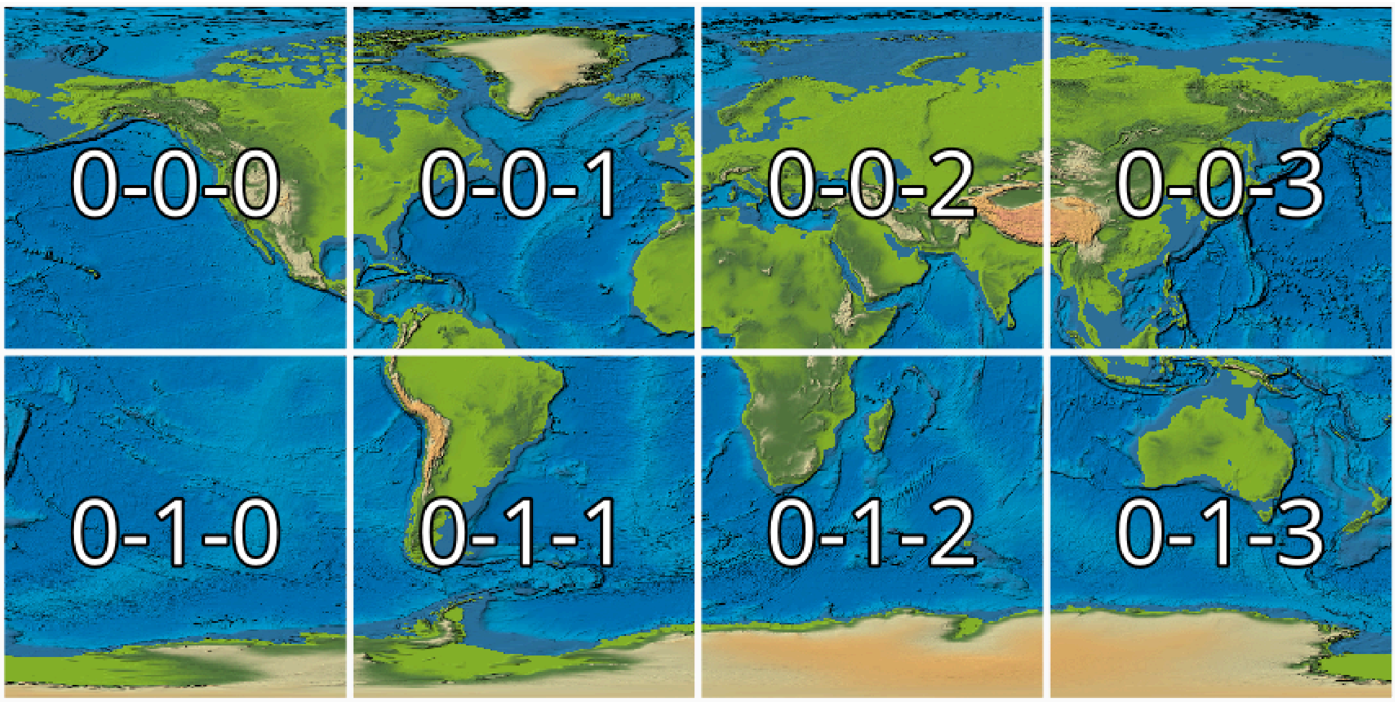


Figure B.9 — The GNOSIS Global Grid level 0 zones and associated identifiers (GEBCO 2014 bathymetry)

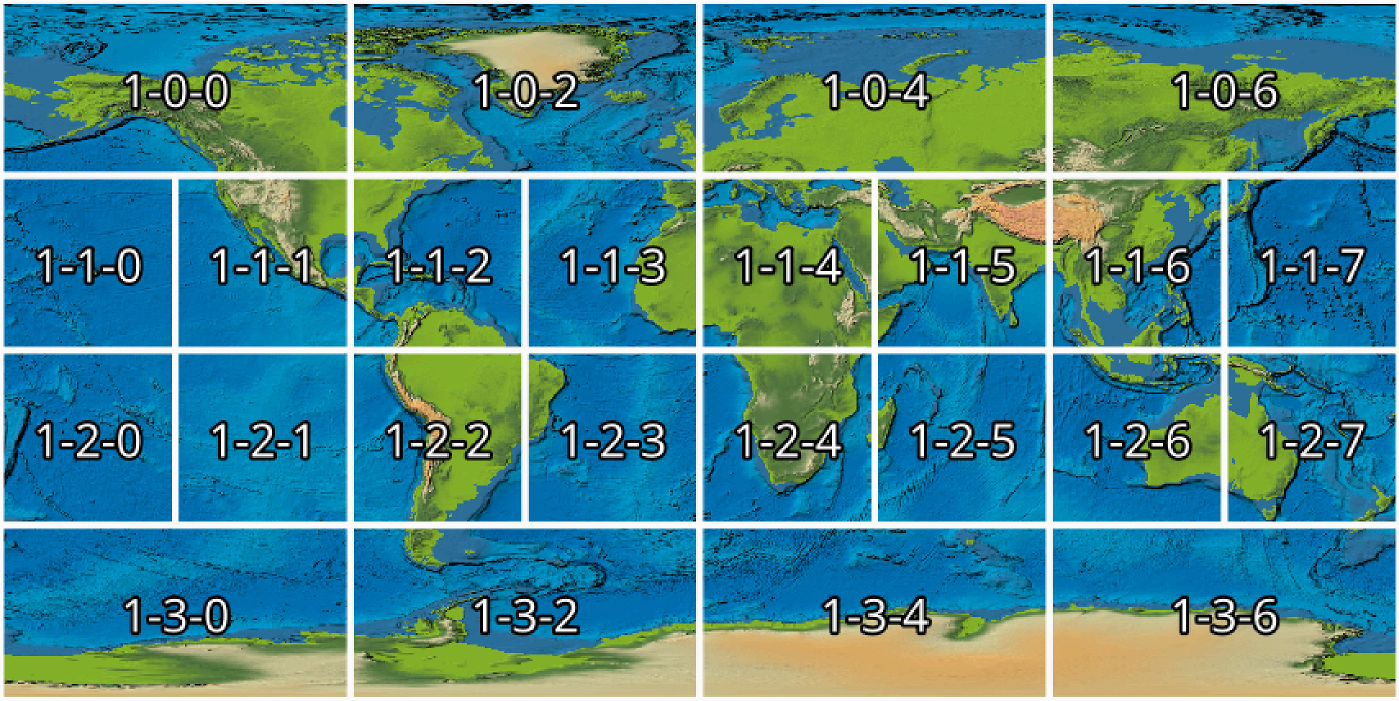


Figure B.10 — The GNOSIS Global Grid level 1 zones and associated identifiers (GEBCO 2014 bathymetry)

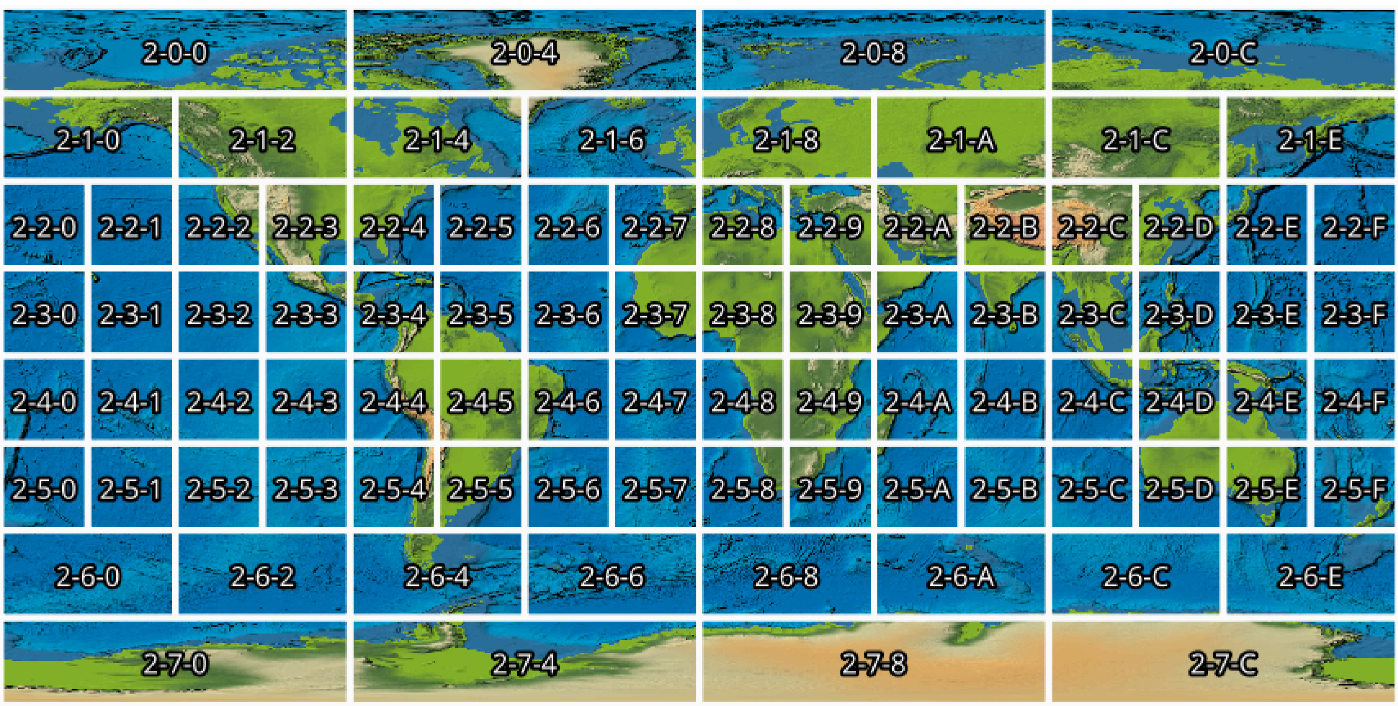


Figure B.11 — The GNOSIS Global Grid level 2 zones and associated identifiers (GEBCO 2014 bathymetry)

C

ANNEX C (INFORMATIVE) EXAMPLES

ANNEX C (INFORMATIVE) EXAMPLES

This annex provides a set of examples illustrating requests and responses for retrieving DGGS zone data and performing zone queries using capabilities defined by this DGGS API.

C.1. Collection description

DGGS resources can be attached to an *OGC API — Common — Part 2* collection origin resource path. The following global GEBCO 2014 bathymetry collection will be used as an example:

https://maps.gnosis.earth/ogcapi/collections/gebco

An example JSON response for this collection description resource, as defined by Common — Part 2 and the Collection DGGS Requirements Class, is presented below, including a link to the list of available DGGRSs:


```
{
   "id" : "gebco",
   "title" : "GEBCO 2014",
   "description" : "General Bathymetric Chart of the Oceans (2014)",
   "attribution" : "General Bathymetric Chart of the Oceans (<a href='https://
www.gebco.net/news_and_media/gebco_2014_grid.html'>GEBCO_2014</a>)",
   "extent" : {
      "spatial" : {
         "bbox" : [ [ -180, -90, 180, 90 ] ],
         "grid" : [
            { "firstCoordinate": -180, "relativeBounds": [ 0, 0 ], "cellsCount"
 : 65537, "resolution" : 0.0054931640625 },
            { "firstCoordinate":  -90, "relativeBounds": [ 0, 0 ], "cellsCount"
 : 32769, "resolution" : 0.0054931640625 }
         ]
   } },
   "minScaleDenominator" : 2183915.0938621787354,
   "minCellSize" : 0.0054931640625,
   "links" : [
      { "rel" : "self", "title" : "Information about this GEBCO data
collection",
        "href" : "/ogcapi/collections/gebco"
      },
      { "rel" : "[ogc-rel:schema]",  "title" : "Schema",
         "href" : "/ogcapi/collections/gebco/schema"
      },
      { "rel" : "[ogc-rel:queryables]",  "title" : "Queryables",
         "href" : "/ogcapi/collections/gebco/queryables"
```

```
      },
      { "rel" : "[ogc-rel:dggrs-list]", "title" : "Available DGGRSs for this
GEBCO data collection",
        "href" : "/ogcapi/collections/gebco/dggs"
      }
   ]
}
```

Listing

C.2. Listing available DGGRSs

Whether accessing DGGS resources for the root of the API (which may correspond to a dataset, as stated in OGC API — Features) or for a particular collection, the first DGGS-specific request made by a client would be to list the available DGGRSs. Following the link obtained from the previous example response, a client can determine which DGGRSs are supported by the API at that endpoint:

https://maps.gnosis.earth/ogcapi/collections/gebco/dggs

An example JSON response for this DGGRS list, as defined by the Core requirements class, is presented below:

```
{
   "links" : [
      { "rel" : "[ogc-rel:geodata]", "href" : "/ogcapi/collections/gebco" }
   ],
   "dggrs" : [
      {
         "id" : "GNOSISGlobalGrid",
         "title" : "GNOSIS Global Grid",
         "uri" : "[ogc-dggrs:GNOSISGlobalGrid]",
         "links" : [
            { "rel" : "self", "title" : "GNOSISGlobalGrid DGGRS for GEBCO",
              "href" : "/ogcapi/collections/gebco/dggs/GNOSISGlobalGrid" },
            { "rel" : "[ogc-rel:dggrs-definition]", "title" :
"GNOSISGlobalGrid DGGRS definition",
              "href" : "/ogcapi/dggrs/GNOSISGlobalGrid"
            }
         ]
      },
      {
         "id" : "ISEA9R",
         "title" : "ISEA Aperture 9 Rhombic",
         "uri" : "[ogc-dggrs:ISEA9R]",
         "links" : [
            { "rel" : "self", "title" : "ISEA9R DGGRS for GEBCO",
              "href" : "/ogcapi/collections/gebco/dggs/ISEA9R"
            },
            { "rel" : "[ogc-rel:dggrs-definition]", "title" : "ISEA9R DGGRS
definition",
              "href" : "/ogcapi/dggrs/ISEA9R"
            }
         ]
      },
      {
```

```
         "id" : "ISEA3H",
         "title" : "ISEA Aperture 3 Hexagonal",
         "uri" : "[ogc-dggrs:ISEA3H]"
         "links" : [
            { "rel" : "self", "title" : "ISEA3H DGGRS for GEBCO",
              "href" : "/ogcapi/collections/gebco/dggs/ISEA3H"
            },
            { "rel" : "[ogc-rel:dggrs-definition]", "title" : "ISEA3H DGGRS
definition",
              "href" : "/ogcapi/dggrs/ISEA3H"
            }
         ]
      }
   ]
}
```

Listing

C.3. Retrieving the description of a specific DGGRS

A client would then select a particular DGGRS of which it already has built-in knowledge, possibly using a software library implementing support for one or more DGGRS available from the server.

In this example, the client selects the ISEA3H DGGRS (because it has built-in support for it) and follows the link to obtain a complete description of the DGGRS:

https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA3H

An example JSON response for this DGGRS description, as defined by the Core requirements class, is presented below. Note how the description is more comprehensive than the summary description included for each item in the previous DGGRS list response, including a description, a default (relative) depth as well as additional links and link templates.

```
{
   "id" : "ISEA3H",
   "title" : "ISEA Aperture 3 Hexagonal",
   "description" : "A Discrete Global Grid Reference System based on the
Icosahedral Snyder Equal Area projection, with aperture 3 hexagonal zones,
using an indexing scheme based on ISEA9R.",
   "uri" : "[ogc-dggrs:ISEA3H]",
   "defaultDepth": 10,
   "maxRefinementLevel": 18,
   "maxRelativeDepth": 13,
   "links" : [
      { "rel" : "self", "title" : "ISEA3H DGGRS for GEBCO",
        "href" : "/ogcapi/collections/gebco/dggs/ISEA3H"
      },
      { "rel" : "[ogc-rel:dggrs-definition]", "title" : "ISEA3H DGGRS
definition",
        "href" : "/ogcapi/dggrs/ISEA3H"
      },
      { "rel" : "[ogc-rel:dggrs-zone-query]", "title" : "ISEA3H DGGRS Zone
Query",
        "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones"
      },
```

```
      { "rel" : "[ogc-rel:geodata]", "href" : "/ogcapi/collections/gebco" }
   ],
   "linkTemplates" : [
      {
         "rel" : "[ogc-rel:dggrs-zone-info]",
         "title" : "DGGRS zone information for a particular ISEA3H zone",
         "uriTemplate" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/{zoneId}"
      },
      {
         "rel" : "[ogc-rel:dggrs-zone-data]",
         "title" : "Data retrieval for a particular ISEA3H zone",
         "uriTemplate" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/{zoneId}/
data"
      }
   ]
}
```

Listing

ISEA3H is an equal area discrete global grid hierarchy based on the Icosahedral Snyder Equal Area projection, using hexagonal grids with a refinement ratio (aperture) of 3. All hexagonal zones of a given refinement level have exactly the same area, while each of the twelve (12) pentagonal zones occupies 5/6th the area of a hexagonal zone of the same level.

See the informative annex B for examples of DGGRS definition responses which the client could obtain by following the `[ogc-rel:dggrs-definition]` links. The annex includes provisionary definitions for the ISEA3H, ISEA9R and GNOSIS Global Grid used in these examples which will likely be used to initially populate an authoritative registry, Additional DGGRS entries would be submitted, reviewed and registered based on the needs of the community.

C.4. Retrieving information for a specific DGGRS zone

Later examples will demonstrate how to list available zones by performing zone queries, but for now these examples will skip directly to retrieving information about a particular zone. This capability is also defined in the Core requirements class and exists primarily for completeness of the API resource tree, making the `../dggs/zones/{zoneId}` resource between `../dggs/zones` and `../dggs/zones/{zoneId}/data` also retrievable. The zone information is useful for enabling exploration of datasets using a browser interface, as well as for educational, demonstration and debugging purposes. An actual DGGS API client for visualization or performing analytics would normally not access this resource at all, since URL templates allow to directly access the data for a particular zone. While the Core requirements class makes several recommendations about statistical information and topological relationships that should be included in the zone information resource, clients may not rely on any of this information being present. Clients can instead rely on their own local software libraries to establish topological relationships, can use zone queries to pose questions that can be answered by a list of zone IDs, or can retrieve zone data packets containing values from multiple sub-zones to perform their own analytics. A future extension could also introduce mandatory requirements making some of the recommended content such as summary statistics or additional useful information mandatory, which clients could then rely on for conforming implementations.

This example assumes a client does for some reason request information in a JSON representation about a particular ISEA3H DGGRS zone, say `E6-317-A`, based on its built-in support for that DGGRS. Perhaps this client is an Executable Test Suite — a set of code providing runtime tests for the assertions defined by the Abstract Test Suite. Implementing this resource is still required to conform to the Core conformance class, including at minimum an `id` and links to the DGGRS and data retrieval, if applicable. The GET request URL and response for this zone information resource associated with the GEBCO collection would then look like the following:

https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-317-A

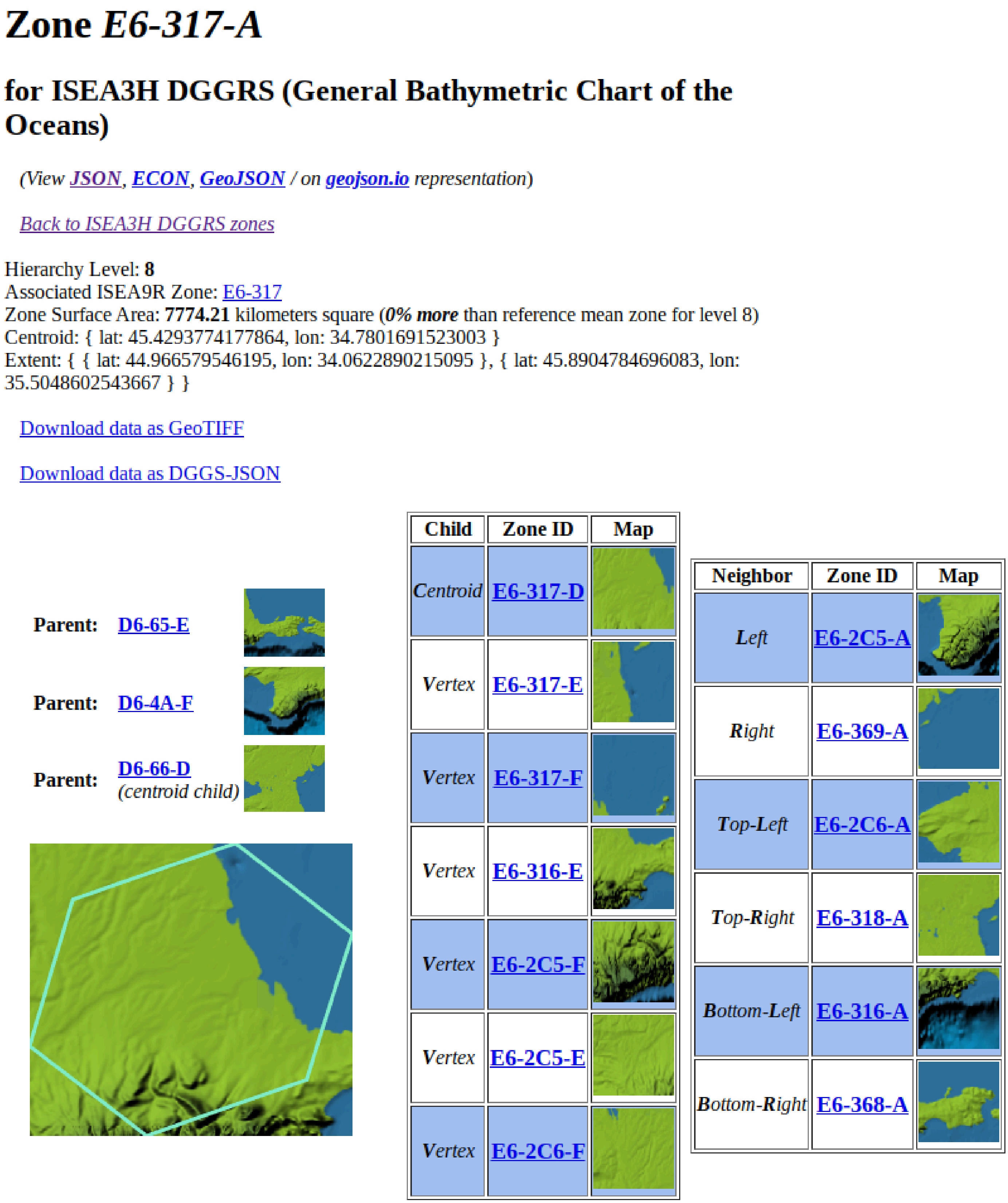

```
{
   "id" : "E6-317-A",
   "links" : [
      { "rel" : "[ogc-rel:geodata]", "href" : "/ogcapi/collections/gebco" },
      { "rel" : "[ogc-rel:dggrs]", "title" : "ISEA3H DGGRS for GEBCO",
        "href" : "/ogcapi/collections/gebco/dggs/ISEA3H"
      },
      { "rel" : "[ogc-rel:dggrs-definition]", "title" : "ISEA3H DGGRS
definition",
        "href" : "/ogcapi/dggrs/ISEA3H"
      },
      { "rel" : "[ogc-rel:dggrs-zone-data]", "title" : "ISEA3H Zone E6-317-A
data for GEBCO",
        "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-317-A/data"
      },

      { "rel" : "[ogc-rel:dggrs-zone-parent]", "title": "Parent zone D6-65-E",
        "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/D6-65-E" },
      { "rel" : "[ogc-rel:dggrs-zone-parent]", "title": "Parent zone D6-4A-F",
        "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/D6-4A-F" },
      { "rel" : "[ogc-rel:dggrs-zone-parent]", "title": "Parent zone D6-66-D
(centroid child)",
        "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/D6-66-D" },

      { "rel" : "[ogc-rel:dggrs-zone-child]", "title": "Child zone E6-317-D
(centroid child)",
        "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-317-D" },
      { "rel" : "[ogc-rel:dggrs-zone-child]", "title": "Child zone E6-317-E
(vertex child)",
        "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-317-E" },
      { "rel" : "[ogc-rel:dggrs-zone-child]", "title": "Child zone E6-317-F
(vertex child)",
        "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-317-F" },
      { "rel" : "[ogc-rel:dggrs-zone-child]", "title": "Child zone E6-316-E
(vertex child)",
        "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-316-E" },
      { "rel" : "[ogc-rel:dggrs-zone-child]", "title": "Child zone E6-2C5-F
(vertex child)",
        "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-2C5-F" },
      { "rel" : "[ogc-rel:dggrs-zone-child]", "title": "Child zone E6-2C5-E
(vertex child)",
        "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-2C5-E" },
      { "rel" : "[ogc-rel:dggrs-zone-child]", "title": "Child zone E6-2C6-F
(vertex child)",
        "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-2C6-F" },

      { "rel" : "[ogc-rel:dggrs-zone-neighbor]", "title": "Neighboring zone E6-
2C5-A (left)",
        "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-2C5-A" },
      { "rel" : "[ogc-rel:dggrs-zone-neighbor]", "title": "Neighboring zone E6-
369-A (right)",
```

```
        "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-369-A" },
      { "rel" : "[ogc-rel:dggrs-zone-neighbor]", "title": "Neighboring zone E6-
2C6-A (top-left)",
        "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-2C6-A" },
      { "rel" : "[ogc-rel:dggrs-zone-neighbor]", "title": "Neighboring zone E6-
318-A (top-right)",
        "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-318-A" },
      { "rel" : "[ogc-rel:dggrs-zone-neighbor]", "title": "Neighboring zone E6-
316-A (bottom-left)",
        "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-316-A" },
      { "rel" : "[ogc-rel:dggrs-zone-neighbor]", "title": "Neighboring zone E6-
368-A (bottom-right)",
        "href" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/E6-368-A" }
   ],
   "shapeType" : "hexagon",
   "areaMetersSquare" : 7774205482.76313,
   "level": 8,
   "crs": "[OGC:CRS84]",
   "centroid": [ 34.7801691523003, 45.4293774177864 ],
   "bbox": [ 34.0622890215095, 44.966579546195, 35.5048602543667,
45.8904784696083 ],
   "statistics" : {
      "Elevation" : {
         "minimum": -56.415161132812,
         "maximum": 499.67742919922,
         "average": 34.243211554643,
         "stdDev": 58.802922870029
      }
   },
   "geometry" : {
      "type" : "Polygon",
      "coordinates" : [
         [ [ 34.9825792437548, 45.8904784696083 ], [ 34.8364790349283,
45.8553425997858 ],
           [ 34.7198070743586, 45.8270885080255 ], [ 34.6033186032223,
45.798706447509  ],
           [ 34.4870121932418, 45.7701975609193 ], [ 34.3708864462471,
45.7415629623275 ],
           [ 34.2549399933378, 45.7128037381496 ], [ 34.2158619560033,
45.6203085346353 ],
           [ 34.1847997584098, 45.5462655152704 ], [ 34.1539133527403,
45.4721817939786 ],
           [ 34.1232007406641, 45.3980582297526 ], [ 34.0926599474139,
45.3238956664313 ],
           [ 34.0622890215095, 45.2496949329617 ], [ 34.1882009344326,
45.181974647101  ],
           [ 34.2824290413424, 45.131089867063  ], [ 34.3660396924306,
45.085791129104  ],
           [ 34.4495117238456, 45.040428739687  ], [ 34.5848590190371,
44.966579546195  ],
           [ 34.6988274980307, 44.9951957400744 ], [ 34.8129599968164,
45.023694616047  ],
           [ 34.9272577135866, 45.0520752443535 ], [ 35.0417218725293,
45.0803366718494 ],
           [ 35.156353724587,  45.1084779211899 ], [ 35.2998813084759,
45.1434839527948 ],
           [ 35.3321842414696, 45.2173283845563 ], [ 35.364672186874,
45.2911278096443 ],
           [ 35.3973472857246, 45.3648812441139 ], [ 35.4302117049963,
45.4385876857853 ],
           [ 35.4632676379116, 45.5122461139102 ], [ 35.5048602543667,
45.6042500443317 ],
```

```
            [ 35.4216718871623, 45.6502103665179 ], [ 35.307054399127,
45.7133043112688 ],
            [ 35.2235259331723, 45.7591167641879 ], [ 35.0665221514834,
45.8448463562633 ],
            [ 34.9825792437548, 45.8904784696083 ] ]
       ]
   }
}
```


Listing

An HTML representation of that zone information resource, allowing a user to browse through parent, children and neighboring zones, could look like the following:



Zone *E6-317-A*

for ISEA3H DGGRS (General Bathymetric Chart of the Oceans)

*(View **JSON**, **ECON**, **GeoJSON** / on **geojson.io** representation)*

Back to ISEA3H DGGRS zones

Hierarchy Level: **8**
Associated ISEA9R Zone: E6-317
Zone Surface Area: **7774.21** kilometers square (***0% more*** than reference mean zone for level 8)
Centroid: { lat: 45.4293774177864, lon: 34.7801691523003 }
Extent: { { lat: 44.966579546195, lon: 34.0622890215095 }, { lat: 45.8904784696083, lon: 35.5048602543667 } }

Download data as GeoTIFF

Download data as DGGS-JSON

Parent:	**D6-65-E**	
Parent:	**D6-4A-F**	
Parent:	**D6-66-D** *(centroid child)*	

Child	Zone ID	Map
Centroid	**E6-317-D**	
Vertex	**E6-317-E**	
Vertex	**E6-317-F**	
Vertex	**E6-316-E**	
Vertex	**E6-2C5-F**	
Vertex	**E6-2C5-E**	
Vertex	**E6-2C6-F**	

Neighbor	Zone ID	Map
Left	**E6-2C5-A**	
Right	**E6-369-A**	
Top-Left	**E6-2C6-A**	
Top-Right	**E6-318-A**	
Bottom-Left	**E6-316-A**	
Bottom-Right	**E6-368-A**	

Figure C.1 — Sample HTML response of an ISEA3H DGGRS zone information resource for GEBCO data

C.5. Simple zone queries

Having selected a particular DGGRS, one use case for the API is for a client to perform spatial queries by requesting a list of zones. The simplest case is to simply request a list of all available zones for a discrete global grid of a given refinement level (`zone-level`), which in the case of a particular collection implies zones for which the collection has data.

C.5.1. Querying a particular refinement level

If no `zone-level` query parameter is provided, the server is free to pick a default zone level for such queries. This would result in a reasonable number of zones returned, based on the area and resolution of the data collection. In this example, the client will explicitly request a list of zones at refinement level 1.

By default, if `compact-zones=false` is not specified by the client, zone queries are compacted. This means that if all sub-zones of a parent zone would be included in the response, they get replaced by that parent zone so that fewer zones need to be enumerated. This can result in zones being returned of a lower refinement level than the requested `zone-level`, and therefore a mix of zones of different levels. A full globe response would therefore always result in the list of all zones of refinement level 0. In these examples, the client will explicitly set `compact-zones=false`.

Following the `[ogc-rel:dggrs-zone-query]` link, specifying these query parameters, and negotiating a JSON response response using an `Accept: application/json` header, the client requests from the endpoint:

https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA3H/zones?zone-level=1&compact-zones=false

An example JSON response for this Zone Query JSON, as defined by the Zone Query requirements class, is presented below:


```
{
   "zones" : [
      "A0-0-D", "A0-0-E", "A0-0-F", "A0-0-G", "A1-0-D", "A1-0-E",
      "A1-0-F", "A2-0-D", "A2-0-E", "A2-0-F", "A3-0-D", "A3-0-E",
      "A3-0-F", "A4-0-D", "A4-0-E", "A4-0-F", "A5-0-D", "A5-0-E",
      "A5-0-F", "A6-0-D", "A6-0-E", "A6-0-F", "A7-0-D", "A7-0-E",
      "A7-0-F", "A8-0-D", "A8-0-E", "A8-0-F", "A9-0-D", "A9-0-E",
      "A9-0-F", "A9-0-H"
   ],
   "returnedAreaMetersSquare" : 510065621724088.5,
   "links" : [
      { "rel" : "[ogc-rel:dggrs]", "title" : "ISEA3H DGGS for GEBCO",
        "href" : "/ogcapi/collections/gebco/dggs/ISEA3H"
      },
      { "rel" : "[ogc-rel:dggrs-definition]", "title" : "ISEA3H DGGRS
definition",
         "href" : "/ogcapi/dggrs/ISEA3H"
      },
      { "rel" : "[ogc-rel:geodata]", "href" : "/ogcapi/collections/gebco" }
   ],
```

```
    "linkTemplates" : [
       { "rel" : "[ogc-rel:dggrs-zone-data]", "title" : "ISEA3H data for GEBCO",
         "uriTemplate" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/{zoneId}/
data",
       }
    ]
}
```

Listing

The server returned all 32 zones of ISEA3H level 1 (which corresponds to a truncated icosahedron — the traditional soccer ball pattern), because the collection has global coverage.

A visualization of the response from negotiating `application/geo+json` for the above request is seen below.

> *Most of the visualizations in this annex were produced using a regular default installation of QGIS, without any special DGGS plugin. Although the use of explicit zone geometry in responses is not ideal for efficient use of this API, this demonstrates the possibility of achieving interoperability with traditional GIS software without an awareness of the DGGRS through the use of encodings for zone data and zone lists intended for this purpose.*

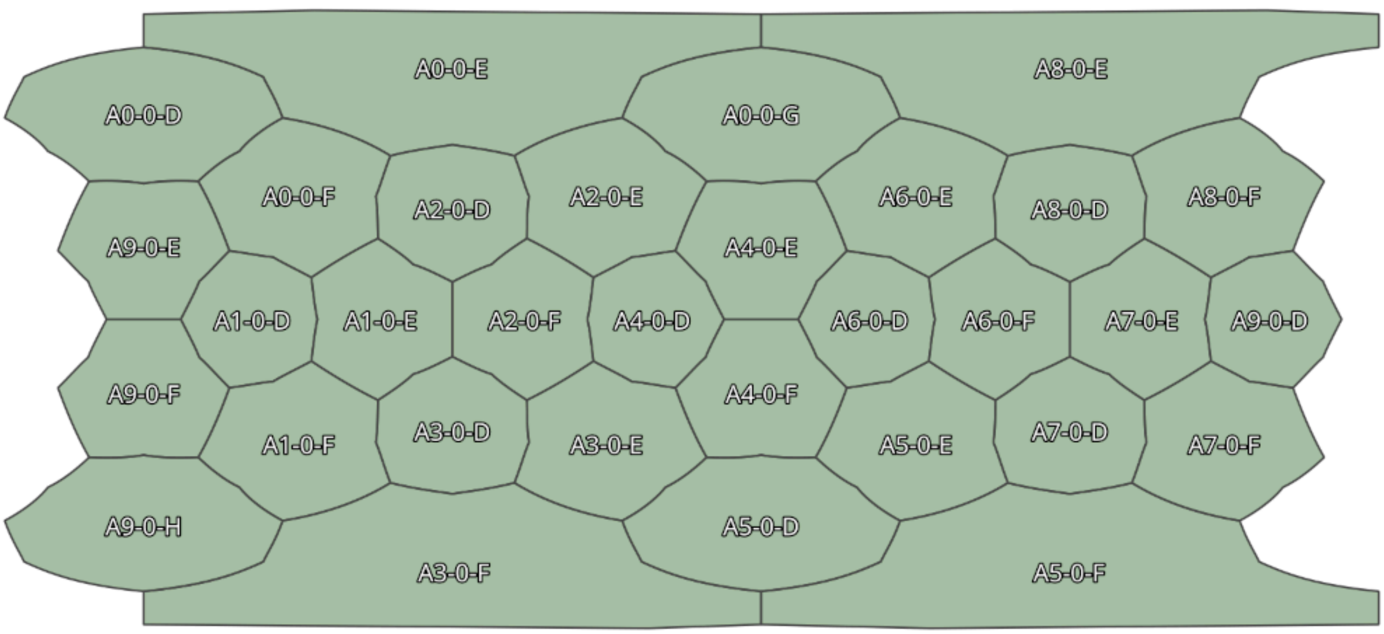


Figure C.2 — GeoJSON response for zone query for ISEA3H DGGRS at level 1 visualized in QGIS

C.5.2. Querying for a spatio-temporal subset

To request a list of zones for an area of interest, the `bbox` or `subset` query parameter can be used. For temporal datasets, the `datetime` can be used as well as `subset=time(…)` to constrain the query to a time of interest. For restricting the same zone query to a bounding box area from (40°N, 30°E) to (60°N,50°E), the client could make either of the following requests:

https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA3H/zones?bbox=30,40,50,60&zone-level=1&compact-zones=false

https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA3H/zones?subset=Lat(40:60),Lon(30:50)&zone-level=1&compact-zones=false

and obtain the following response (negotiating a JSON zone list again):


```
{
   "zones" : [ "A0-0-G", "A6-0-E" ],
   "returnedAreaMetersSquare" : 31170676883138.8,
   "links" : [
      { "rel" : "[ogc-rel:dggrs]", "title" : "ISEA3H DGGS for GEBCO",
        "href" : "/ogcapi/collections/gebco/dggs/ISEA3H"
      },
      { "rel" : "[ogc-rel:dggrs-definition]", "title" : "ISEA3H DGGRS
definition",
         "href" : "/ogcapi/dggrs/ISEA3H"
      },
      { "rel" : "[ogc-rel:geodata]", "href" : "/ogcapi/collections/gebco" }
   ],
   "linkTemplates" : [
      { "rel" : "[ogc-rel:dggrs-zone-data]", "title" : "ISEA3H data for GEBCO",
        "uriTemplate" : "/ogcapi/collections/gebco/dggs/ISEA3H/zones/{zoneId}/
data",
      }
   ]
}
```


Listing

This time, the server returned only two zones of ISEA3H level 1.

A visualization of the response from negotiating `application/geo+json` for the above request, together with the bounding box used, is seen below.

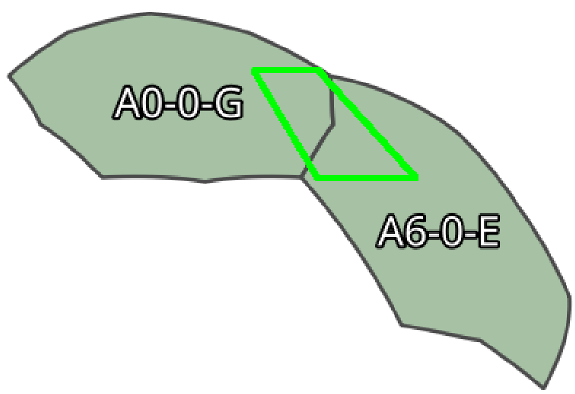
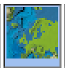
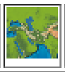


Figure C.3 — GeoJSON response for zone query for ISEA3H DGGRS at level 1 for a (40°N, 30°E) to (60°N,50°E) bounding box visualized in QGIS

As seen below, an HTML representation could include rendered previews of the data for the zone area, summary information, as well as links to retrieve the data and access the zone information resource for each zone. Note that the significant variance in area (-11.31% and +6.66% from the reference mean zone (whose area is calculated by dividing the Earth surface area by the total number of zones) is due to the fact that `A0-0-G` is a pentagon (in the ISEA projection — on the Earth surface it looks more like a decagon) which occupies 5/6th the area of hexagons such as `A6-0-E` (which on the Earth surface is one of the hexagons appearing as a nonagon), as well as the fact that at level 1, the number of pentagons (12) is significant compared to the total number of zones (32).

Zone ID	Map	Data	Zone Geometry	Area	Variance	Min Lat	Min Lon	Max Lat	Max Lon
Level 1 (variance relative to **15,939,550.68** km² reference mean zone surface area)									
A0-0-G		data	geojson.io download	**14,168,489.49** km²	-11.11 %	39.4695395487835	-29.3049078772015	79.237136301031	51.7049078779127
A6-0-E		data	geojson.io download	**17,002,187.39** km²	+6.67 %	12.16084689288	27.0928967551823	58.6117167667327	83.1292006259654

Total surface area: **31,170,676.88** km²

(~15,939,550.68 km² level 1 zone precision)

Figure C.4 — Sample HTML response for a zone query for ISEA3H DGGRS at level 1 for a (40°N, 30°E) to (60°N,50°E) bounding box

Of course with a higher zone level, many more zones can be returned:

https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA3H/zones?bbox=30,40,50,60&zone-level=8&compact-zones=false

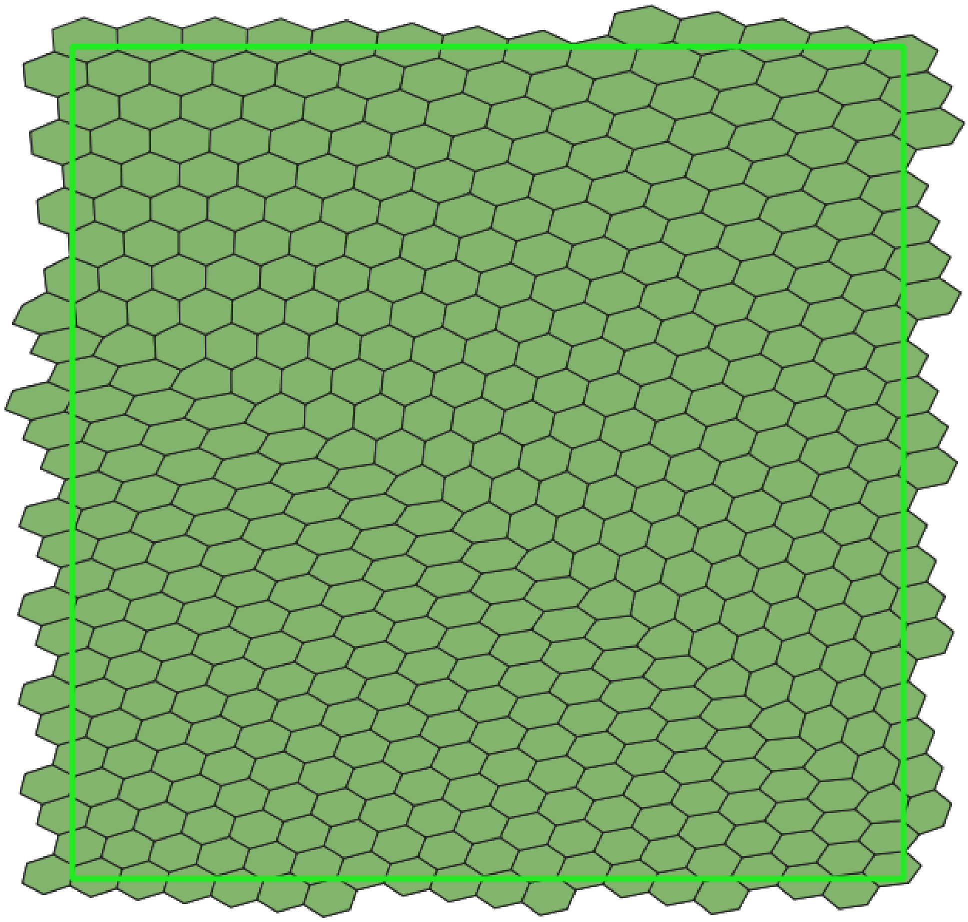


Figure C.5 — Sample HTML response for a zone query for ISEA3H DGGRS at level 8 for a (40°N, 30°E) to (60°N,50°E) bounding box

C.5.3. Zones representing data availability

The primary purpose of the Zone Query requirements class is to ask questions of the type *“Where is it?”*. The answer is a region of spacetime represented by a list of zones. This region might correspond to where data is available for a particular collection of data, whether that data is static or the result of a processing workflow. Later examples will demonstrate more complex queries using CQL2 expressions. These resulting spatiotemporal regions could otherwise have been communicated as a multi-polygon vector mask or as a rasterized 1-bit image or opacity channel.

The following example illustrates the case of a collection (elevation data from Viewfinder Panoramas) which does not have data for everywhere on the globe (only over land), using a higher zone level of 6 for the query.

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA3H/zones?zone-level=6&compact-zones=false

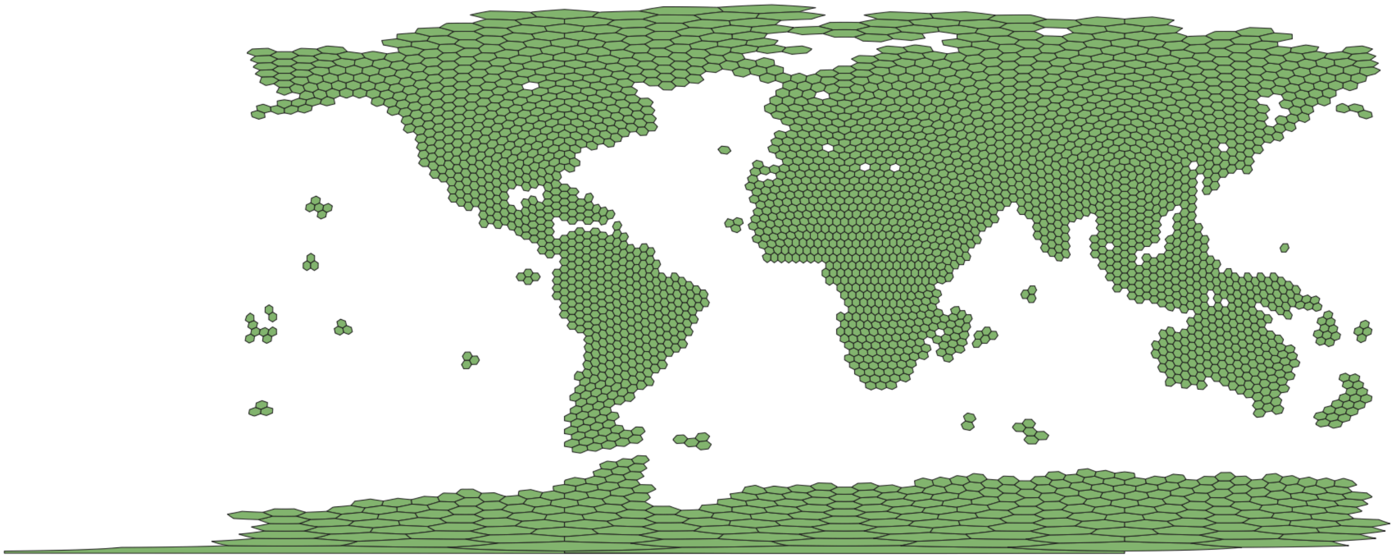


Figure C.6 — GeoJSON response for zone query for ISEA3H DGGRS at level 6 for a Viewfinder Panoramas land elevation collection, visualized in QGIS

Using a bounding box to restrict the query to an area of interest, a higher zone level of 10 could be used:

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA3H/zones?zone-level=10&bbox=30,40,50,50&compact-zones=false

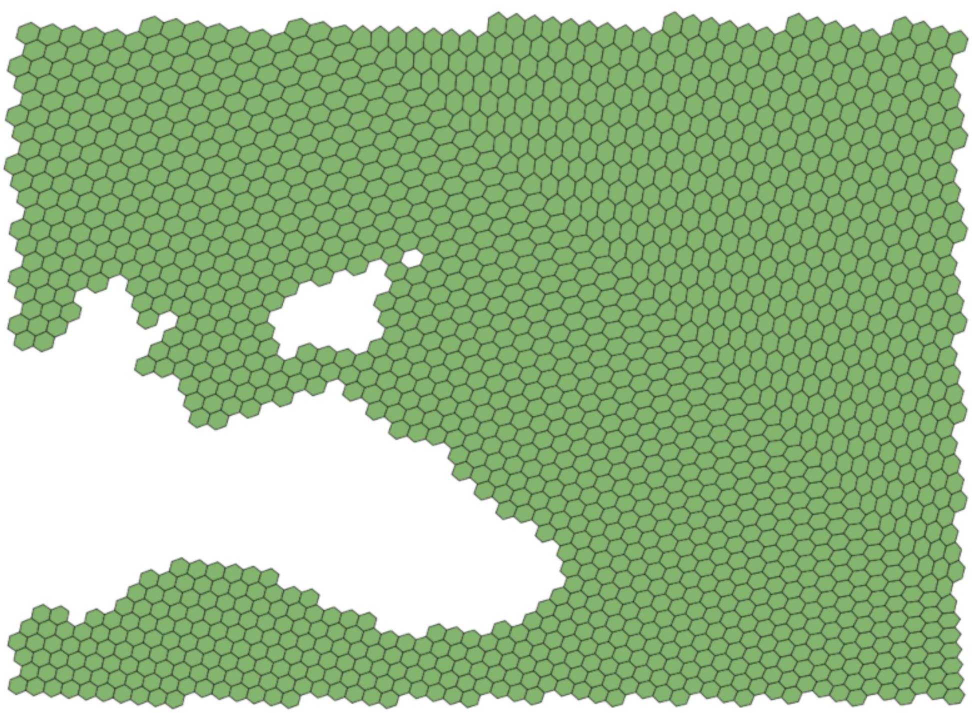


Figure C.7 — GeoJSON response for zone query for ISEA3H DGGRS at level 10, using a (40°N, 30°E) to (40°N,50°E) bounding box, for a Viewfinder Panoramas land elevation collection, visualized in QGIS

An implementation of the DGGS API works independently of the nature of the data, whether dealing with gridded coverage or vector features. This allows easily performing complex analytics involving heterogeneous data sources. The previous examples all used gridded coverage data sources. The following example illustrates the same capabilities using the Natural Earth states and provinces collection:

https://maps.gnosis.earth/ogcapi/collections/NaturalEarth:cultural:ne_10m_admin_1_states_provinces/dggs/ISEA3H/zones?zone-level=6&compact-zones=false

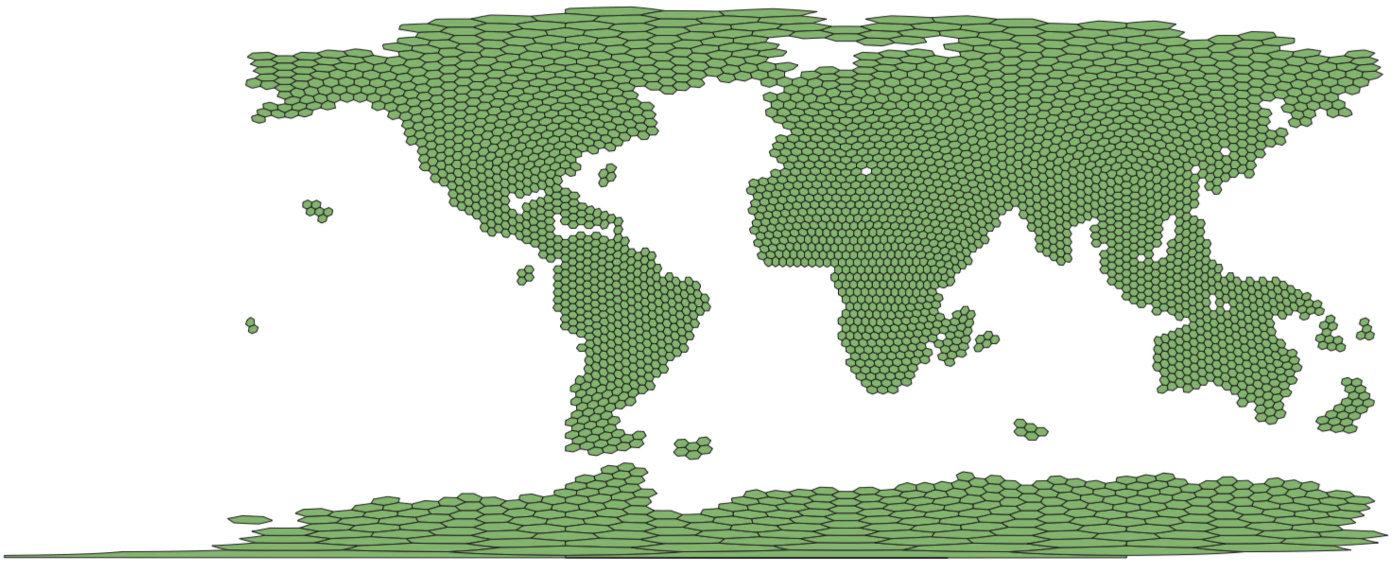


Figure C.8 — GeoJSON response for zone query for ISEA3H DGGRS at level 6, for a Natural Earth states and provinces collection, visualized in QGIS

The similarity between the response for this collection of states and provinces features compared to the same request earlier for a gridded elevation coverages showcases the potential for DGGS zone queries to facilitate data integration.

C.5.4. Querying sub-zones of a parent zone

The Zone Query requirements class defines the `parent-zone` query parameter as one more way to restrict the query spatially (and temporally in the case of a temporal DGGRS), which is particularly useful for hierarchical exploration. For example, to restrict the query to sub-zones of `A6-0-E` rather than using a bounding box, the client could perform the following request (at level 8):

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA3H/zones?zone-level=8&parent-zone=A6-0-E&compact-zones=false

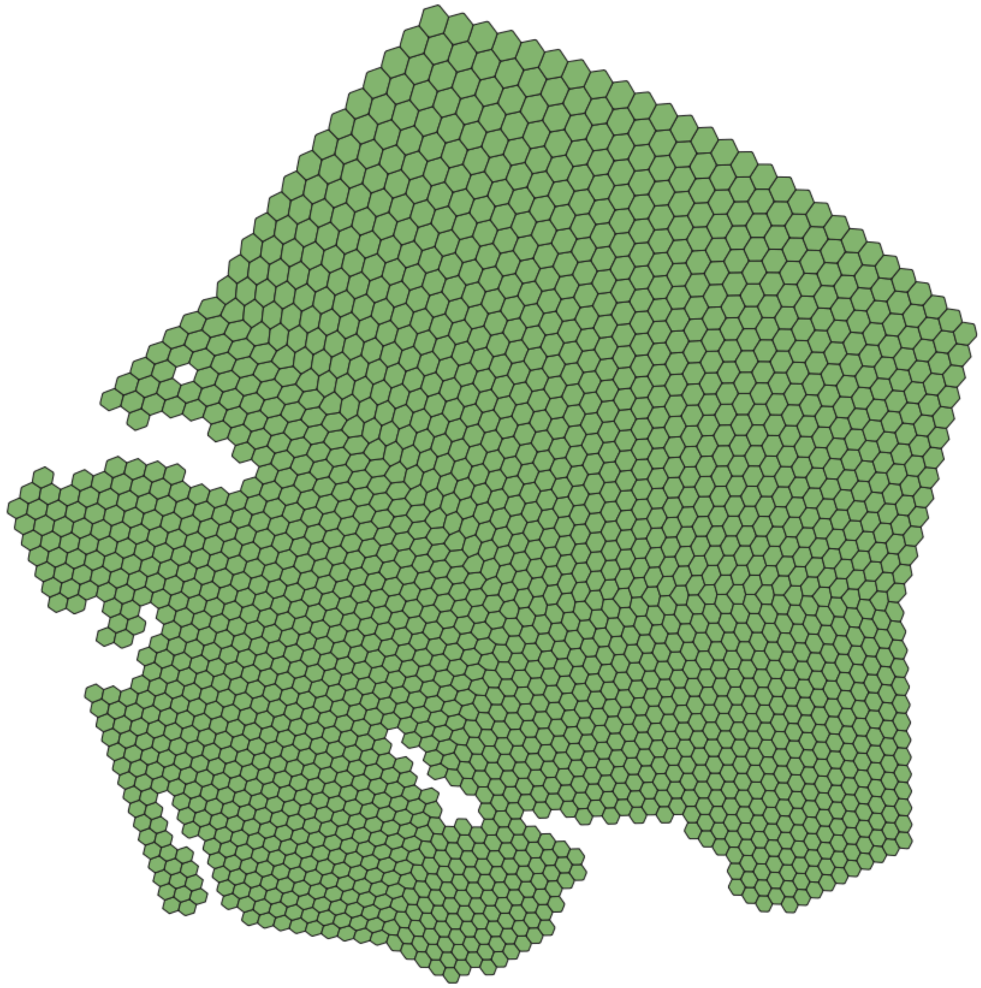


Figure C.9 — GeoJSON response for zone query for ISEA3H DGGRS at level 8, for parent zone A6-0-E, for a Viewfinder Panoramas land elevation collection, visualized in QGIS

The client could then pick one of the returned zones, say `E6-317-A`, for deeper exploration at level 16:

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA3H/zones?zone-level=16&parent-zone=E6-317-A&compact-zones=false

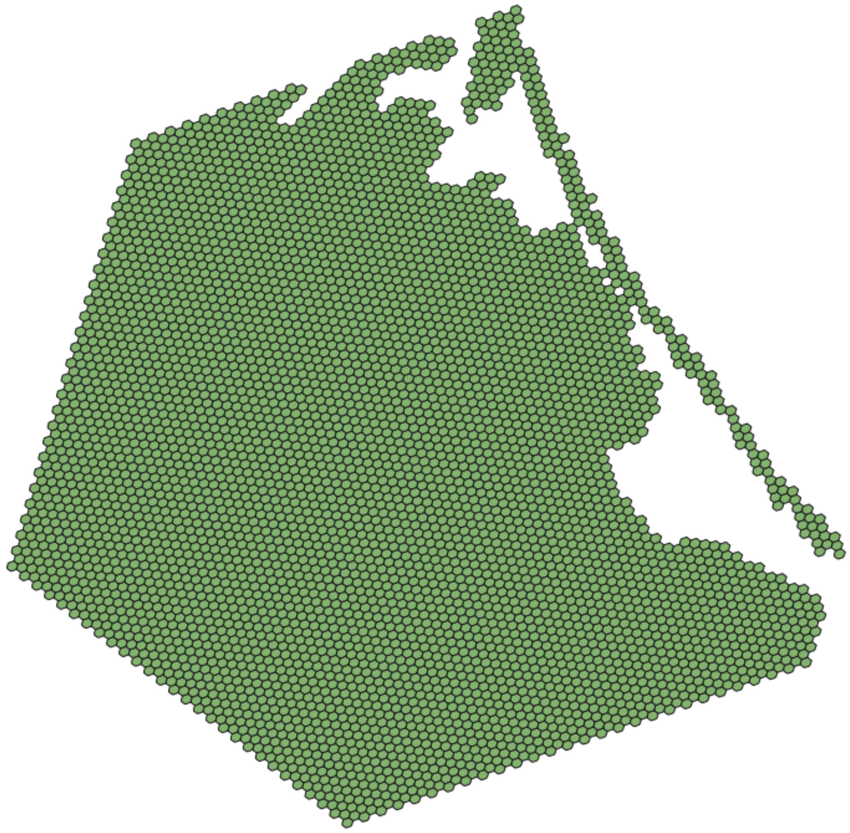


Figure C.10 — GeoJSON response for zone query for ISEA3H DGGRS at level 16, for parent zone E6-317-A, for a Viewfinder Panoramas land elevation collection, visualized in QGIS

Going back to A6-0-E (a level 1 zone), this could also be used to retrieve the immediate children by requesting the next level (2), to step down the hierarchy a single refinement level at a time:

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA3H/zones?zone-level=2&parent-zone=A6-0-E&compact-zones=false

Since they all contain data, the seven children zones (one centroid child and six vertex children) are returned.

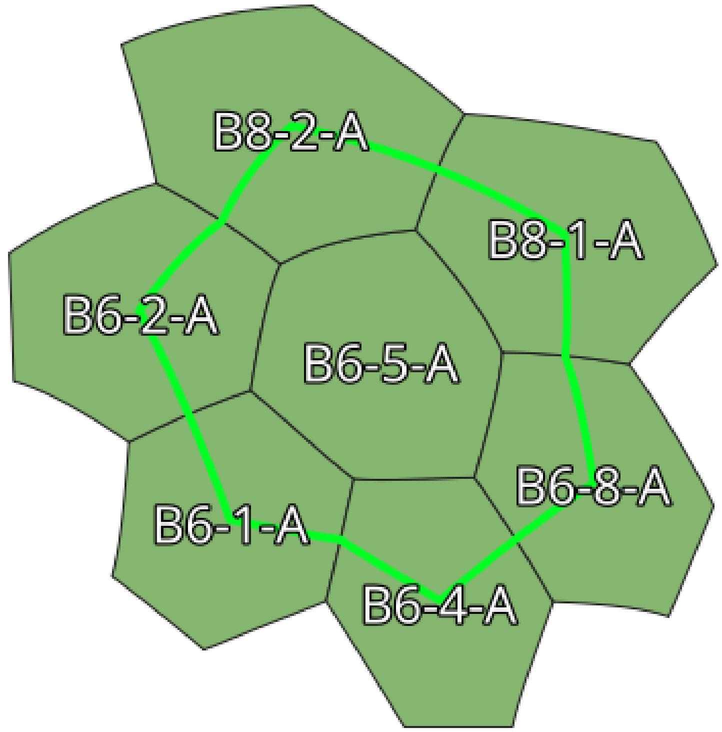


Figure C.11 — GeoJSON response for zone query for ISEA3H DGGRS at level 2, for parent zone A6-0-E (outline shown), for a Viewfinder Panoramas land elevation collection, visualized in QGIS

C.5.5. ISEA9R (rhombus) Zone Queries

Although the examples so far have used the ISEA3H DGGRS, the DGGS API Standard is completely agnostic of the DGGRS. Future planned parts for OGC Abstract Topic 21 will standardize new categories of DGGH, including volumetric (Part 2), temporal (Part 3) and axis-aligned (Part 4). The DGGS API Standard, as demonstrated in this examples section, is ready to handle any DGGRS, including DGGRSs based on these new DGGH types.

The following examples demonstrates equivalent zone queries using the ISEA9R DGGS, a dual DGGS of ISEA3H even refinement levels, which can be considered axis-aligned in a CRS derived from rotating and shearing the ISEA planar projection. In the ISEA planar projection, the ISEA9R zones are shaped as rhombuses.

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA9R/zones?zone-level=3&compact-zones=false

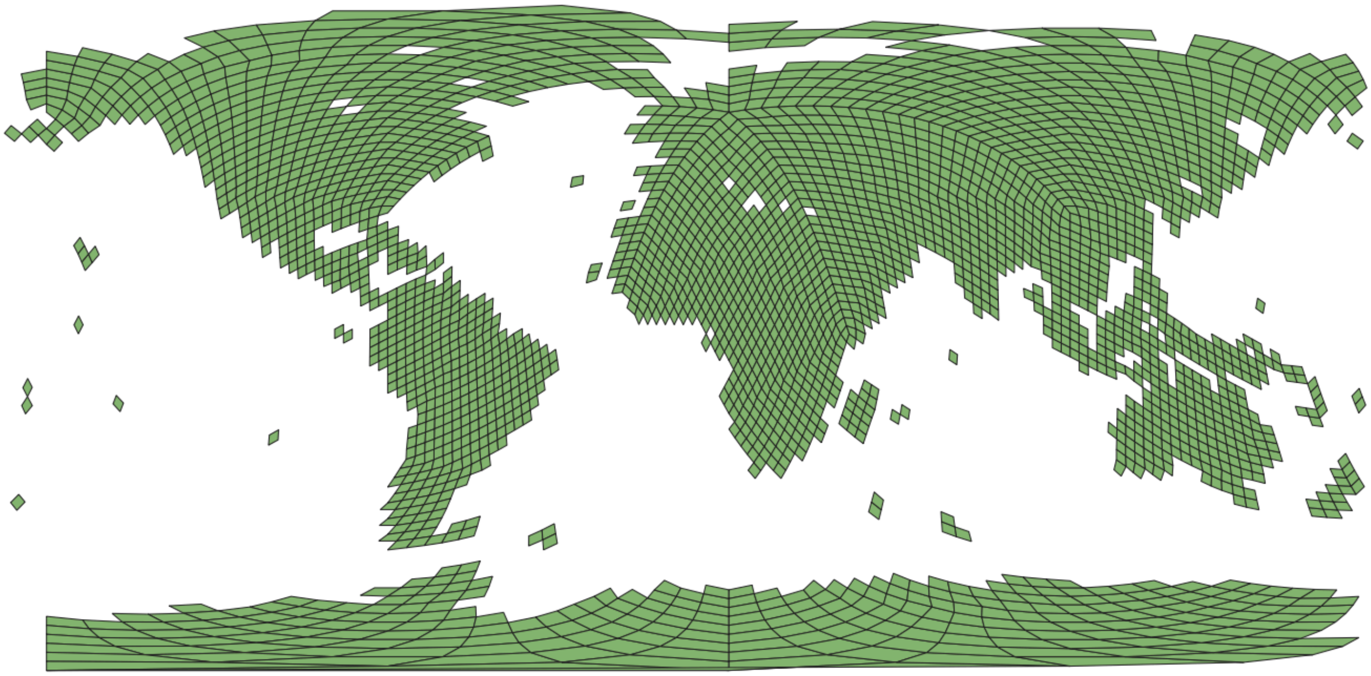


Figure C.12 — GeoJSON response for zone query for ISEA9R DGGRS at level 3 for a Viewfinder Panoramas land elevation collection visualized in QGIS

https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA9R/zones?bbox=30,40,50,60&zone-level=4&compact-zones=false

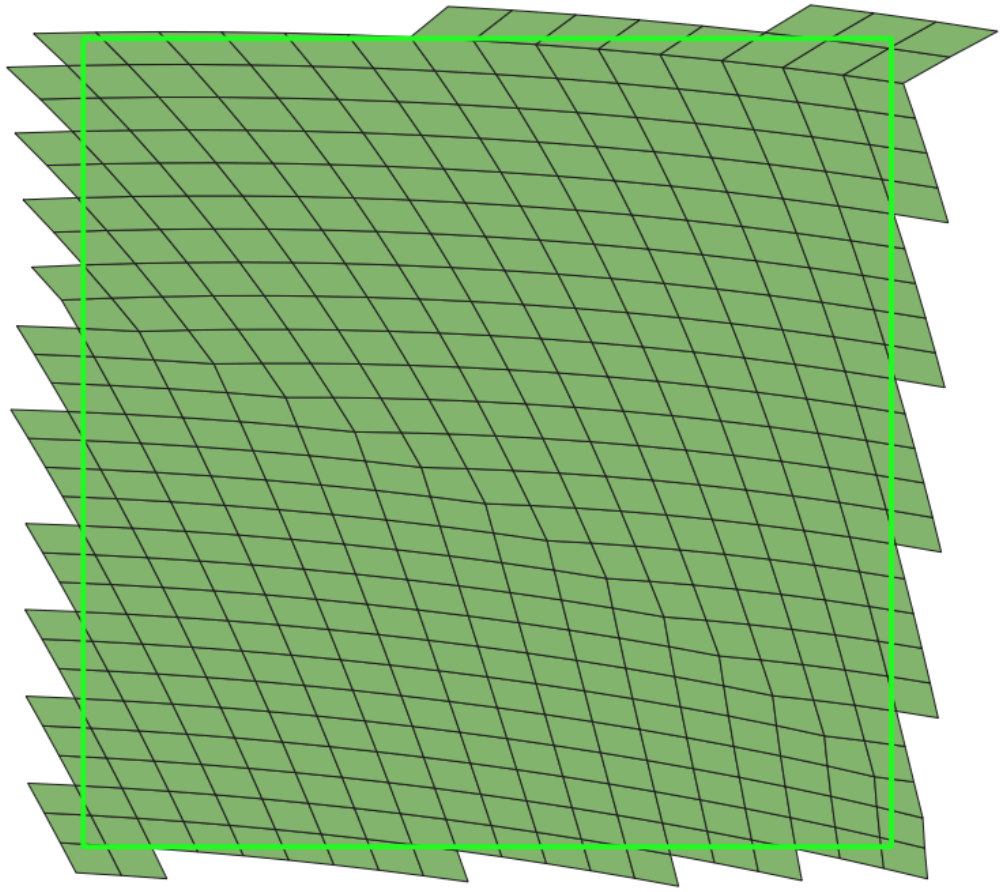


Figure C.13 — GeoJSON response for zone query for ISEA9R DGGRS at level 4 for a (40°N, 30°E) to (60°N,50°E) bounding box visualized in QGIS

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA9R/zones?zone-level=4&parent-zone=B6-2&compact-zones=false

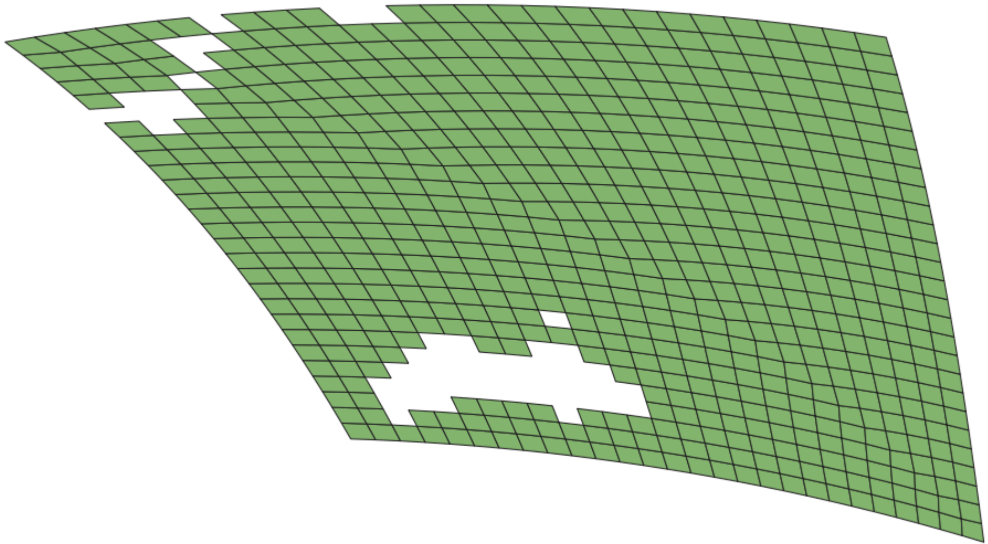


Figure C.14 — GeoJSON response for zone query for ISEA9R DGGRS at level 4 for parent zone B6-2, for Viewfinder Panoramas, visualized in QGIS

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA9R/zones?zone-level=8&parent-zone=E6-317&compact-zones=false

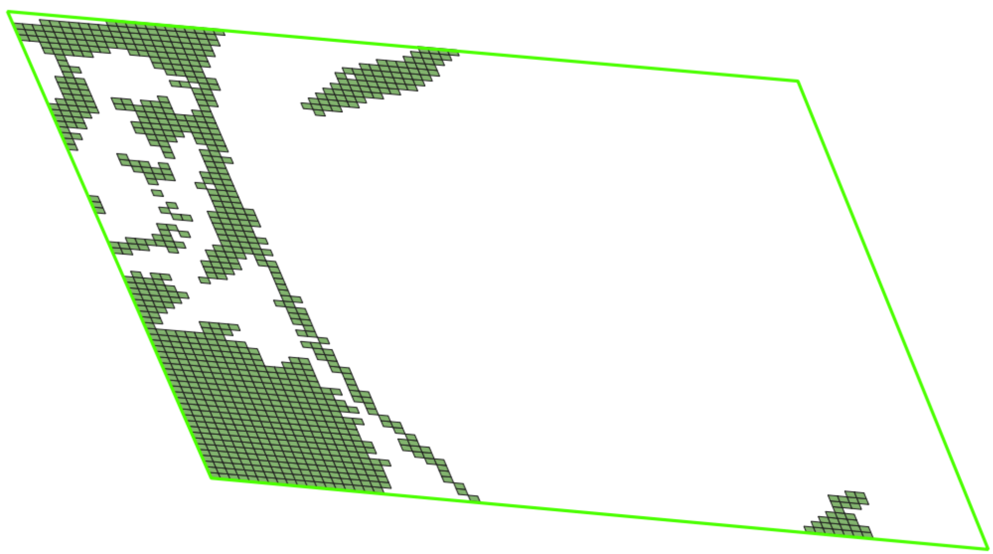


Figure C.15 — GeoJSON response for zone query for ISEA9R DGGRS at level 8 for parent zone E6-317, for Viewfinder Panoramas, visualized in QGIS

Although ISEA9R is axis-aligned, it is also perfectly equal area.

C.5.6. GNOSIS Global Grid (WGS84 rectangles) Zone Queries

The next zone query examples use the GNOSIS Global Grid DGGRS. The GNOSIS Global Grid is also axis-aligned, with the axes being the geographic latitude and longitude. On a plate carée projection, the zones are shaped as rectangles, which conveniently correspond to typical EPSG:4326 / CRS84 bounding boxes.

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA3H/zones?zone-level=5&compact-zones=false

Figure C.16 — GeoJSON response for zone query for GNOSIS Global Grid DGGRS at level 5 for a Viewfinder Panoramas land elevation collection visualized in QGIS

https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/GNOSISGlobalGrid/zones?bbox=30,40,50,60&zone-level=7&compact-zones=false

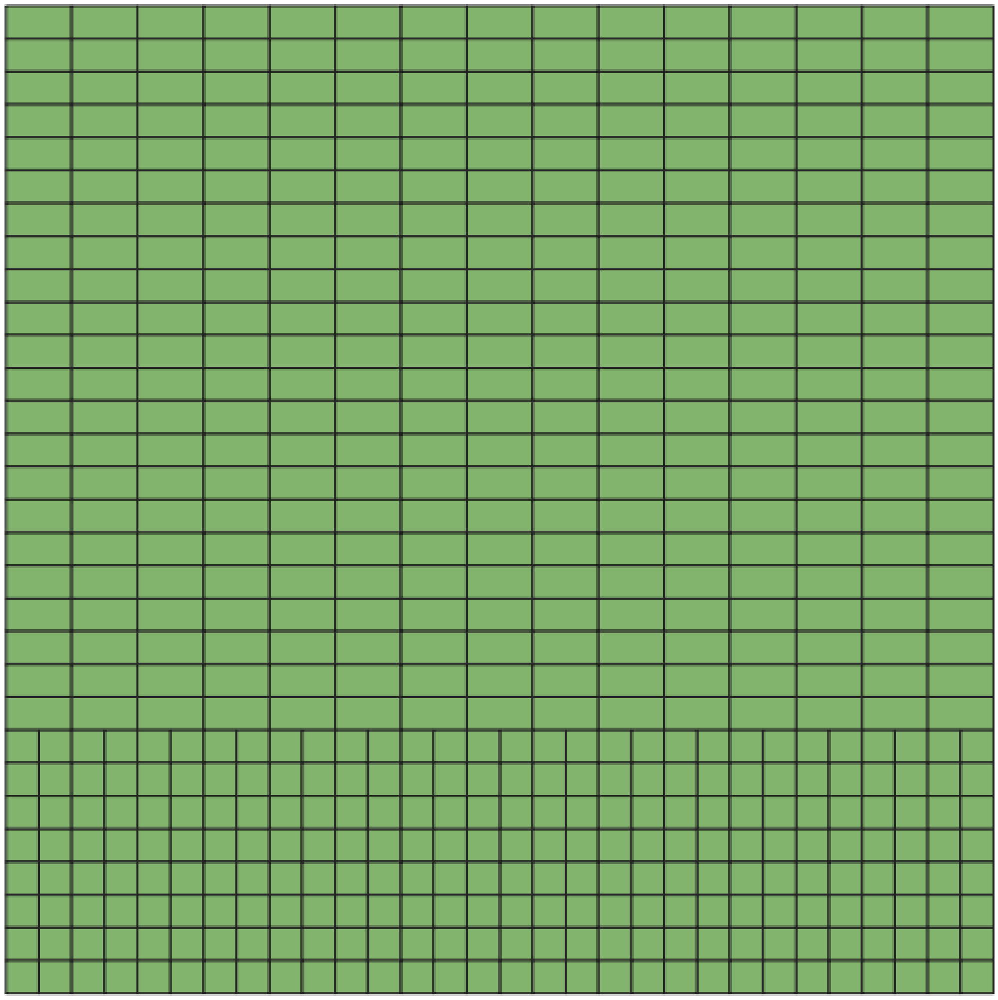


Figure C.17 — GeoJSON response for zone query for GNOSIS Global Grid DGGRS at level 7 for a (40°N, 30°E) to (60°N,50°E) bounding box visualized in QGIS

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/GNOSISGlobalGrid/zones?zone-level=8&parent-zone=2-2-9&compact-zones=false

Figure C.18 — GeoJSON response for zone query for GNOSIS Global Grid DGGRS at level 8 for parent zone 2-2-9, for Viewfinder Panoramas, visualized in QGIS

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/GNOSISGlobalGrid/zones?zone-level=13&parent-zone=6-1F-98&compact-zones=false

Figure C.19 — GeoJSON response for zone query for GNOSIS Global Grid DGGRS at level 13 for parent zone 6-1F-98, for Viewfinder Panoramas, visualized in QGIS

C.6. Compact zone queries

The following examples are reprises of some of the earlier examples leaving compact zones enabled. These examples showcase how the zone compacting mechanism allows to communicate the same spatial area using fewer zones.

C.6.1. ISEA3H compact zone queries

Compacting ISEA3H zones is complicated by sub-zones not being fully congruent. The approach demonstrated here skips a refinement level, taking advantage of the fact that seven of the grandchildren of a zone are congruent. Overlaps are seen, as six holes are left at the vertices of the grandparent, each occupying 1/3rd the size of a congruent grandchild. Note how 7 + 6 x 1/3 = 9 (3 x 3), which corresponds to the two subsequent refinements at the refinement ratio of 3. If the neighboring grandparent of the same refinement level cannot be used, then the non-congruent vertex children must be included, resulting in these overlaps.

https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA3H/zones?bbox=30,40,50,60&zone-level=8

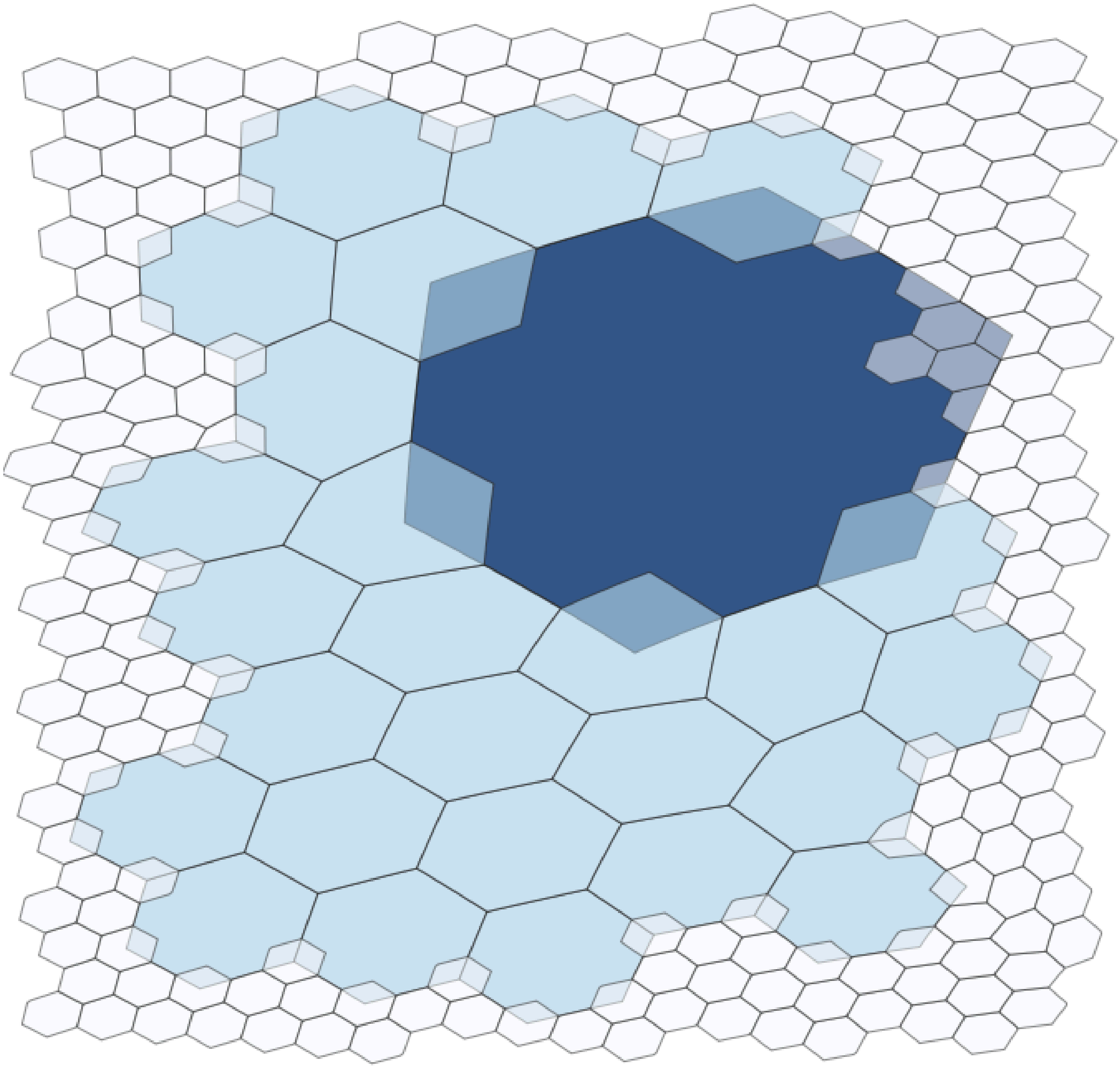


Figure C.20 — GeoJSON compact zones response querying ISEA3H DGGRS at level 8 for a (40°N, 30°E) to (60°N,50°E) bounding box

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA3H/zones?zone-level=6

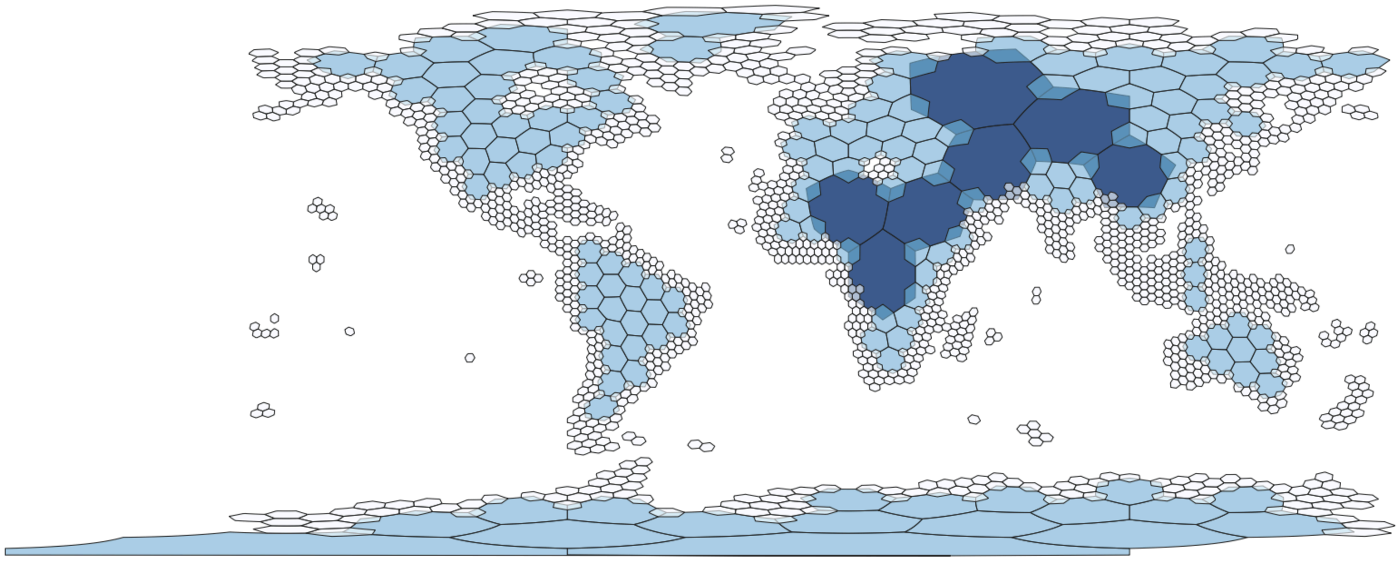


Figure C.21 — GeoJSON compact zones response for querying ISEA3H DGGRS at level 6 for a Viewfinder Panoramas land elevation collection, visualized in QGIS

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA3H/zones?zone-level=10&bbox=30,40,50,50

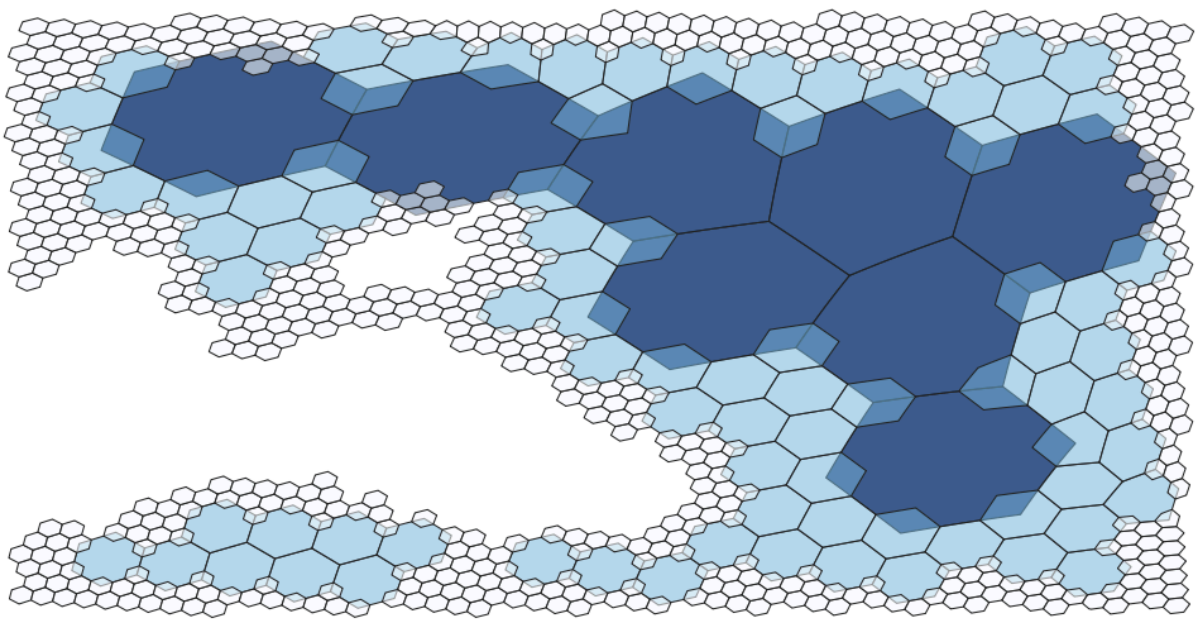


Figure C.22 — GeoJSON compact zones response for querying ISEA3H DGGRS at level 10, using a (40°N, 30°E) to (40°N,50°E) bounding box, for a Viewfinder Panoramas land elevation collection, visualized in QGIS

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA3H/zones?zone-level=16&parent-zone=E6-317-A

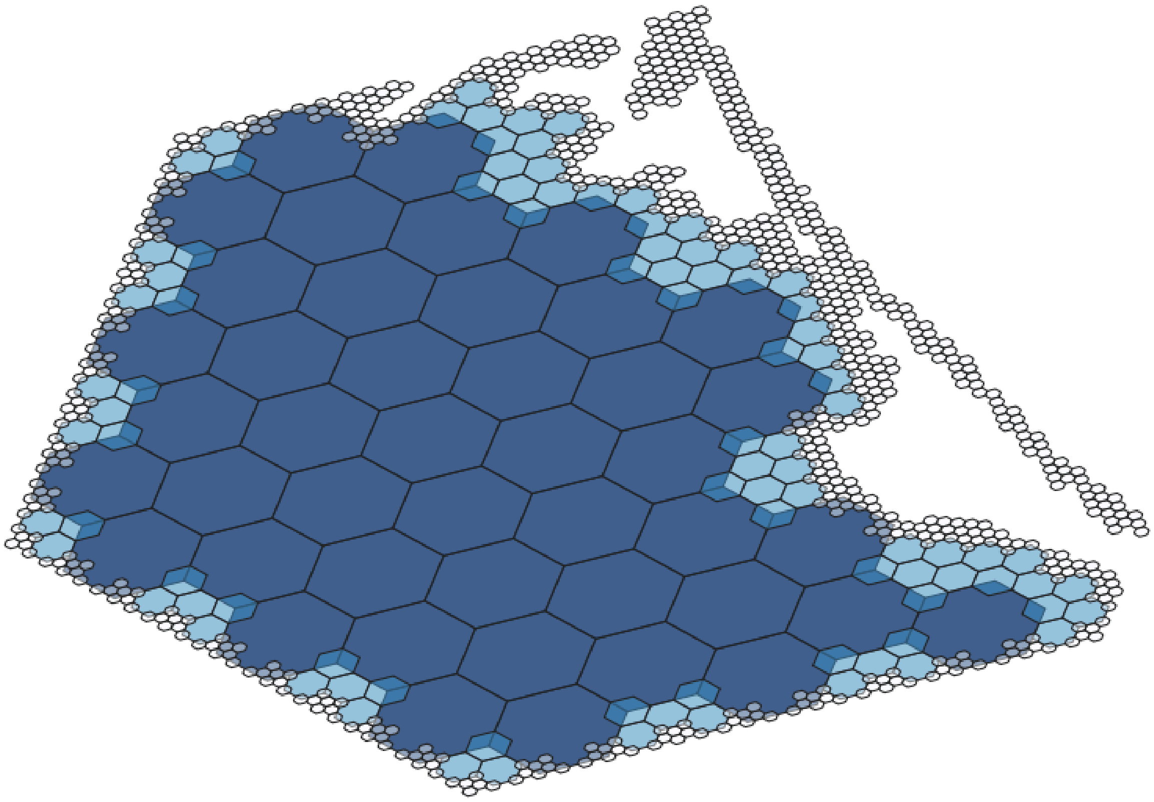


Figure C.23 — GeoJSON compact zones response for querying ISEA3H DGGRS at level 16, for parent zone E6-317-A, for a Viewfinder Panoramas land elevation collection, visualized in QGIS

C.6.2. ISEA9R compact zone queries

Compacting ISEA9R zones can be achieved quite simply given that all sub-zones are fully congruent.

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA9R/zones?zone-level=3

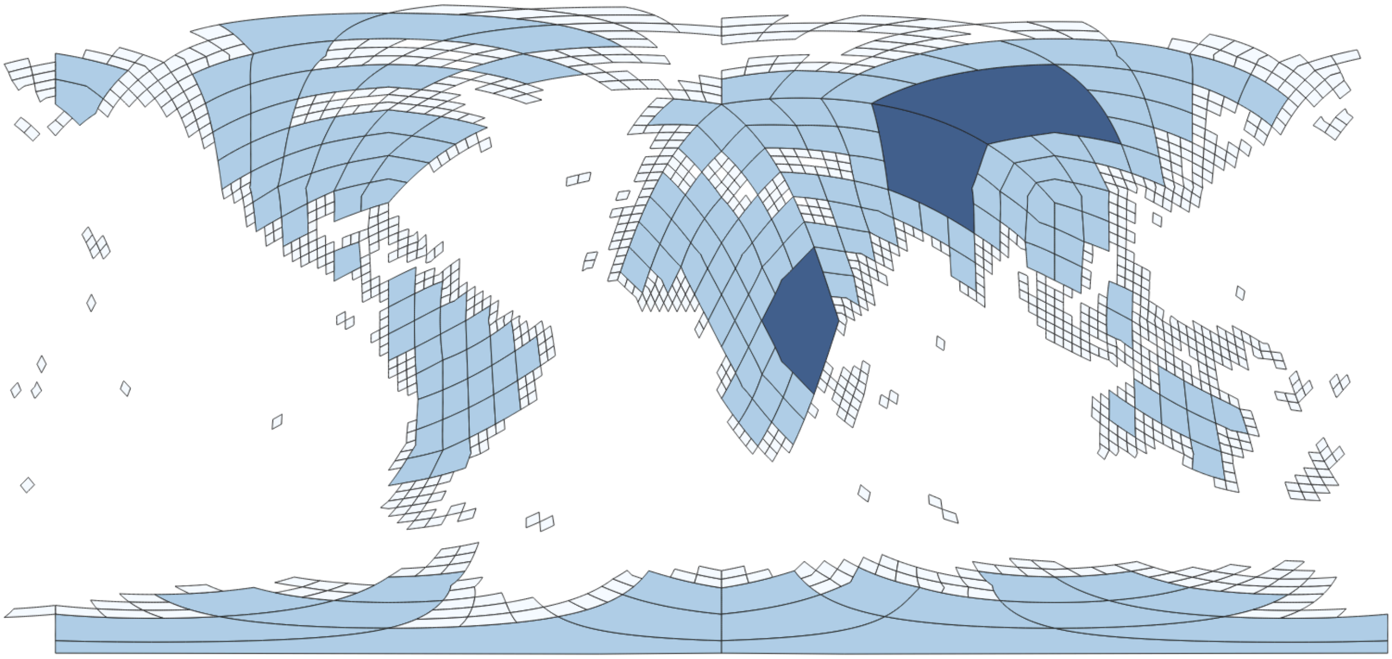


Figure C.24 — GeoJSON compact zones response for querying ISEA9R DGGRS at level 3 for a Viewfinder Panoramas land elevation collection visualized in QGIS

https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA9R/zones?bbox=30,40,50,60&zone-level=4

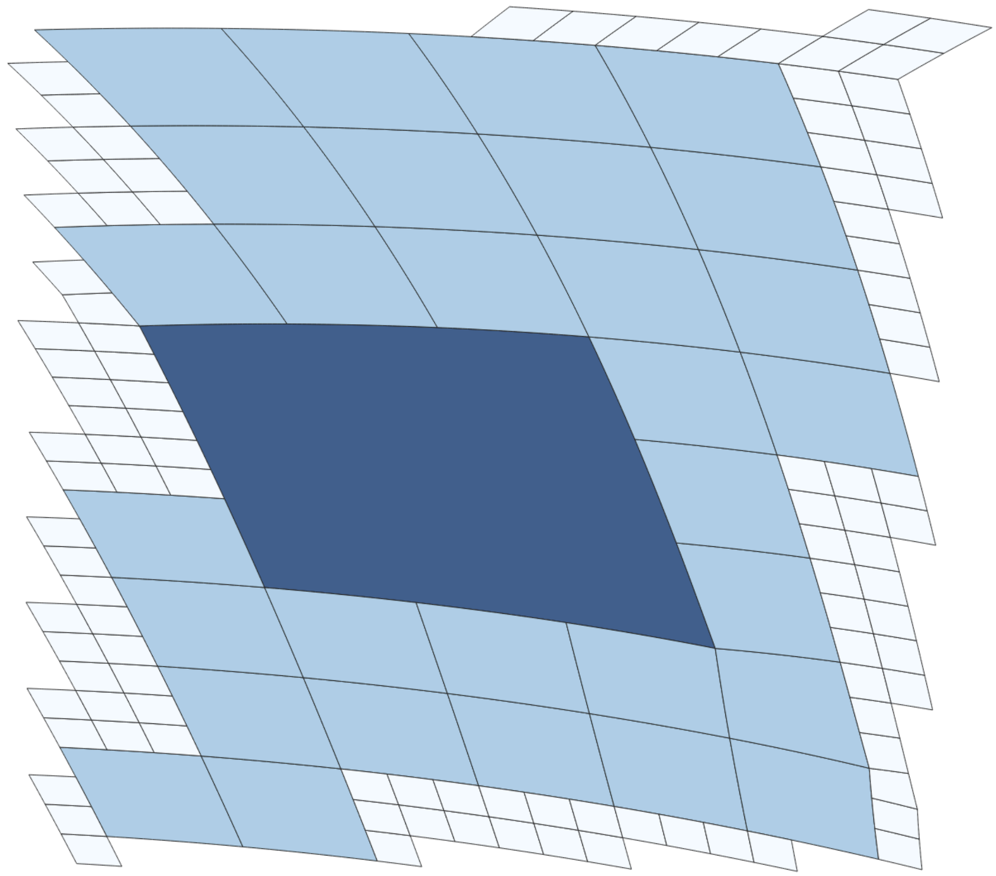


Figure C.25 — GeoJSON compact zones response for querying ISEA9R DGGRS at level 4 for a (40°N, 30°E) to (60°N,50°E) bounding box visualized in QGIS

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA9R/zones?zone-level=4&parent-zone=B6-2

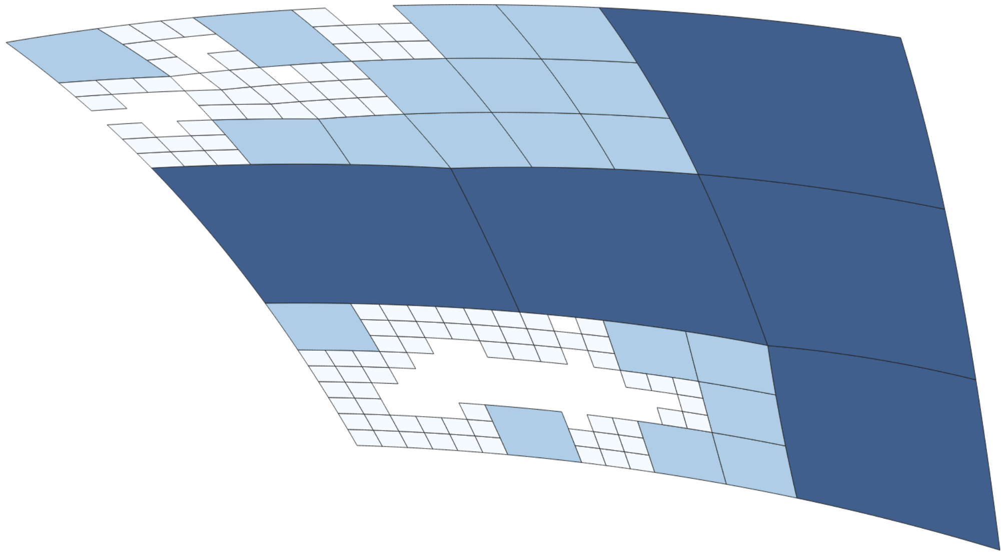


Figure C.26 — GeoJSON compact zones response for querying ISEA9R DGGRS at level 4 for parent zone B6-2, for Viewfinder Panoramas, visualized in QGIS

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA9R/zones?zone-level=8&parent-zone=E6-317

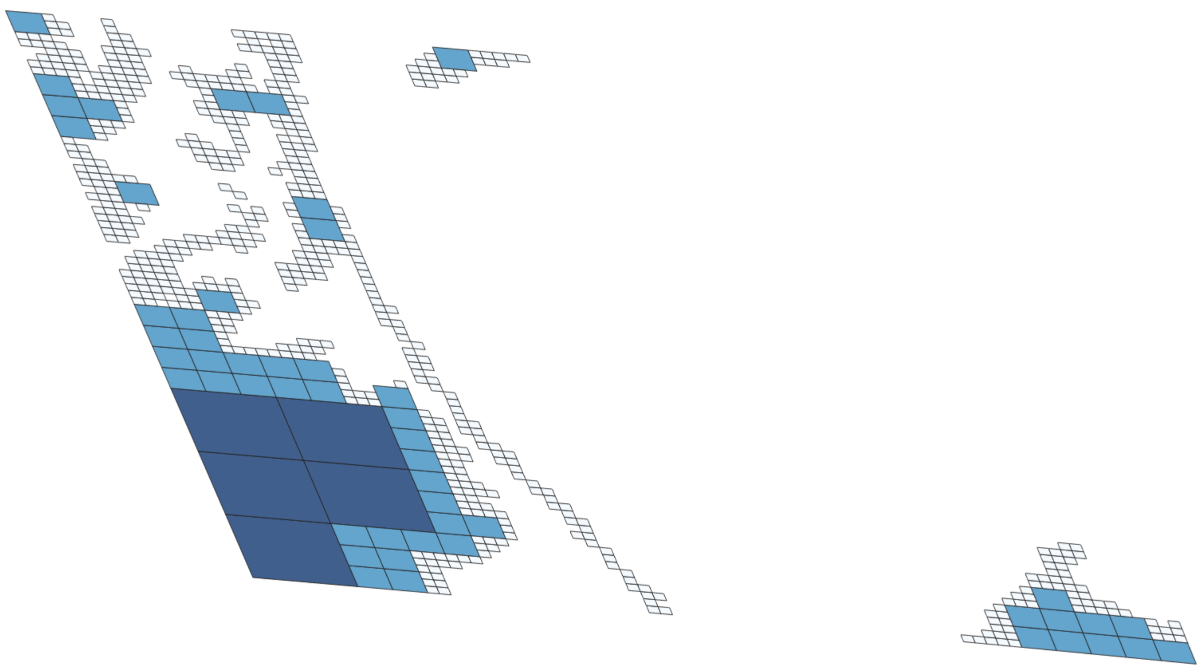


Figure C.27 — GeoJSON compact zones response for querying ISEA9R DGGRS at level 8 for parent zone E6-317, for Viewfinder Panoramas, visualized in QGIS

C.6.3. GNOSIS Global Grid compact zone queries

Compacting GNOSIS Global Grid zones is also simple to achieve since sub-zones are also fully congruent.

To illustrate how the width of the zones attempts to remain equal to the height, the following examples are visualized using the world Mercator projection. As with the previous examples for ISEA3H and ISEA9R, the zones of a given refinement level are filled with the same color. Note that the zones closer to the poles are wider in longitude than zones closer to the equator, and can be seen in a 2-to-1 configuration at jumps in the variable widths coalescence factor such as at the 45° parallel, but these zones still occupy a similar area within the error budget. However, in these illustrations, the polar zones appear much larger than zones of the same level closer to the equator because the Mercator projection enlarges features away from the equator.

https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/ISEA3H/zones?zone-level=5

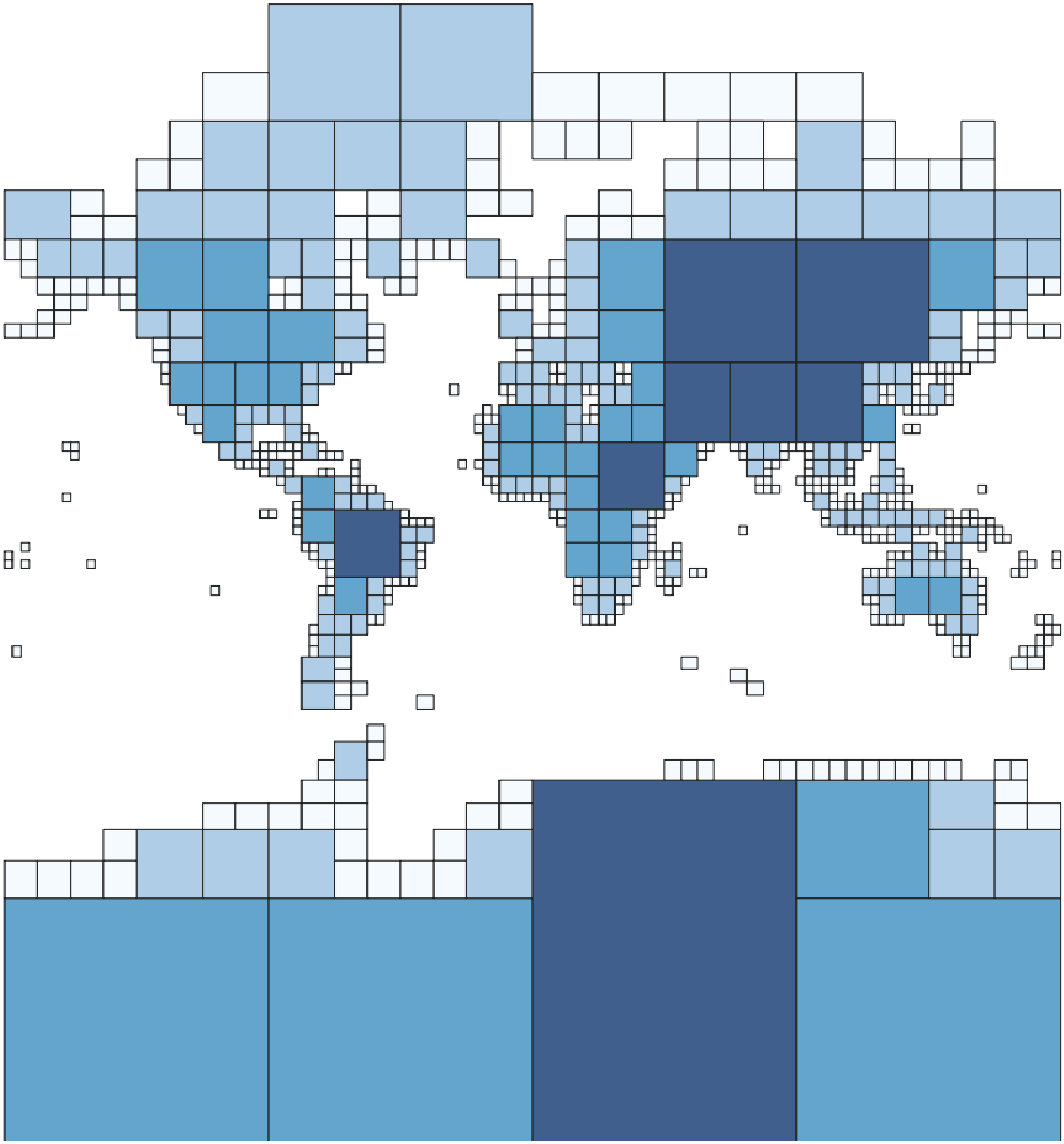


Figure C.28 — GeoJSON compact zones response for querying GNOSIS Global Grid DGGRS at level 5 for a Viewfinder Panoramas land elevation collection visualized in QGIS https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/GNOSISGlobalGrid/zones?bbox=30,40,50,60&zone-level=7

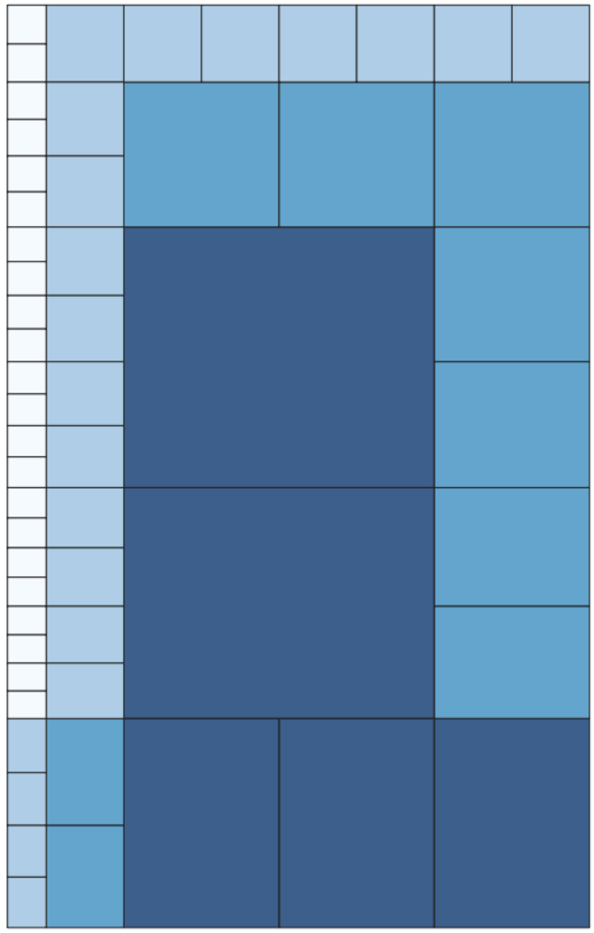


Figure C.29 — GeoJSON compact zones response for querying GNOSIS Global Grid DGGRS at level 7 for a (40°N, 30°E) to (60°N,50°E) bounding box visualized in QGIS https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/GNOSISGlobalGrid/zones?zone-level=8&parent-zone=2-2-9

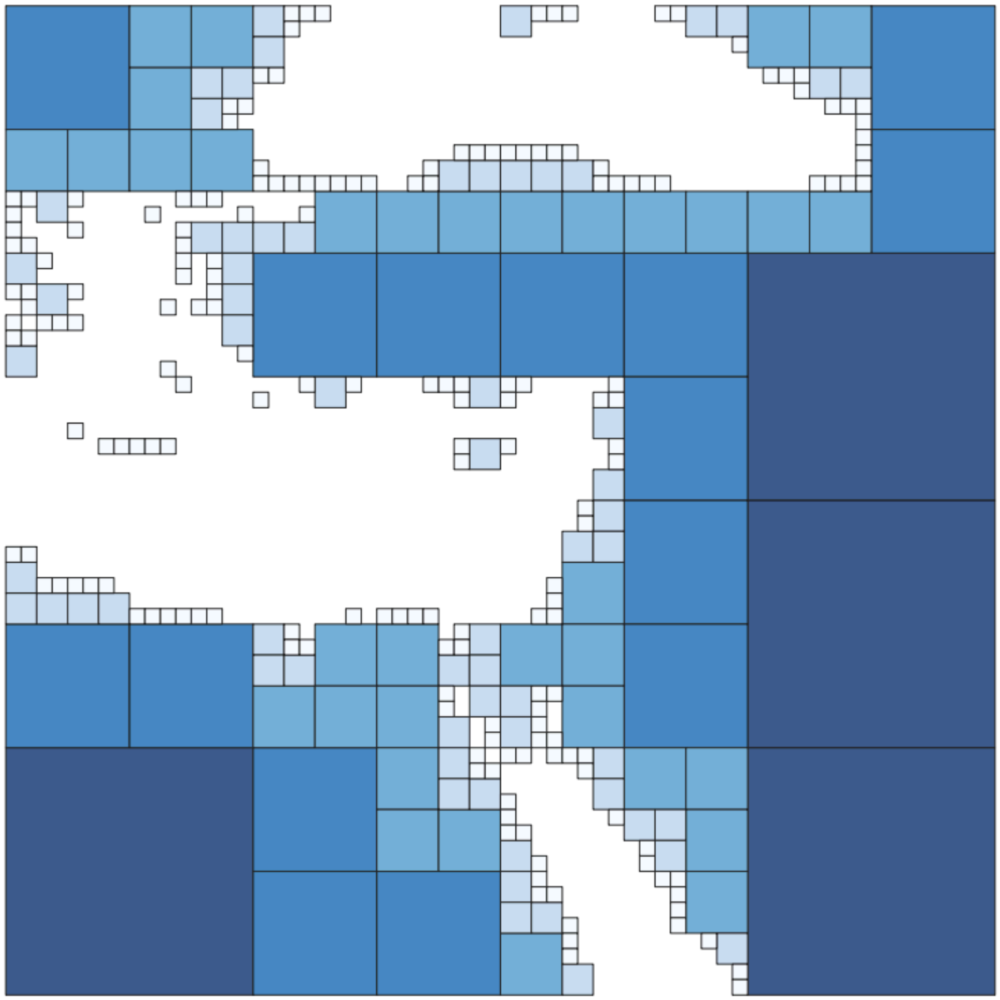


Figure C.30 — GeoJSON compact zones response for querying GNOSIS Global Grid DGGRS at level 8 for parent zone 2-2-9, for Viewfinder Panoramas, visualized in QGIS https://maps.gnosis.earth/ogcapi/collections/SRTM_ViewFinderPanorama/dggs/GNOSISGlobalGrid/zones?zone-level=13&parent-zone=6-1F-98

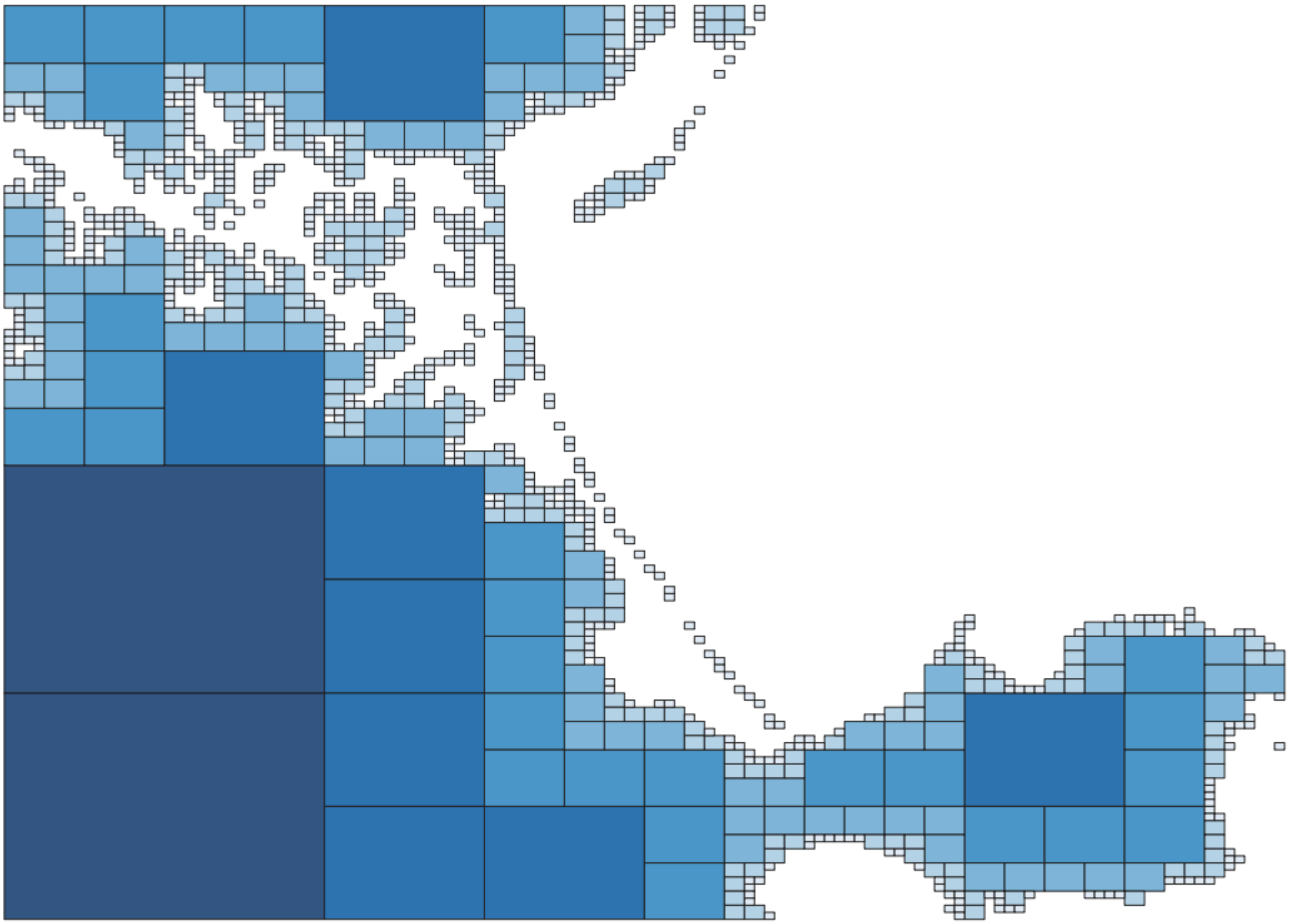


Figure C.31 — GeoJSON compact zones response for querying GNOSIS Global Grid DGGRS at level 13 for parent zone 6-1F-98, for Viewfinder Panoramas, visualized in QGIS

C.7. Filtering zone queries using CQL2 expressions

The Zone Query capabilities are most useful for use cases where the retrieved zone list is the result of some operation, whether a processing workflow or the evaluation of a query expressed by the client. The CQL2 filter capability supports expressing such queries using the OGC Common Query Language (CQL2). Note that the processing workflow could also itself include CQL2 expressions using input/output field modifiers.

The CQL2 requirements class in the OGC API — DGGS Standard specifically calls for the CQL2-Text encoding, which is an intuitive representation with many similarities to SQL. CQL2 defines minimum capabilities (Basic CQL2) as well as additional optional capabilities, such as arithmetic operations and spatial relation functions. A DGGS API implementation's conformance resource

(`/conformance` as defined in OGC API — Common — Part 1: Core "Landing Page" requirements class) enables the client to know which capabilities are supported.

Basic CQL2 includes support for logical operators as well as to compare the value of queryables against literals using relational operators. The following examples will use the same GEBCO collection as earlier examples to illustrate these capabilities. The previous simple zone queries using this collection were not particularly interesting since the collection covers the entire globe, but the use of CQL2 will enable the use of its data values. First, the client would retrieve the queryables by following the `[ogc-rel:queryables]` link from the collection description:

https://maps.gnosis.earth/ogcapi/collections/gebco/queryables

The `application/schema+json` response would look like so:

```
{
   "$schema" : "https://json-schema.org/draft/2020-12/schema",
   "$id" : "https://maps.gnosis.earth/ogcapi/collections/gebco/queryables",
   "title" : "General Bathymetric Chart of the Oceans",
   "type" : "object",
   "properties" : {
      "Elevation" :
      {
         "type" : "number",
         "x-ogc-propertySeq" : 1,
         "x-ogc-definition": "https://qudt.org/vocab/quantitykind/Altitude",
         "x-ogc-unit": "m"
      },
      "rec.geom" : {
         "x-ogc-role" : "primary-geometry",
         "format" : "geometry-point"
      }
   }
}
```

Listing

This informs the client that an `Elevation` queryable is available for use in CQL2 expressions. Although this collection is a gridded coverage, a `rec.geom` queryable is still included in the schema which represents the geometry of each individual grid cell. Since the value associated with these cells represent a point in this collection, the format indicates `geometry-point`. Later examples will make use of this `rec.geom` property.

The client could then submit the following zone query to return only zones where elevation is below sea level (`Elevation < 0`), at refinement level 6:

https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/isea3h/zones?filter=Elevation < 0&zone-level=6

The following image is a visualization of the list of zones returned.

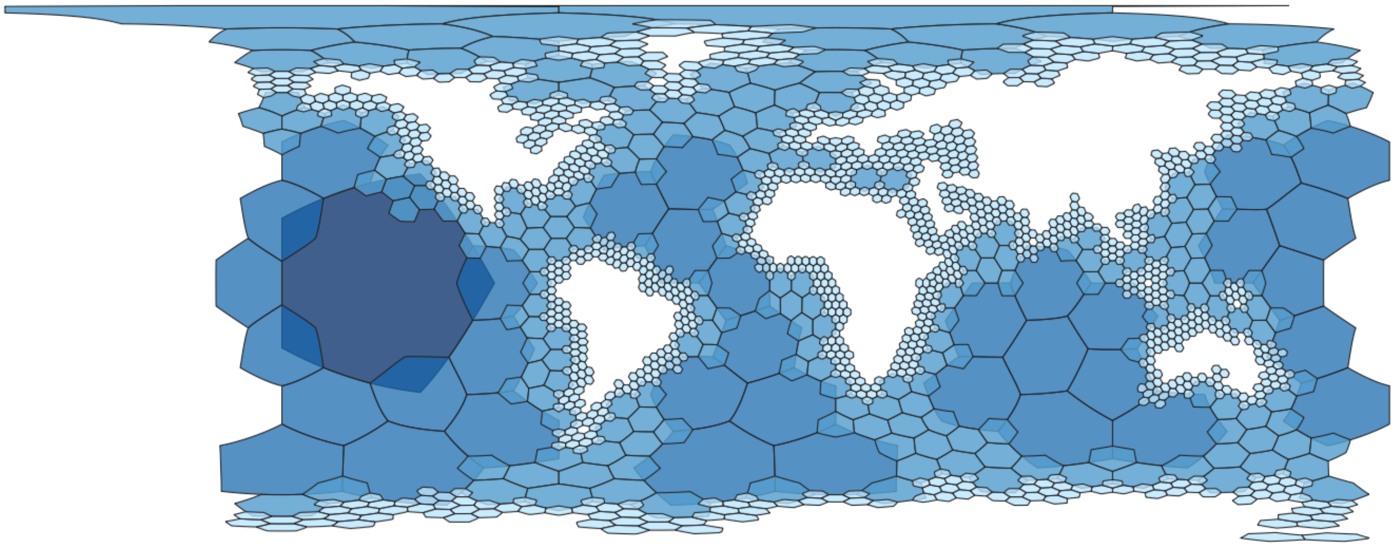


Figure C.32 — GeoJSON compact zones response for querying ISEA3H DGGRS for GEBCO 2014 bathymetry with CQL2 expression `Elevation < 0` visualized in QGIS

The client could also request deeper elevations e.g., `Elevation < -4000`:

https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/isea3h/zones?filter=Elevation < -4000&zone-level=6

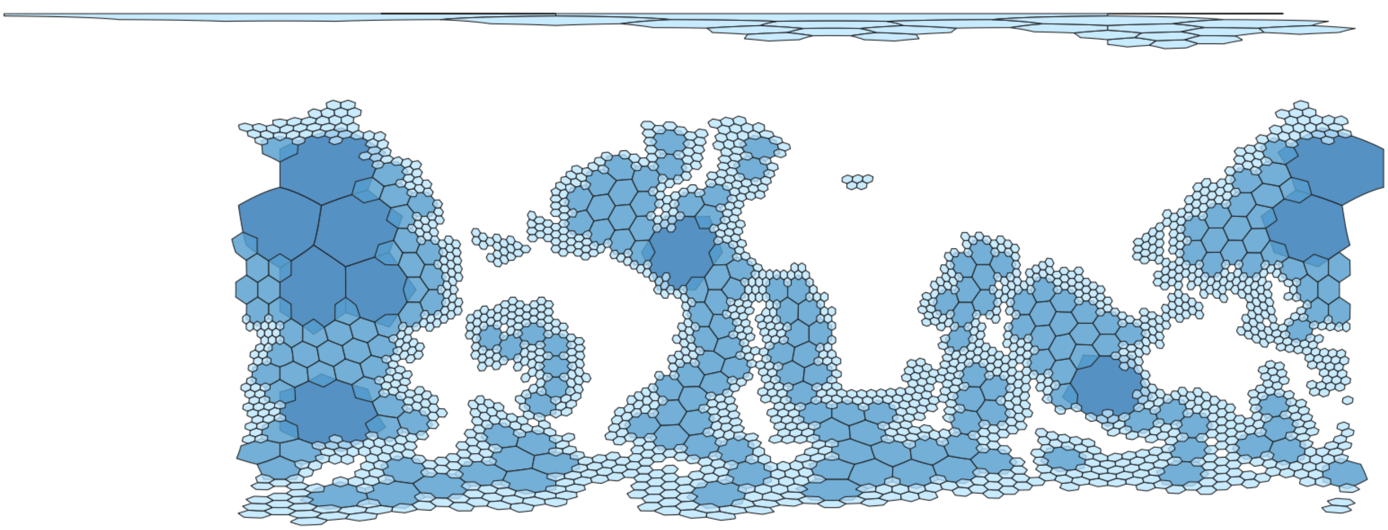


Figure C.33 — GeoJSON compact zones response for querying ISEA3H DGGRS for GEBCO 2014 bathymetry with CQL2 expression `Elevation < -4000` visualized in QGIS

or for high elevations e.g., `Elevation > 3000`:

https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/isea3h/zones?filter=Elevation > 3000&zone-level=6

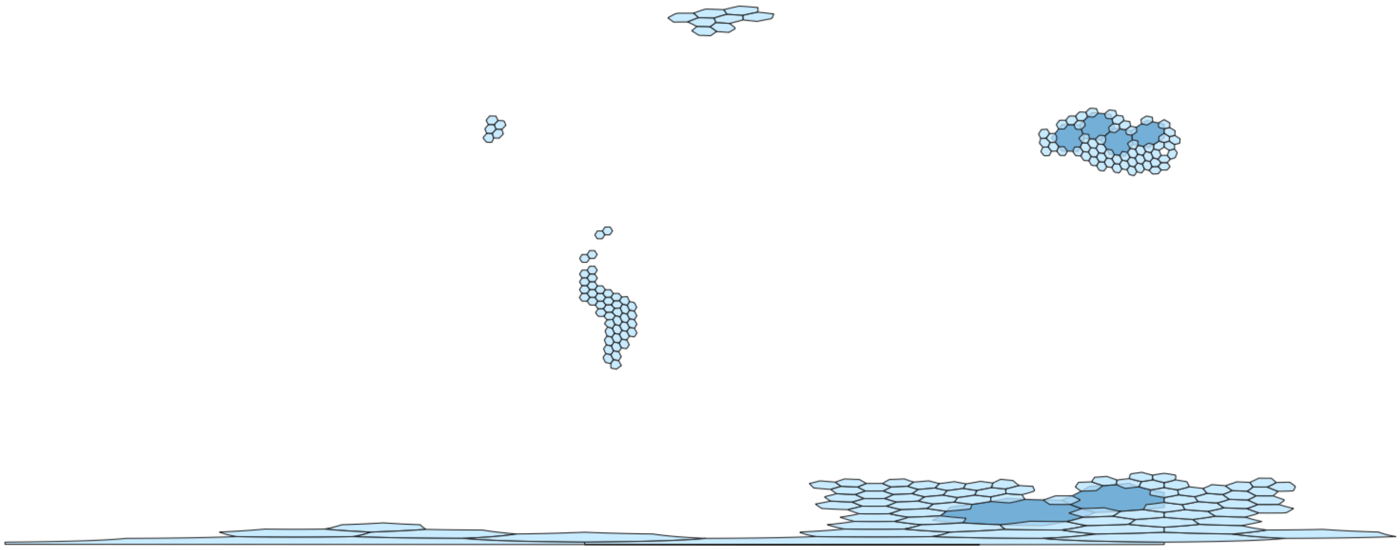


Figure C.34 — GeoJSON compact zones response for querying ISEA3H DGGRS for GEBCO 2014 bathymetry with CQL2 expression `Elevation > 3000` visualized in QGIS

or for either deep or high elevations e.g., `Elevation < -4000 or Elevation > 3000`:

https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/isea3h/zones?filter=Elevation < -4000 or Elevation > 3000&zone-level=6

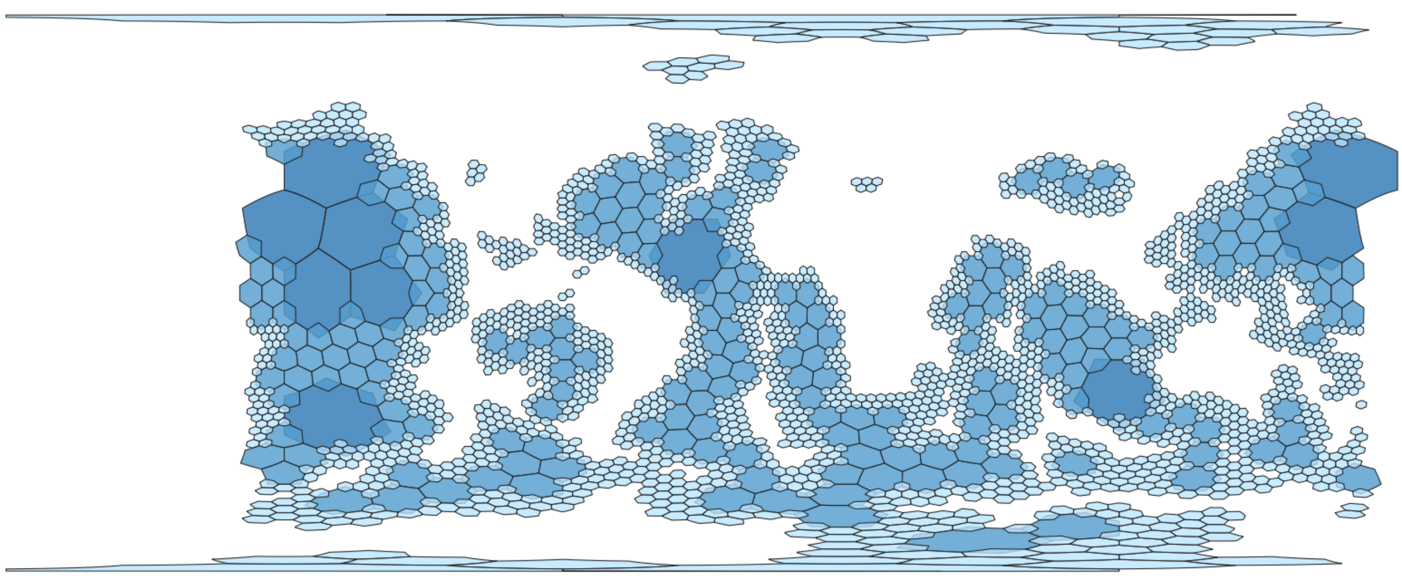


Figure C.35 — GeoJSON compact zones response for querying ISEA3H DGGRS for GEBCO 2014 bathymetry with CQL2 expression `Elevation < -4000 or Elevation > 3000` visualized in QGIS

This last visualization highlights the fact that the Andes and Antarctica mountains are actually very close to a very deep sea.

If the implementation supports spatial relation functions, a query could test intersection with a geometry. In CQL2-Text, spatial geometries are expressed using Well-Known Text (WKT) literals.

The following example returns zones intersecting the Bermuda Triangle, making use of the `rec.geom` queryable, which represents the geometry of the individual coverage grid cells, mentioned earlier, at refinement level 9.

https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/isea3h/zones?zone-level=9&filter=S_INTERSECTS(rec.geom,POLYGON-64.8 32.3,-65.5 18.3,-80.3 25.2,-64.8 32.3)

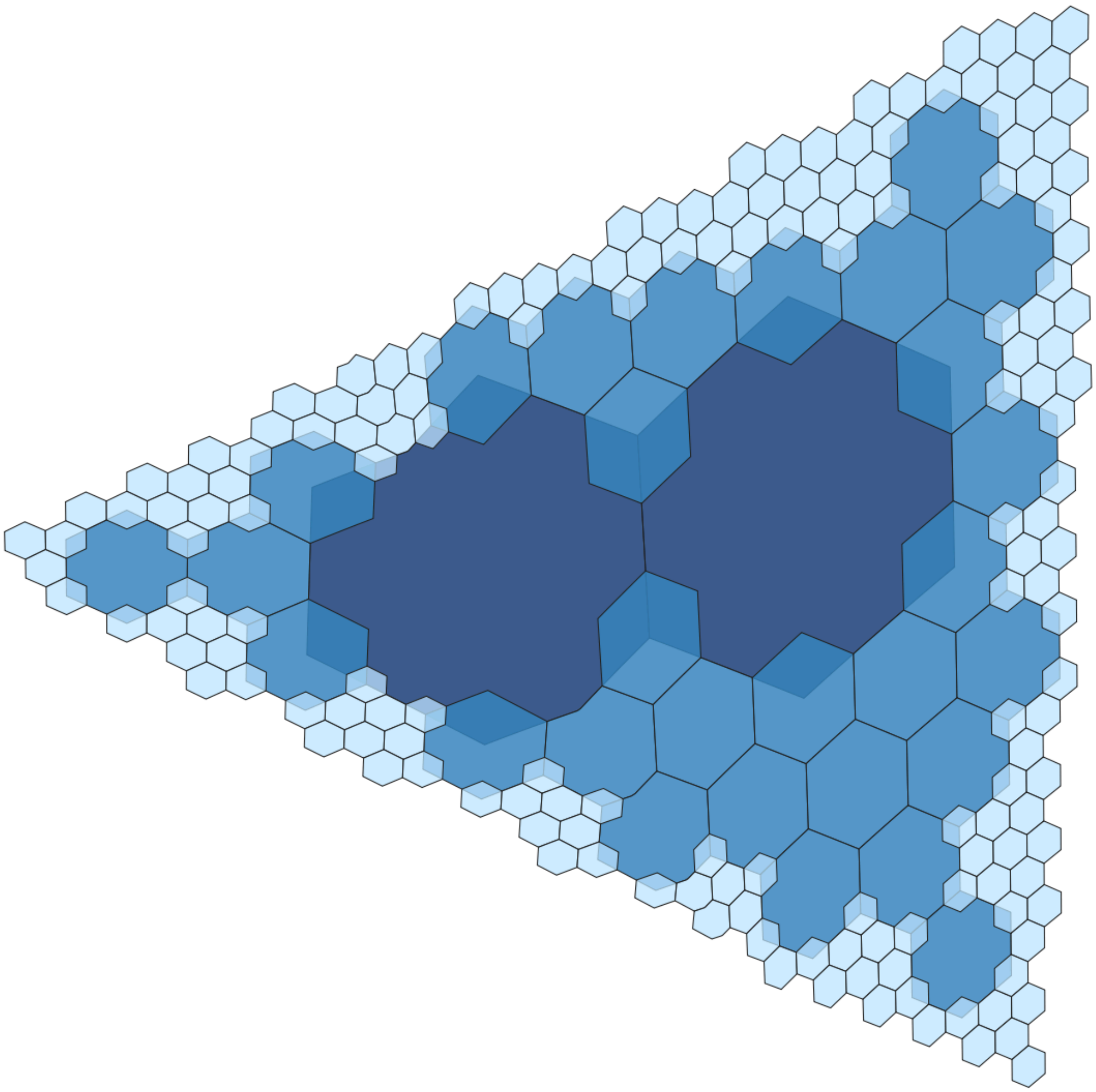


Figure C.36 — GeoJSON compact zones response for querying ISEA3H DGGRS for GEBCO 2014 bathymetry with a CQL2 expression using S_INTERSECTS() for a WKT triangular polygon visualized in QGIS

This spatial filter could also be combined with a relational operator as in the following query for `Elevation < -5500` within the Bermuda Triangle:

https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/isea3h/zones?zone-level=11&filter=Elevation < -5500 and S_INTERSECTS(rec.geom,POLYGON-64.8 32.3,-65.5 18.3,-80.3 25.2,-64.8 32.3)

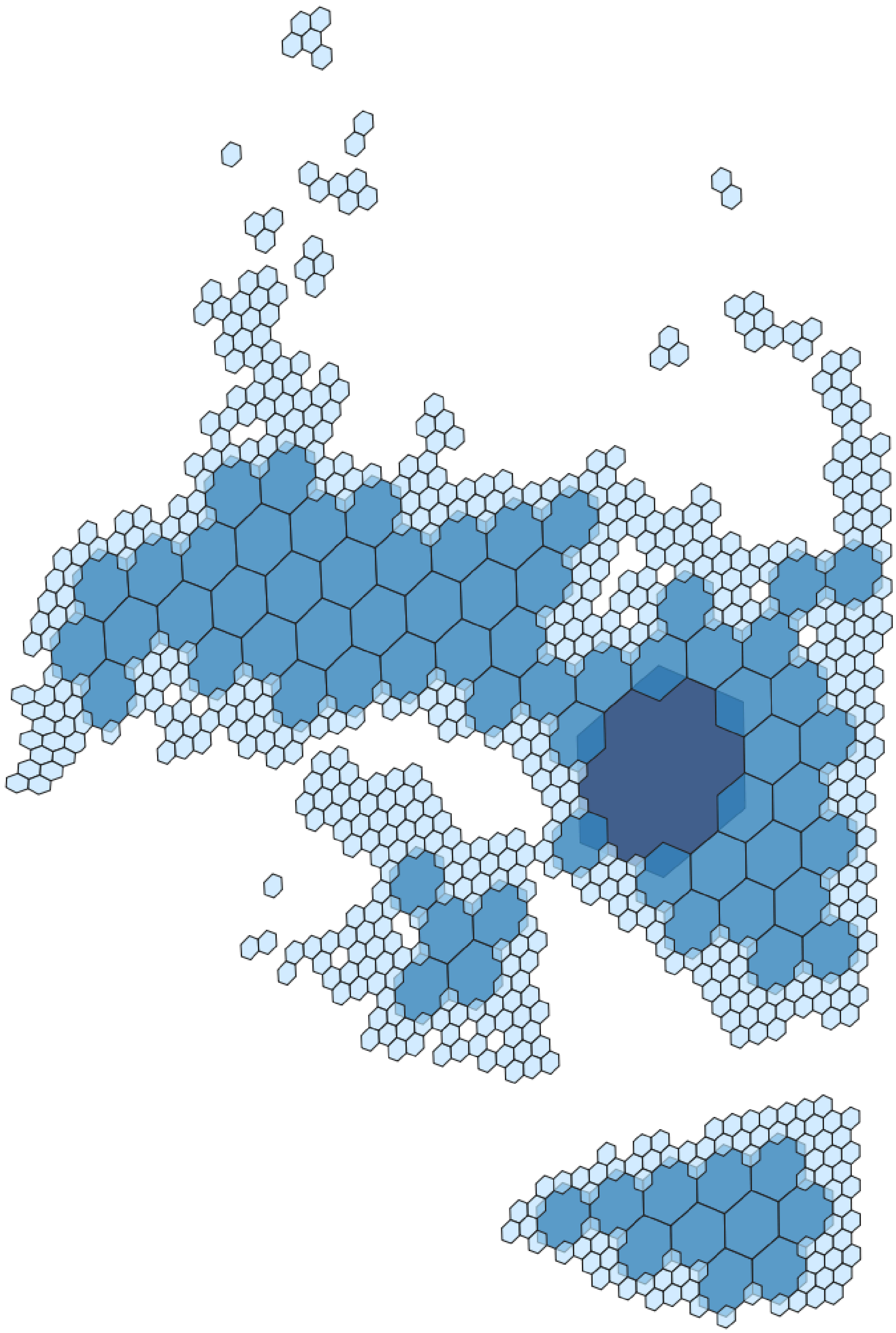


Figure C.37 — GeoJSON compact zones response for querying ISEA3H DGGRS for GEBCO 2014 bathymetry with a CQL2 expression using both `Elevation < -5500` as well as S_INTERSECTS() for a WKT polygon visualized in QGIS

To illustrate the use of CQL2 arithmetic operators, the following example uses a collection of daily climate CMIP5 data from the Copernicus Climate Data Store. The request is for zones where the difference between two different fields of the coverage, the maximum (`tasmax`) and minimum (`tasmin`) daily near-surface air temperature, is greater than 10 degrees Kelvin. In addition, this request uses the `datetime` query parameter to subset the temporal dimension for June 1, 2023.

https://maps.gnosis.earth/ogcapi/collections/climate:cmip5:singlePressure/dggs/isea3h/zones?zone-level=7&filter=(tasmax-tasmin)>10&datetime=2023-06-01

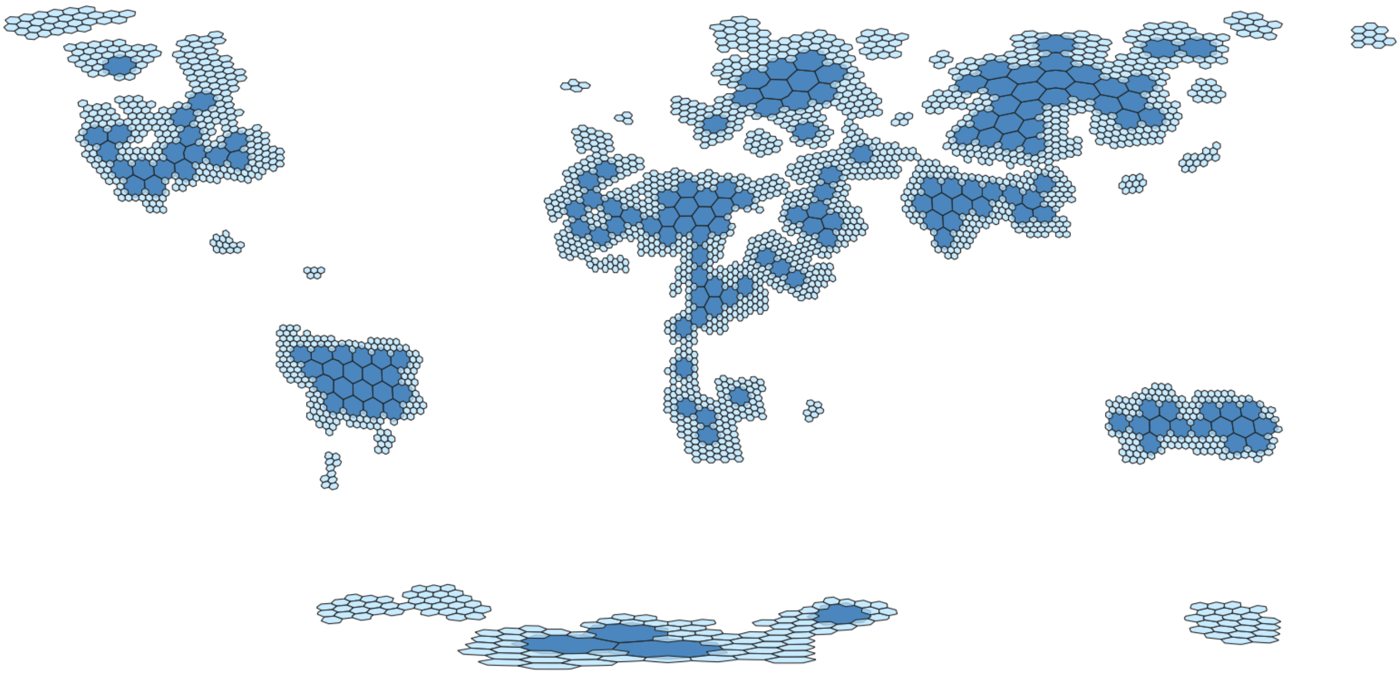


Figure C.38 — GeoJSON compact zones response for querying ISEA3H DGGRS for CMIP5 daily surface temperature from Copernicus Climate Data Store comparing delta with CQL2 expression `(tasmax - tasmin) > 10` visualized in QGIS

CQL2 expressions can also be used for queries using attributes of vector features. The following examples use a collection of building polygon features in Ottawa extracted from the OpenStreetMap dataset, filtering buildings using both a spatial subset and the `name` attribute of the features.

This first example illustrates querying for the National Arts Centre at ISEA3H level 30:

https://maps.gnosis.earth/ogcapi/collections/osm:ottawa:buildings/dggs/ISEA3H/zones?subset=Lat(45.4198:45.4264),Lon(-75.704:-75.6935)&filter=name='National Arts Centre'&zone-level=30

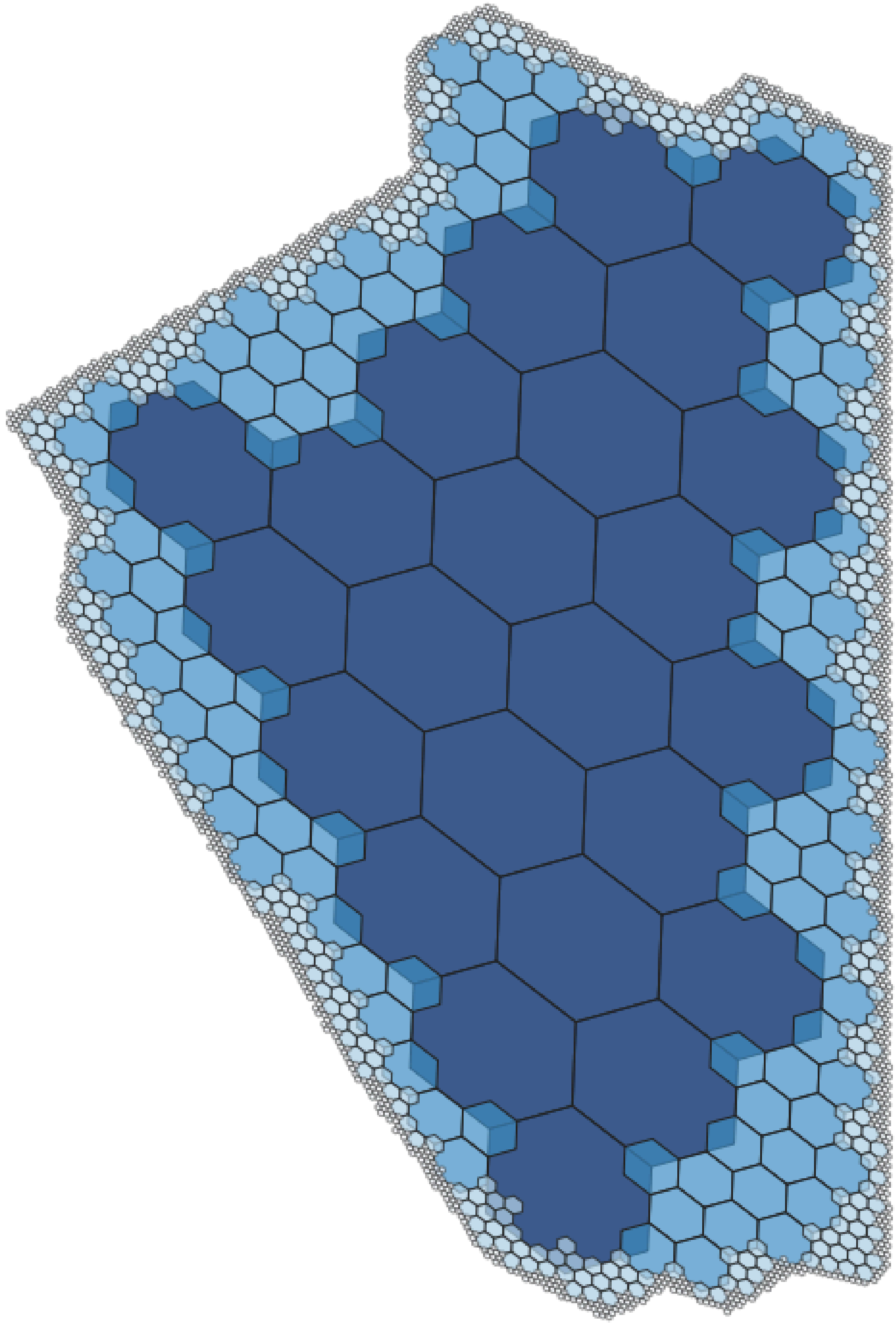


Figure C.39 — GeoJSON compact zones response for querying ISEA3H DGGRS for Ottawa buildings (© OpenStreetMap contributors) with CQL2 expression `name='National Arts Centre'` at level 30, visualized in QGIS

This second example illustrates querying for the East Block at ISEA9R level 15 (corresponding to ISEA3H level 30):

https://maps.gnosis.earth/ogcapi/collections/osm:ottawa:buildings/dggs/ISEA9R/zones?subset=Lat(45.4198:45.4264),Lon(-75.704:-75.6935)&filter=name='East Block'&zone-level=15

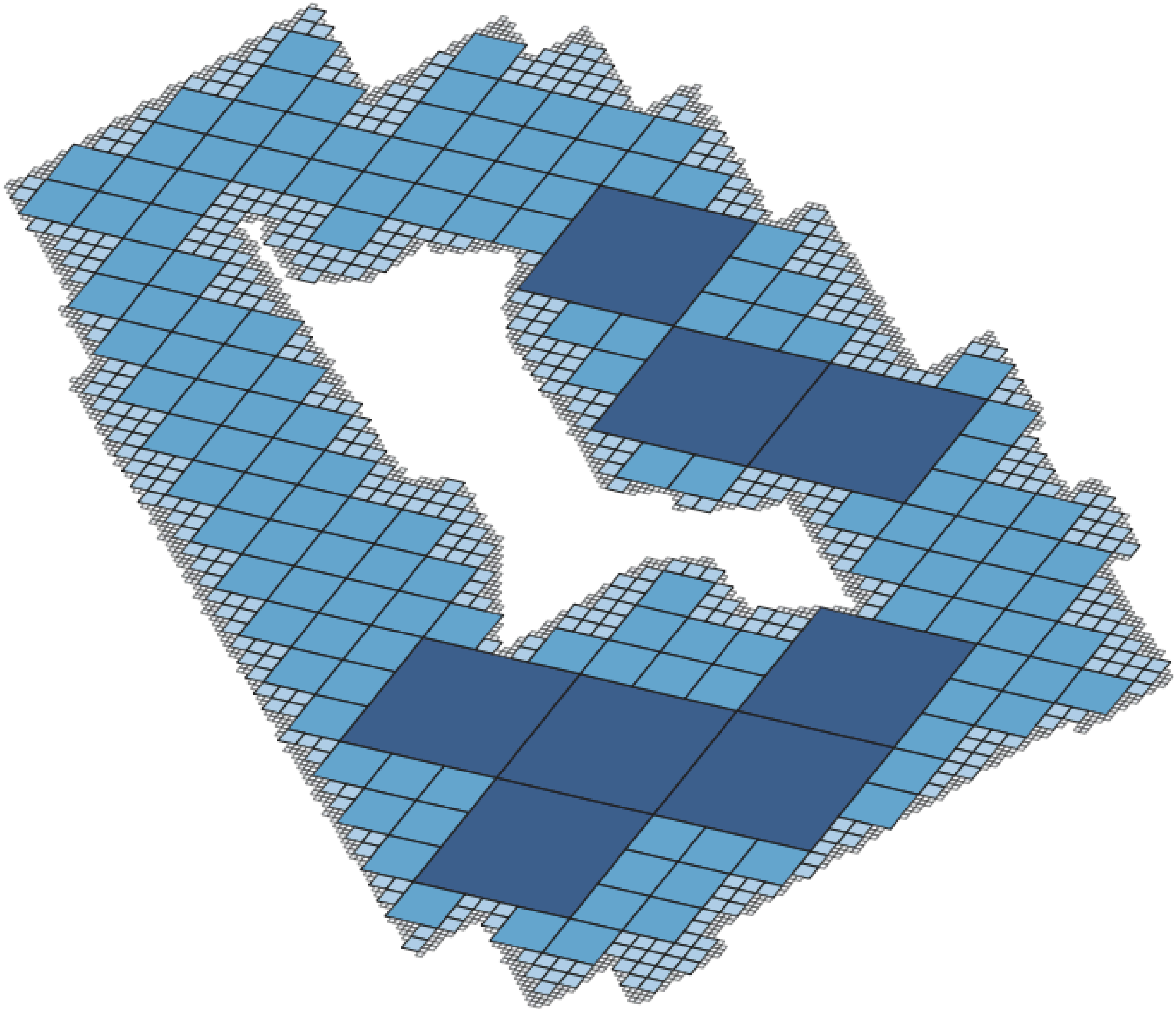


Figure C.40 — GeoJSON compact zones response for querying ISEA9R DGGRS for Ottawa buildings (© OpenStreetMap contributors) with CQL2 expression `name='East Block'` at level 15, visualized in QGIS

This third example illustrates querying for the Centre Block at GNOSIS Global Grid level 24 (corresponding approximately to same resolution):

https://maps.gnosis.earth/ogcapi/collections/osm:ottawa:buildings/dggs/GNOSISGlobalGrid/zones?subset=Lat(45.4198:45.4264),Lon(-75.704:-75.6935)&filter=name='Centre Block'&zone-level=24

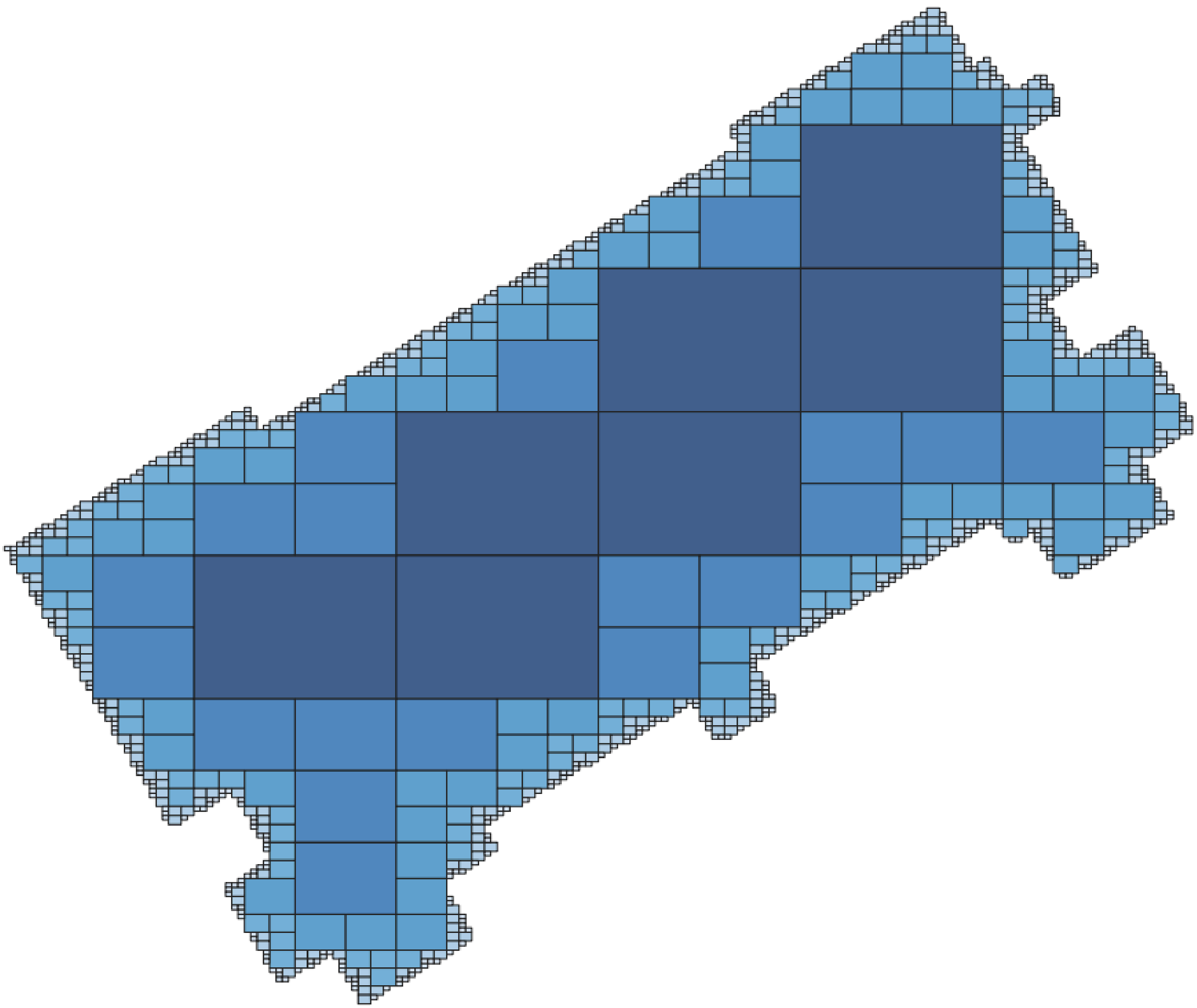


Figure C.41 — GeoJSON compact zones response for querying GNOSIS Global Grid DGGRS for Ottawa buildings (© OpenStreetMap contributors) with CQL2 expression `name='Centre Block'` at level 24, visualized in QGIS

A future extension could also enable cross-collection queries, possibly using a `joinCollections` query parameter which would make queryables from different collections available.

C.8. Retrieving data for a DGGRS Zone

Data can be retrieved for a parent zone of interest, including values for all sub-zones within it, at the …`/dggs/{dggrsId}/zones/{zoneId}/data` resource path.

The following examples illustrate this capability for different DGGRSs, representation encodings, CRSs, and other query parameters.

C.8.1. Retrieving data from an ISEA3H DGGRS

Examples in this section retrieve data for pentagonal and hexagonal zones of an ISEA3H DGGRS, based on the ISEA projection.

This first example retrieves elevation data for the GEBCO collection for ISEA3H zone `A2-0-A` at a relative depth of 8. The default zone depth is 10 for this DGGRS in this particular deployment of a DGGS API, as specified in the `defaultDepth` property of the ISEA3H DGGRS description response). These requests override this default by using the `zone-depth` query parameter defined in the “Data Custom Depths” requirements class.

C.8.1.1. DGGS-JSON Examples

https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA3H/zones/A2-0-A/data?zone-depth=8

A snippet of the DGGS-JSON response follows, including an array of elevation values ordered as per the deterministic zone order defined by this ISEA3H DGGRS.

```
{
   "dggrs" : "https://www.opengis.net/def/dggrs/OGC/1.0/ISEA3H",
   "zoneId" : "A2-0-A",
   "depths" : [ 8 ],
   "values" : {
      "Elevation" : [
         {
            "depth" : 8,
            "shape" : { "count" : 5536, "subZones" : 5536 },
            "data" : [ -14.209265189621, 57.2747655762701, ...,
2075.7176486103044 ]
         }
      ]
   }
}
```

Listing

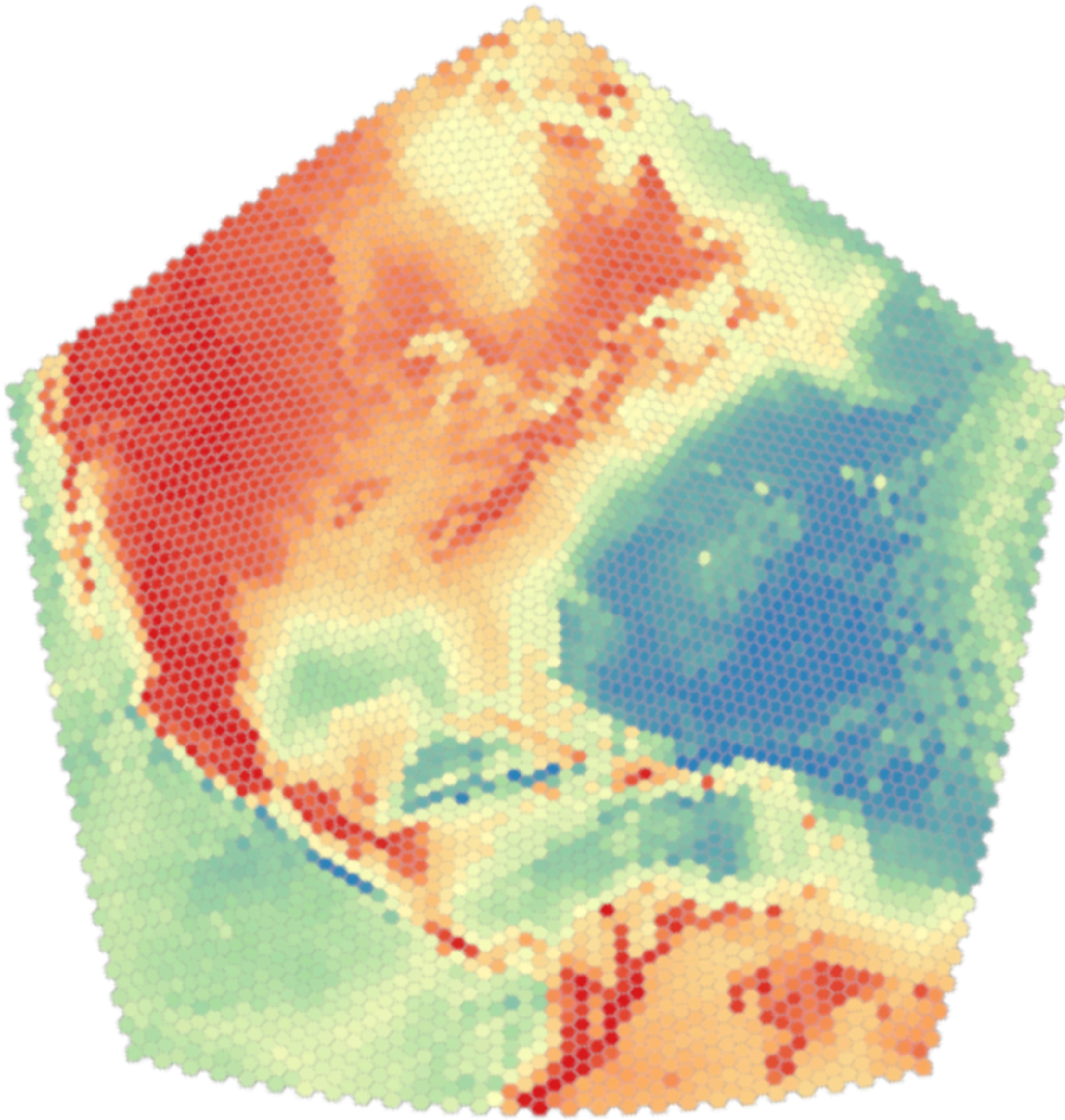


Figure C.42 — Visualization of a DGGS-JSON response for retrieving data for ISEA3H DGGRS zone A2-0-A of GEBCO 2014 bathymetry at a relative depth of 8

C.8.1.2. GeoTIFF Examples

In a GeoTIFF representation, rather than being quantized to sub-zones, the Elevation would be quantized to the closest rectilinear gridded coverage cells, at a resolution high enough to include at least one value per corresponding sub-zones. The data may still be clipped to the shape of the zone geometry. By default, this implementation of the DGGS API Standard returns the data in an OGC:CRS84 CRS.

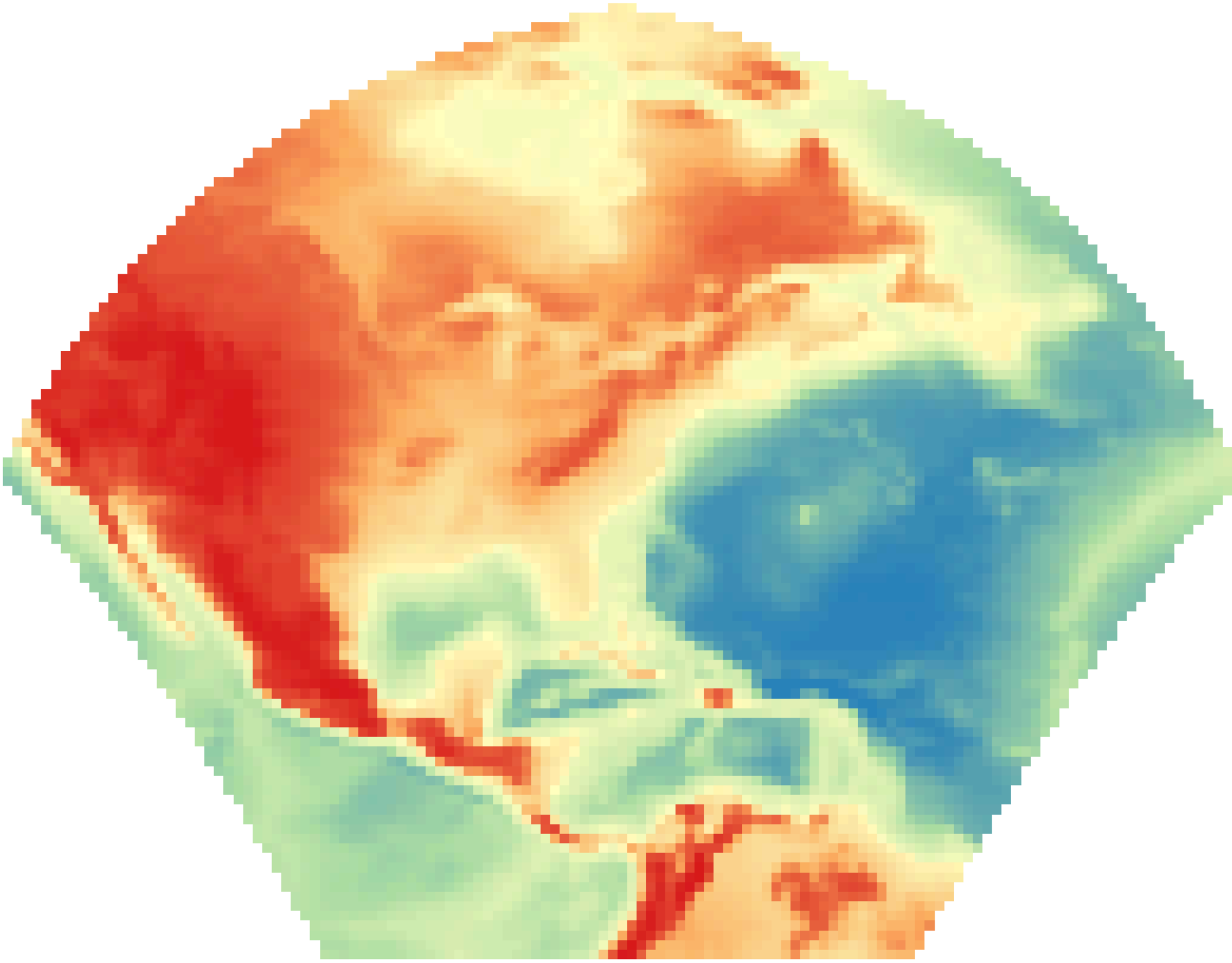


Figure C.43 — Visualization of a CRS84 GeoTIFF response for retrieving data for ISEA3H DGGRS zone A2-0-A of GEBCO 2014 bathymetry at a relative depth of 8

Visualizing this data on 3D globe would look like the following image.

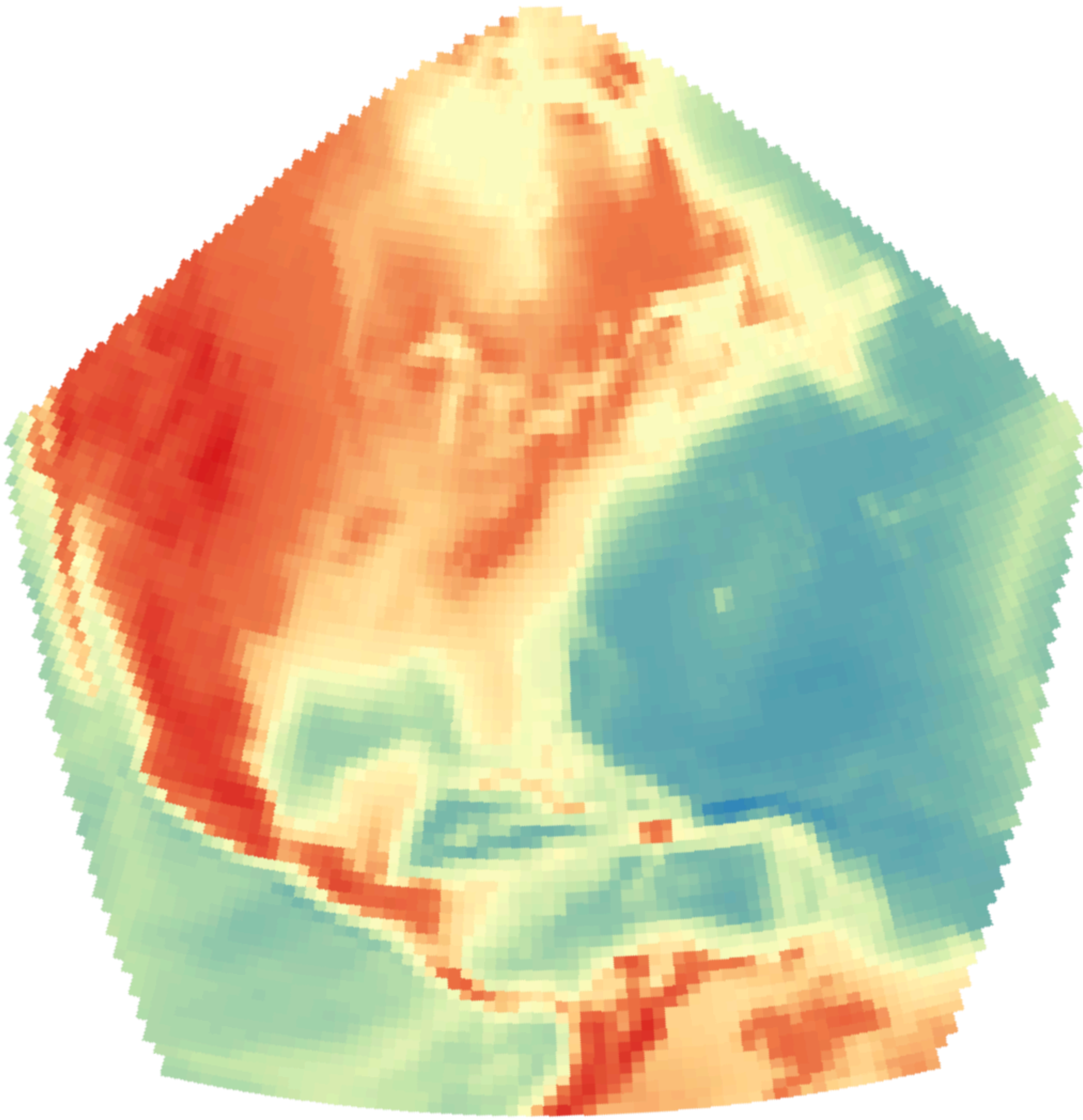


Figure C.44 — Visualization on a 3D Globe of a GeoTIFF response for retrieving data for ISEA3H DGGRS zone A2-0-A of GEBCO 2014 bathymetry at a relative depth of 8

Alternatively, the DGGS API implementation also supports the `crs` query parameter to select the ISEA planar projection as an alternate CRS. In this CRS, this pentagonal zone spans the interruption between two triangles of the base icosahedron.

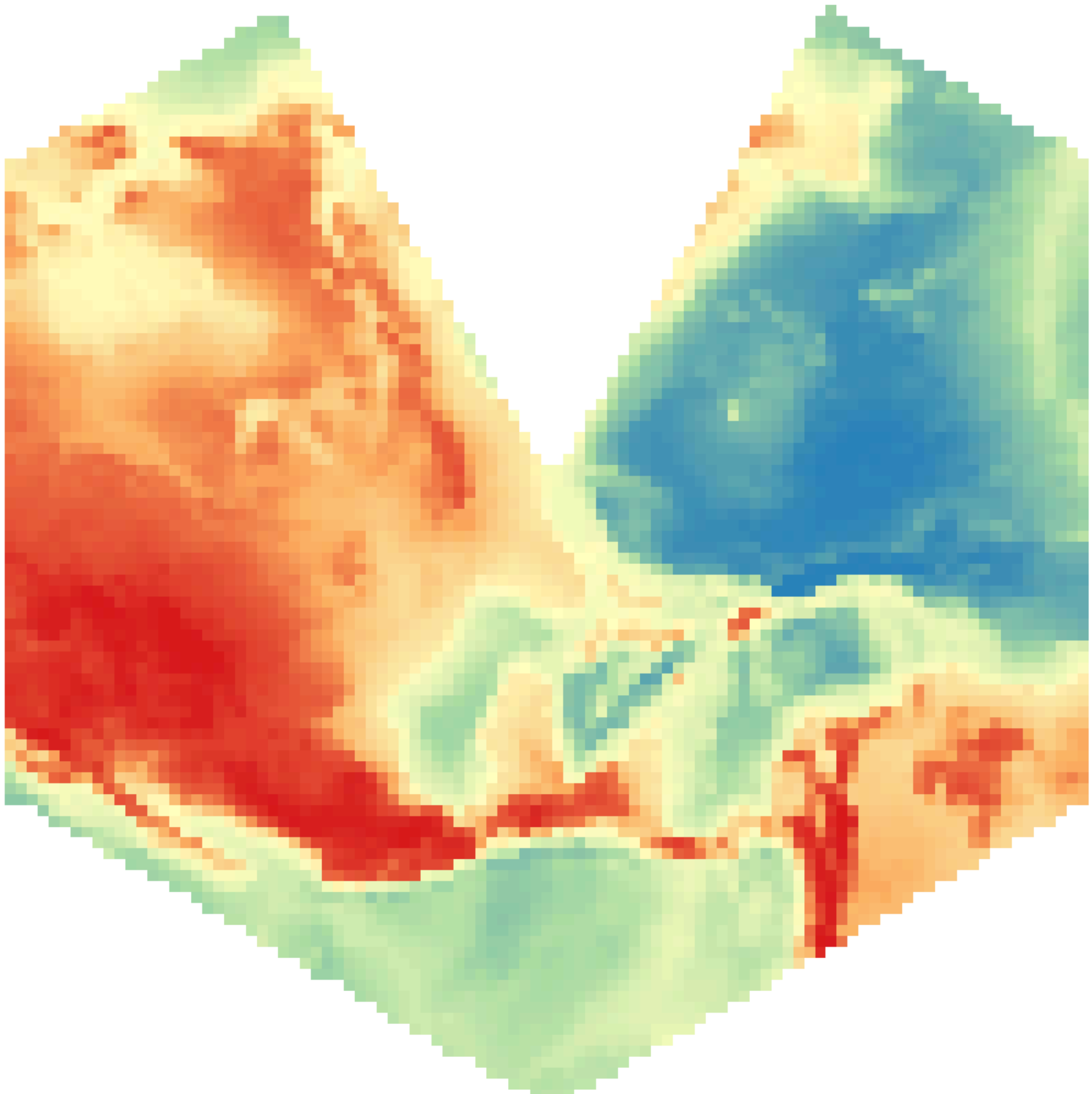


Figure C.45 — Visualization of an ISEA planar projection GeoTIFF response for retrieving data for pentagonal ISEA3H DGGRS zone A2-0-A of GEBCO 2014 bathymetry at a relative depth of 8

A major inconvenience of the ISEA planar projection is that hexagonal zones further along the interruptions will be split on each side on the interruption, resulting in a large separation between the two halves of the data, as illustrated for example with zone C2-4-A.

Figure C.46 — Visualization of an ISEA planar projection GeoTIFF response for retrieving data for ISEA3H DGGRS zone C2-4-A (a hexagon split across an ISEA interruption) of GEBCO 2014 bathymetry

On the other hand, a hexagon not spanning an interruption will appear in the ISEA planar projection as typical hexagons with two edges aligned with either the horizontal (for odd parent zone levels) or vertical axis (for even levels).

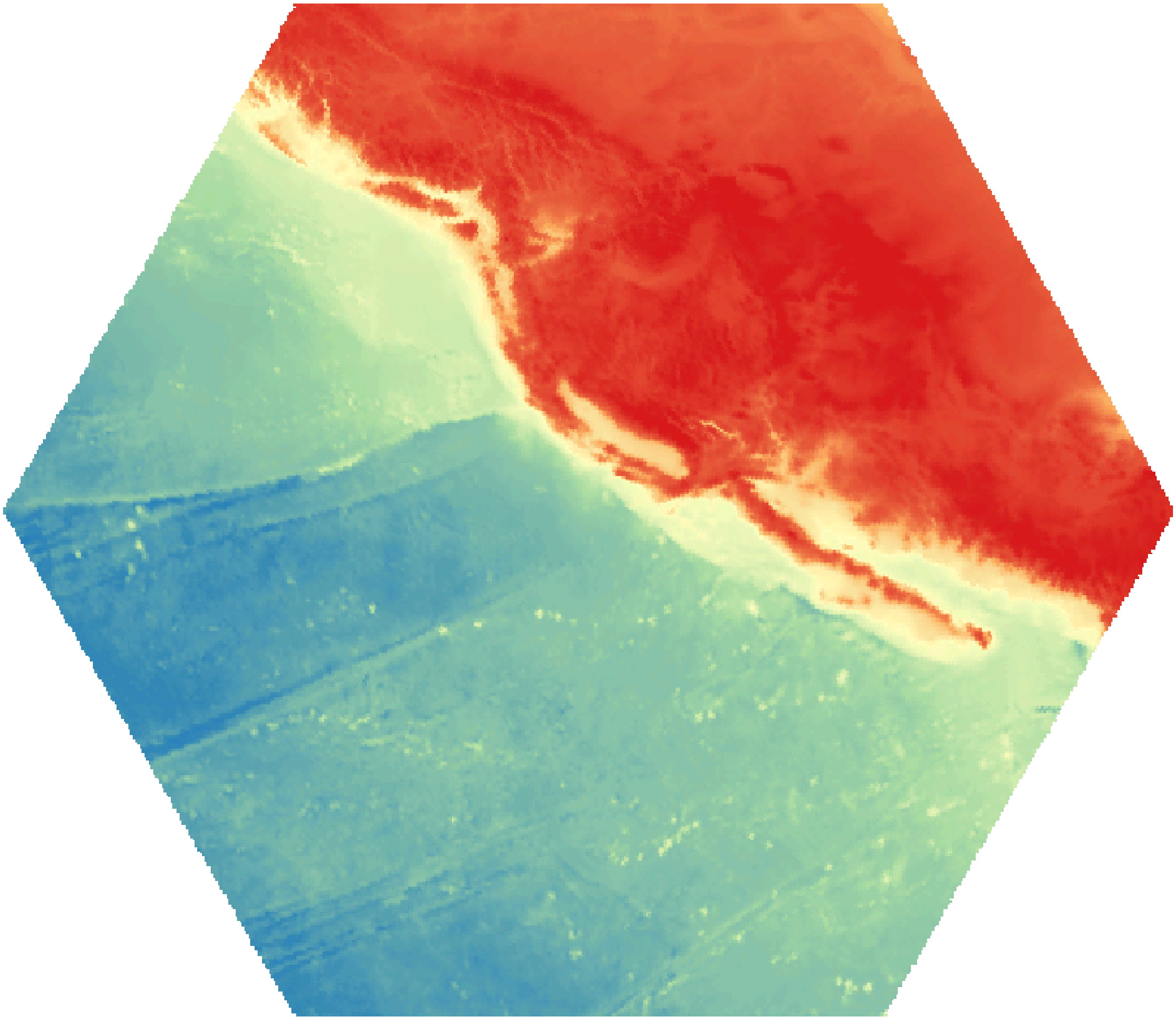


Figure C.47 — Visualization of an ISEA planar projection GeoTIFF response for retrieving data for ISEA3H DGGRS zone A0-0-F of GEBCO 2014 bathymetry

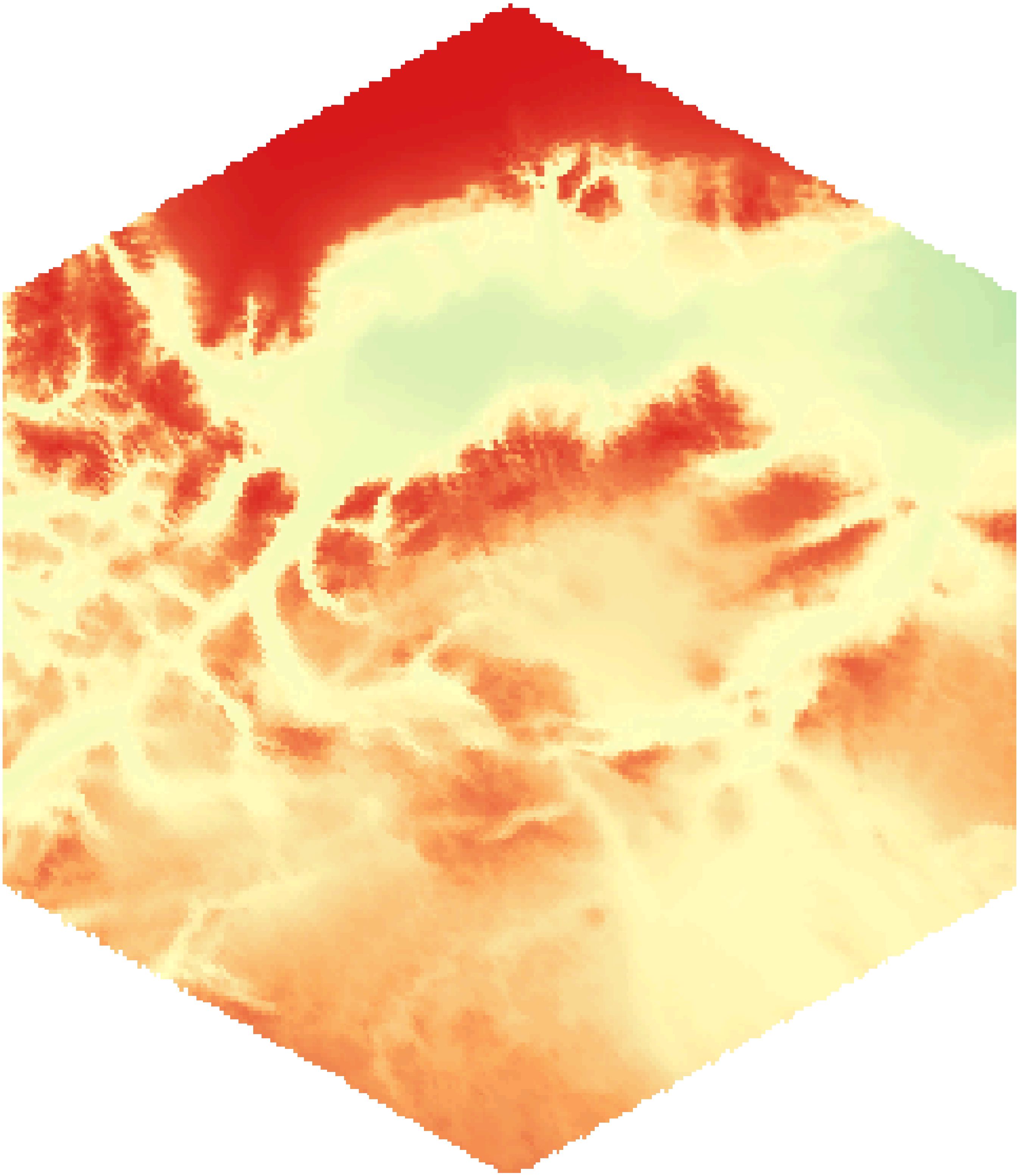


Figure C.48 — Visualization of an ISEA planar projection GeoTIFF response for retrieving data for ISEA3H DGGRS zone B0-5-A of GEBCO 2014 bathymetry whereas the same zones in OGC:CRS84 will have various shapes not easily recognizable as hexagons:

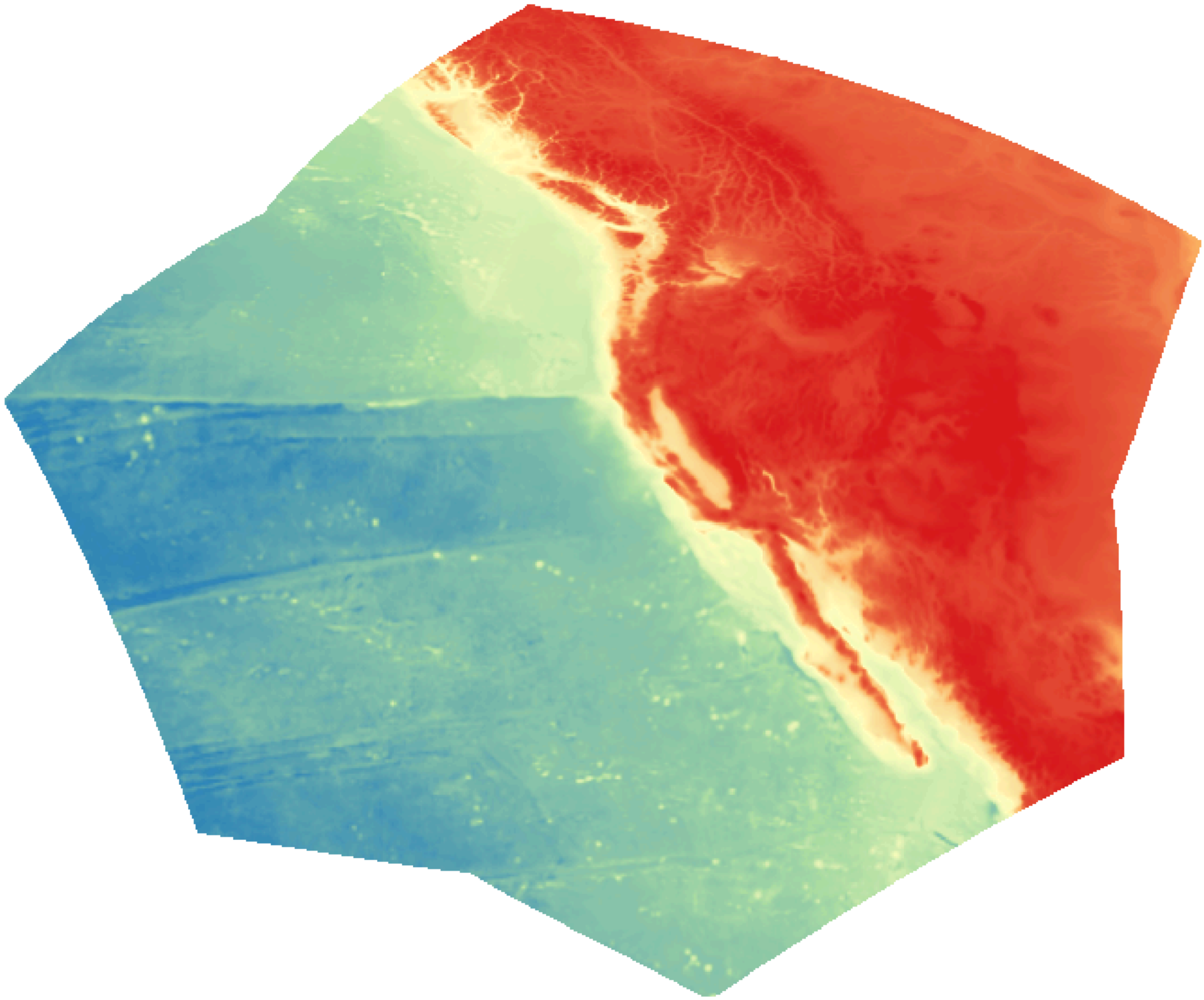


Figure C.49 — Visualization of a GeoTIFF response in OGC:CRS84 for retrieving data for ISEA3H DGGRS zone A0-0-F (odd level 1) of GEBCO 2014 bathymetry

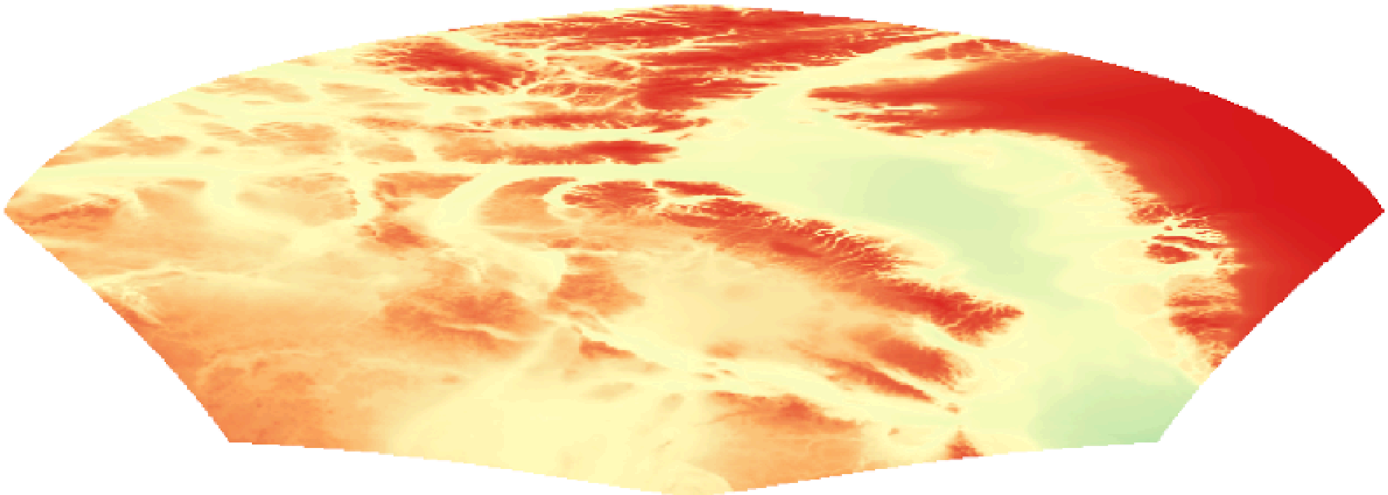


Figure C.50 — Visualization of a GeoTIFF response in OGC:CRS84 for retrieving data for ISEA3H DGGRS zone B0-5-A (even level 2) of GEBCO 2014 bathymetry

While both DGGS-JSON and GeoTIFF are encodings appropriate for raster data, DGGS-JSON is better suited for representing values quantized to the hierarchical discrete global grids. Encoding vector data in these formats essentially implies rasterizing the vector features, using either a selection of the feature properties as values, or numerical feature identifiers for which associated properties may be queryable elsewhere (e.g., in an OGC API — Features deployment).

C.8.1.3. GeoJSON Examples

Although gridded elevation and bathymetry data is better suited to raster formats such as DGGS-JSON and GeoTIFF, a GeoJSON and/or Features Geometry JSON representation could also be offered where each zone is represented as a feature, with the associated property providing the data values such as the elevation. This would be equivalent to the GeoJSON representation for a zone list where `compact-zones` is set to `false`, the `parent-zone` is the zone for which data is requested, and the `zone-level` is the sum of the parent zone plus the relative zone depth at which data is requested (the default depth or `zone-depth` query parameter), with the addition of these properties associating data values with each zone.

If the geometry for these zone features is provided as polygons (`geometry=zone-region`), visualizing this vector data would produce an output like the following partial response, which would result in a visualization identical to Figure C.42.

```
{
   "type" : "FeatureCollection",
   "features" : [ {
      "type" : "Feature",
      "id" : 1,
      "geometry" : {
         "type" : "Polygon",
         "coordinates" : [ [
            [-77.51874470, 68.96298626], [-77.598412, 68.94863327],
            [-77.67832934, 68.93432964], [-77.75852144, 68.92009304],
            [-77.83901433, 68.90594517], [-77.91983505, 68.8919127],
            ...
```

```
            [-77.52048177, 68.99500861], [-77.51874470, 68.96298626]
         ]
      },
      "properties" : {
         "Elevation" : -14.209265189621,
         "zoneID" : "E0-8C1-A"
      }
   }, {
      "type" : "Feature",
      "id" : 2,
      ...
```

Listing

Alternatively, the geometry of each zone could also be provided as a single point for its centroid (`geometry=zone-centroid`), in which case the response and associated visualization would look like the following:

```
{
   "type" : "FeatureCollection",
   "features" : [ {
      "type" : "Feature",
      "id" : 1,
      "geometry" : {
         "type" : "Point",
         "coordinates" : [-78.7500000030306, 69.2228046463321]
      },
      "properties" : {
         "Elevation" : -14.209265189621,
         "zoneID" : "E0-8C1-A"
      }
   }, {
      "type" : "Feature",
      "id" : 2,
      "geometry" : {
         "type" : "Point",
         "coordinates" : [-82.3566944995815, 68.4176458502864]
      },
      "properties" : {
         "Elevation" : 57.2747655762701,
         "zoneID" : "E0-911-A"
      }
   },
   ...
```

Listing

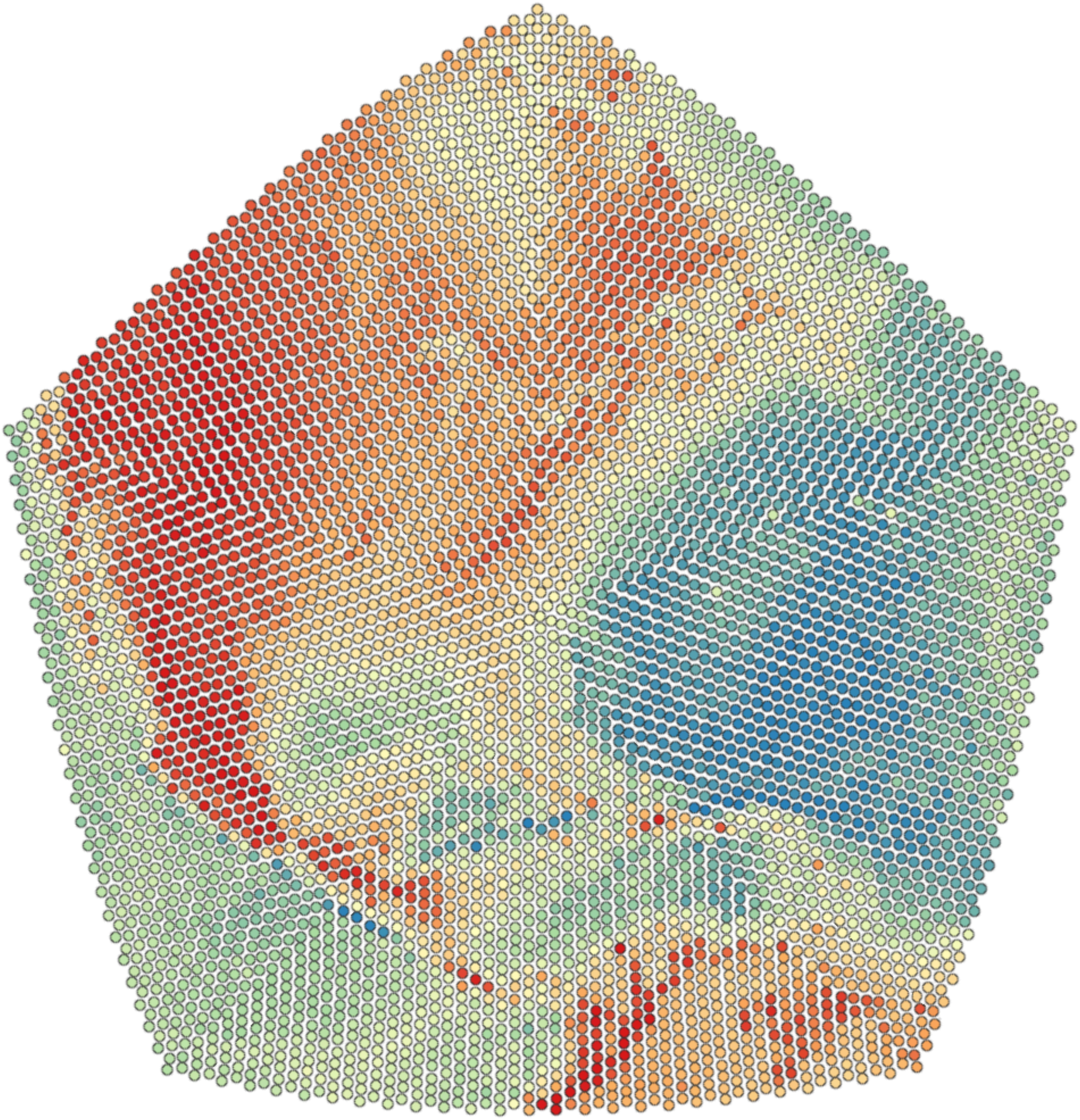


Figure C.51 — Visualization of a GeoJSON point features response for data retrieved from ISEA3H DGGRS zone A2-0-A of GEBCO 2014 bathymetry at a relative depth of 8

For data of a vector nature, a vector format such as GeoJSON allows to encode zone data response in a more compact manner without having to rasterize the data. Ideally, especially for features spanning a large spatial extent (such as country boundaries, or the coastline of a large landmass), the features would be clipped to the geometry of the zone for which data is being requested, or as least to limited extent around the zone. Unfortunately, GeoJSON and JSON-FG do not yet have a standard mechanism to identify artificial edges which may result from such clipping. This complicates re-joining features spread across multiple zones, and may result in rendering artifacts for polygonal features when rendered with a stroke, as the stroke would also be drawn on those artificial segments at the bound edges.

The following response for a zone data request and its visualization illustrates vector features in Ottawa from OpenStreetMap returned as GeoJSON. The features and geometry coordinates in this case have no relation to sub-zones, and are only clipped to the zone for which data is being requested. The zone depth (default zone depth of 10 in this case) can however control cartographic generalization of the geometry. This could potentially include filtering point features, based on rules associated with the dataset, as decided by the implementation, as well as the precision of the coordinates.

https://maps.gnosis.earth/ogcapi/collections/osm:ottawa:roads/dggs/ISEA3H/zones/I0-1EEBFA7-E/data


```
{
   "type" : "FeatureCollection",
   "features" : [ {
      "type" : "Feature",
      "id" : 2305843009218664049,
      "geometry" : {
         "type" : "LineString",
         "coordinates" : [
            [-75.70190343, 45.42180362],
            [-75.70214786, 45.4220908],
            [-75.70255148, 45.42257149]
         ]
      },
      "properties" : {
         "highway" : "service",
         "name" : "Parliament Road",
         "name:fr" : "chemin Parliament",
         "access" : "no"
      }
   }, {
      "type" : "Feature",
      "id" : 2305843009218664071,
      "geometry" : {
         "type" : "LineString",
         "coordinates" :
   ...
```


Listing

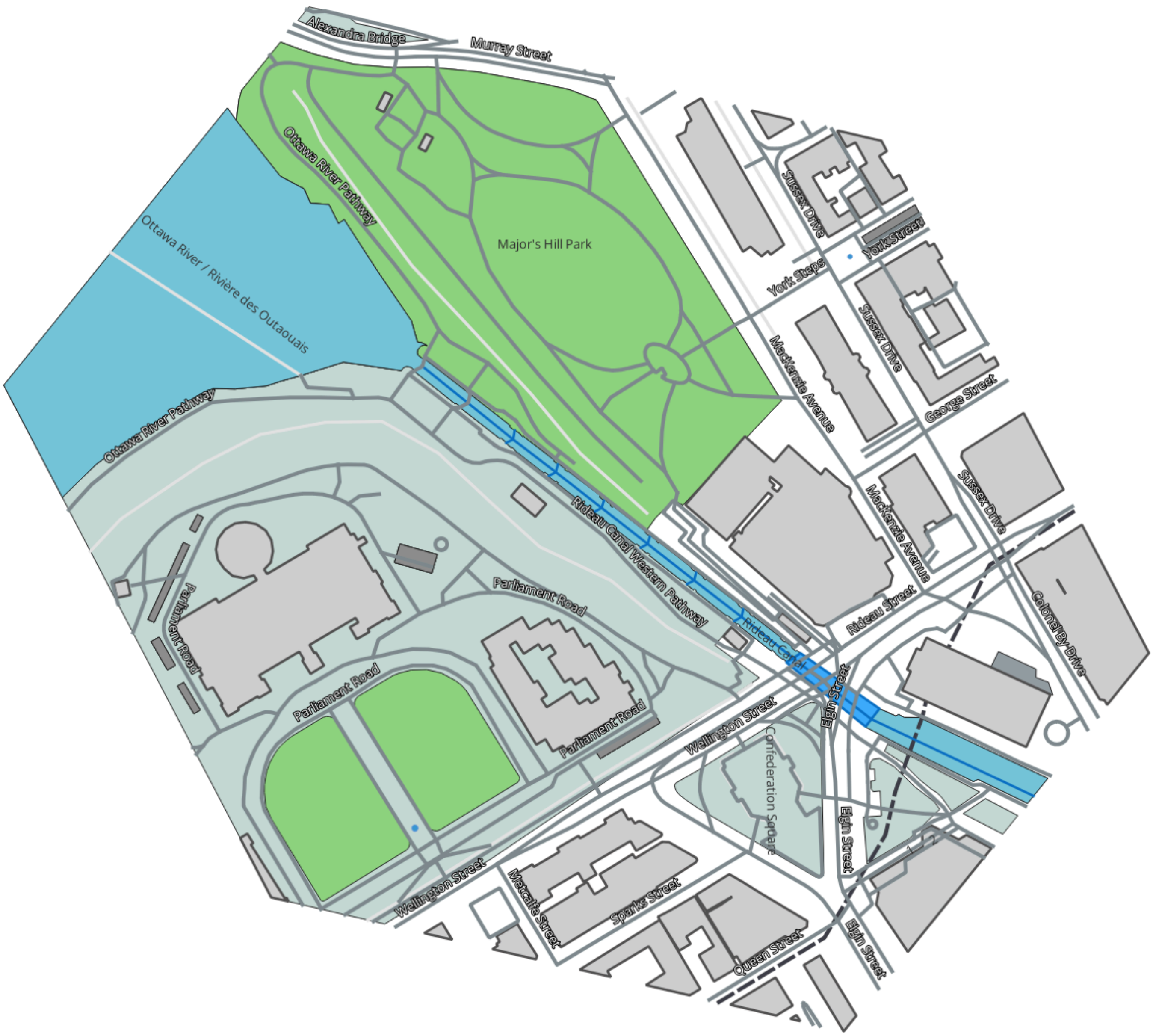


Figure C.52 — Visualization in a world Mercator projection of a GeoJSON feature collection with mixed geometry types for data retrieved from ISEA3H DGGRS zone I0-1EEBFA7-E of OpenStreetMap (c) OpenStreetMap contributors

In an ISEA planar projection, the same data could be visualized as follows.

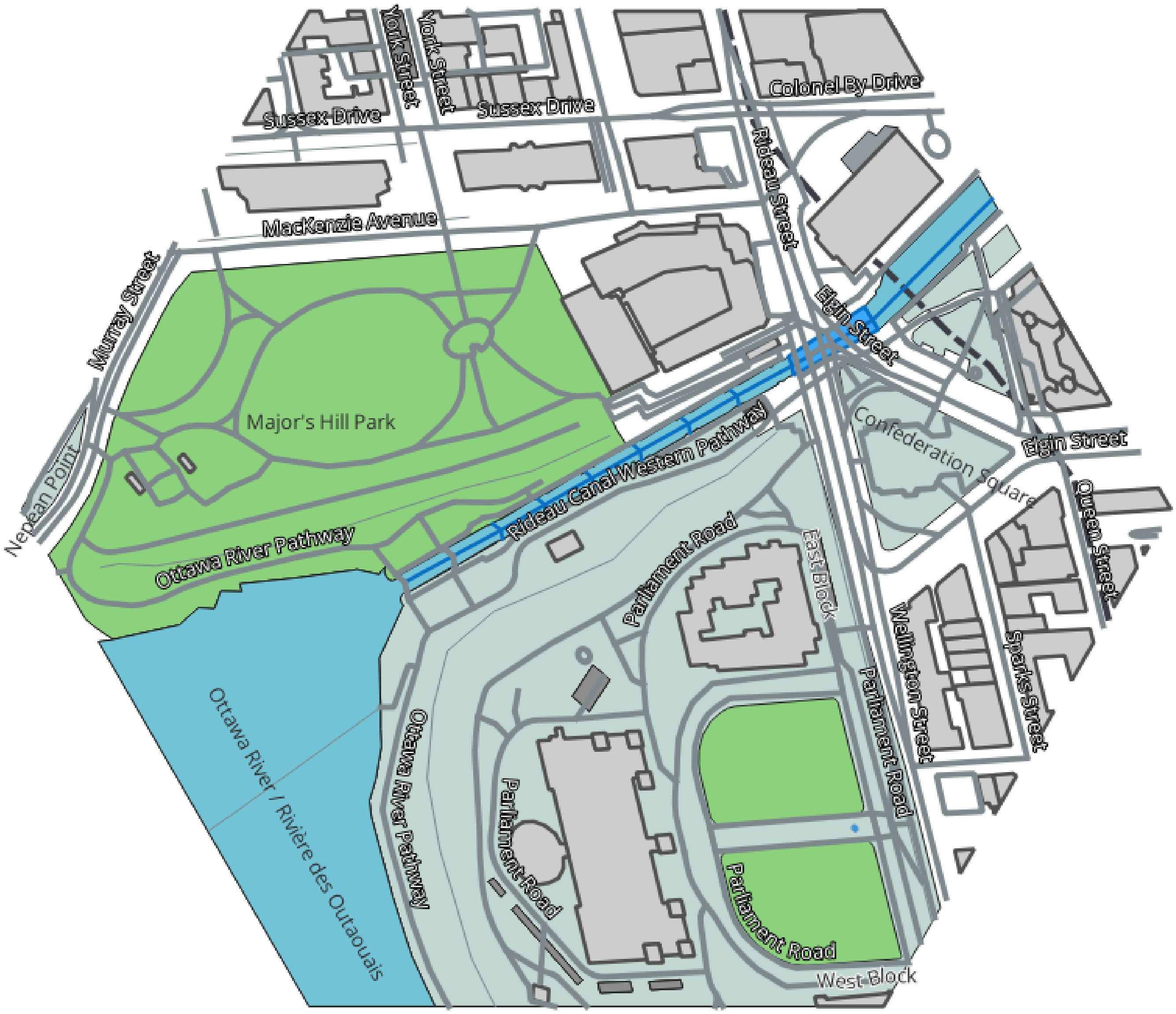


Figure C.53 — Visualization in an ISEA planar projection of a GeoJSON feature collection with mixed geometry types for data retrieved from ISEA3H DGGRS zone I0-1EEBFA7-E of OpenStreetMap (c) OpenStreetMap contributors

The following zone data request is for bathymetry polygons from the Natural Earth physical features.

https://maps.gnosis.earth/ogcapi/collections/NaturalEarth:physical:bathymetry:ne_10m_bathymetry_L_0/dggs/ISEA3H/zones/B0-5-A/data


```
{
   "type": "FeatureCollection",
   "features": [
      {
         "type": "Feature",
         "id": 1,
         "properties": {
```

```
          "scalerank": 0,
          "featurecla": "Bathymetry"
        },
        "geometry": {
          "type": "MultiPolygon",
          "coordinates": [
            [
              [ [ -55.63744954, 59.87006944 ], [ -56.91595155, 59.7924323
],
                [ -58.19026295, 59.70335469 ], [ -59.45970834, 59.60311811
],
                [ -60.72365746, 59.49204741 ], [ -61.98152851, 59.37051649
],
                ...
```


Listing

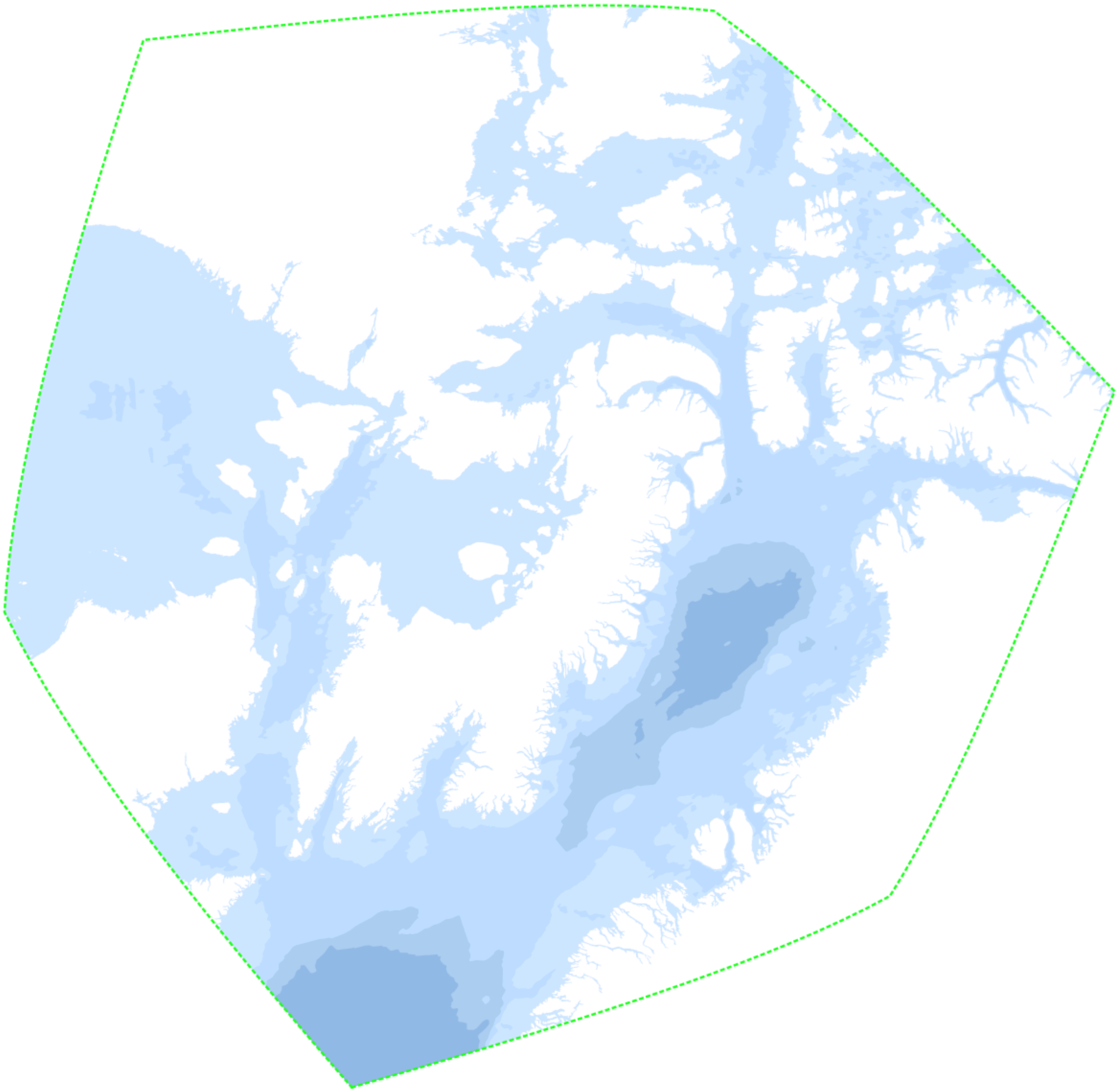


Figure C.54 — Visualization in a North polar projection of GeoJSON polygon contour features for data retrieved from ISEA3H DGGRS zone B0-5-A of Natural Earth bathymetry contours.

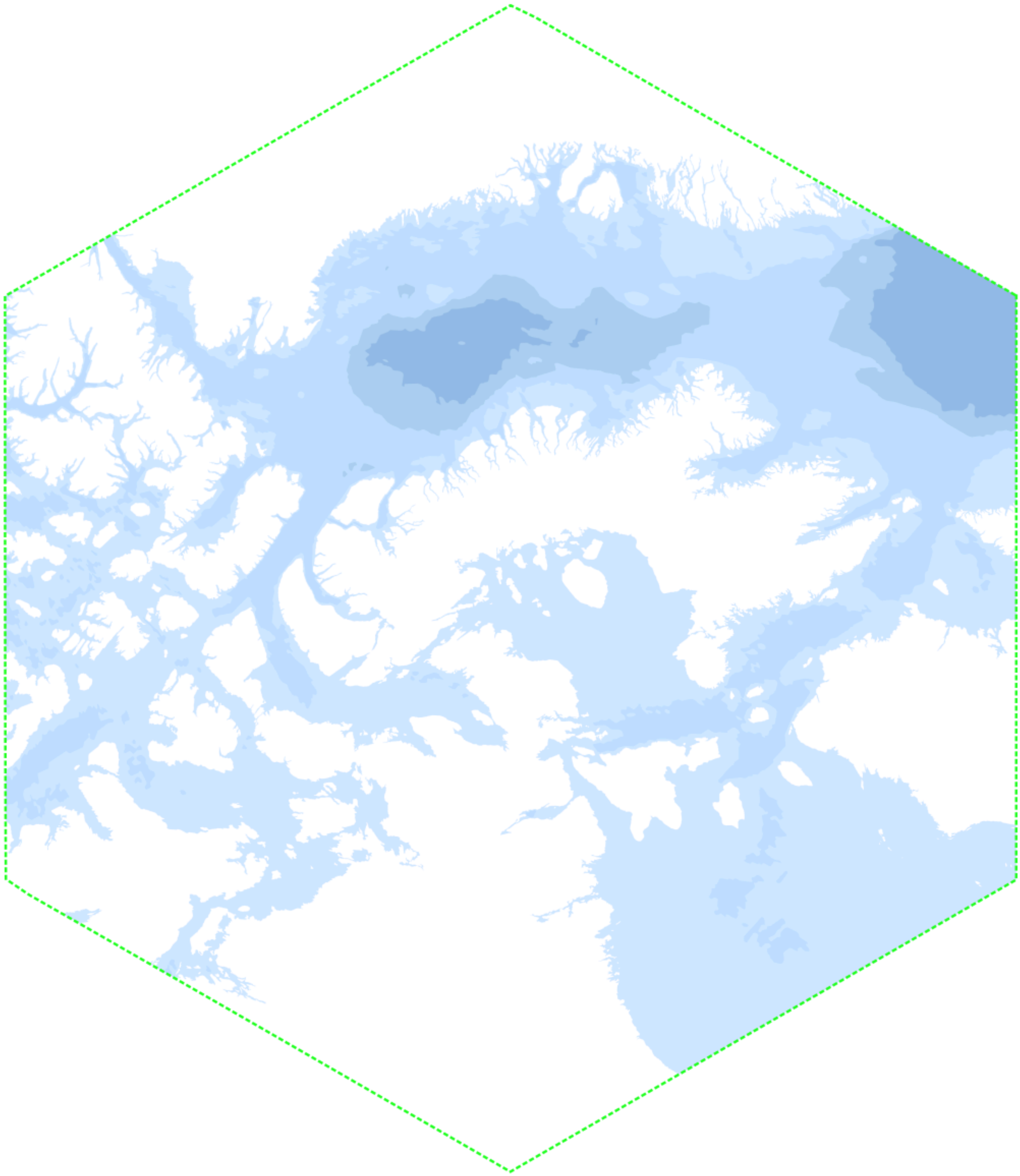


Figure C.55 — Visualization in an ISEA planar projection of GeoJSON polygon contour features for data retrieved from ISEA3H DGGRS zone B0-5-A of Natural Earth bathymetry contours.

C.8.1.4. DGGS-JSON-FG Examples

The DGGS-JSON-FG format is the vector counterpart to DGGS-JSON, leveraging the same concepts of DGGRS, parent zone, sub-zones at a given relative depth, and deterministic sub-zone order, while also building on top of the familiar Features Geometry JSON and GeoJSON formats, which are themselves built on OGC Simple Features.

In DGGS-JSON-FG, spatial coordinates of the feature geometries corresponding to the spatial dimensions of the DGGRS are encoded as integer sub-zone order indices ranging from 1 to the number of sub-zones at the selected relative depth. This relative depth corresponds to the precision of the coordinates at which geometry is being defined, analogous to the number of decimals that would be used in GeoJSON or JSON-FG. Because this essentially corresponds to a coordinate system local to the zone for which data is being retrieved, high precision can be achieved with much fewer information than is required to address the entire globe.

This representation also addresses the issue of artificial segments. This is done by using a special zone order index value of *0* for representing coordinates outside of the zone, which facilitates merging geometry across adjacent zones and easily identifying artificial segments of polygon contours. For polygons, rather than tracing points along the clipping zone edge(s), only the exit and entry points will exist with a single *0* coordinate value in between, indicating that the portion of the zone edge(s) between those two points is within the contour. A DGGS-JSON-FG representation is requested using the `application/geo+json` media type together with the `profile` query parameter set to a value of `jsonfg-dggs`, `jsonfg-dggs-plus`, `jsonfg-dggs-zoneids` or `jsonfg-dggs-zoneids-plus`.

The DGGS-JSON-FG representation for the previous OpenStreetMap example extract follows, at a relative depth precision of 15, resulting in coordinates within a ~275 cm² region, ~16 cm precision, with sub-zone indices between 1 and 14,355,469 (roughly equivalent to latitude and longitude with six decimal places).

```
{
   "conformsTo": [
      "https://www.opengis.net/spec/json-fg-1/0.2/conf/core",
      "https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-dggs-fgjson"
   ],
   "links": [
      {
         "rel" : "profile",
         "href" : "https://www.opengis.net/def/profile/ogc/0/jsonfg-dggs"
      }
   ],
   "dggrs": "[ogc-dggrs:ISEA3H]",
   "zoneId": "I0-1EFA652-D",
   "depth": "15",
   "type" : "FeatureCollection",
   "features" : [ {
      "type" : "Feature",
      "id" : 2305843009218664049,
      "geometry" : null,
      "place" : null,
      "time" : null,
      "dggsPlace" : {
         "type" : "LineString",
         "coordinates" : [ 13866054, 13899376, 13953688 ]
```

```
      },
      "properties" : {
         "highway" : "service",
         "name" : "Parliament Road",
         "name:fr" : "chemin Parliament",
         "access" : "no"
      }
   }, {
      "type" : "Feature",
      "id" : 2305843009218664071,
      "geometry" : null,
      "place" : null,
      "time" : null,
      "dggsPlace" : {
         "type" : "LineString",
         "coordinates" :
   ...
```

Listing

The DGGS-JSON-FG representation for the previous arctic bathymetry polygons example extract follows, at a relative depth precision of 15, resulting in coordinates within a ~0.39 km² region (the size of the previous OpenStreetMap example level 17 zone), ~627 m precision, with sub-zone indices between 1 and 14,355,469 (equivalent to decimal latitude and longitude between two and three decimal places).

```
{
   "conformsTo": [
      "https://www.opengis.net/spec/json-fg-1/0.2/conf/core",
      "https://www.opengis.net/spec/ogcapi-dggs-1/1.0/conf/data-dggs-fgjson"
   ],
   "links": [
      {
         "rel" : "profile",
         "href" : "https://www.opengis.net/def/profile/ogc/0/jsonfg-dggs"
      }
   ],
   "dggrs": "[ogc-dggrs:ISEA3H]",
   "zoneId": "B0-5-A",
   "depth": "15",
   "geometryDimension": 2,
   "type": "FeatureCollection",
   "features": [
      {
         "type": "Feature",
         "id": 1,
         "properties": {
            "scalerank": 0,
            "featurecla": "Bathymetry"
         },
         "geometry" : null,
         "place": null,
         "time" : null,
         "dggsPlace": {
            "type": "MultiPolygon",
            "coordinates": [
               [
                  [ 14355360, 14355250, 14355141, 14355032, 14354922, 14354813,
 ...
```

Listing

C.8.2. Retrieving data from an ISEA9R DGGRS

Examples in this section retrieve data for rhombic zones of an ISEA9R DGGRS, based on the ISEA projection.

C.8.2.1. DGGS-JSON Examples

The following request is for data from the A2-0 level 0 rhombic ISEA9R zone at a relative depth of 4 in DGGS-JSON:

https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA9R/zones/A2-0/data?zone-depth=4


```
{
   "dggrs" : "https://www.opengis.net/def/dggrs/OGC/1.0/ISEA9R",
   "zoneId" : "A2-0",
   "depths" : [ 4 ],
   "values" : {
      "Elevation" : [
         {
            "depth" : 4,
            "shape" : { "count" : 6561, "subZones" : 6561 },
            "data" : [ -684.1801215563401, -725.808053553661, ..., -
4279.7074446855995 ]
         }
      ]
   }
}
```


Listing

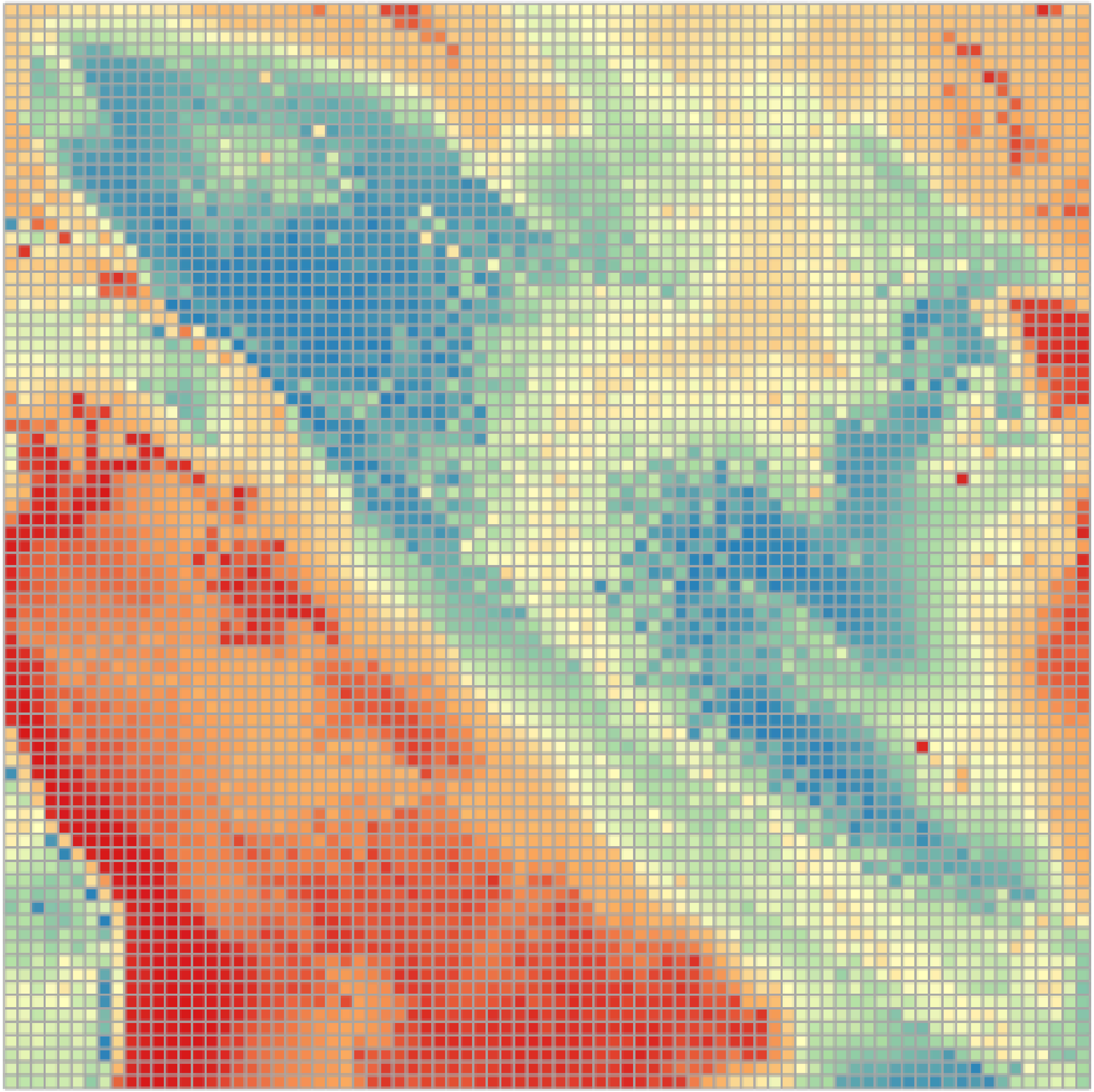


Figure C.56 — Visualization in the ISEA 5x6 rotated and sheared space of a DGGS-JSON response for retrieving data for ISEA9R DGGRS zone A2-0 of GEBCO 2014 bathymetry at a relative depth of 4

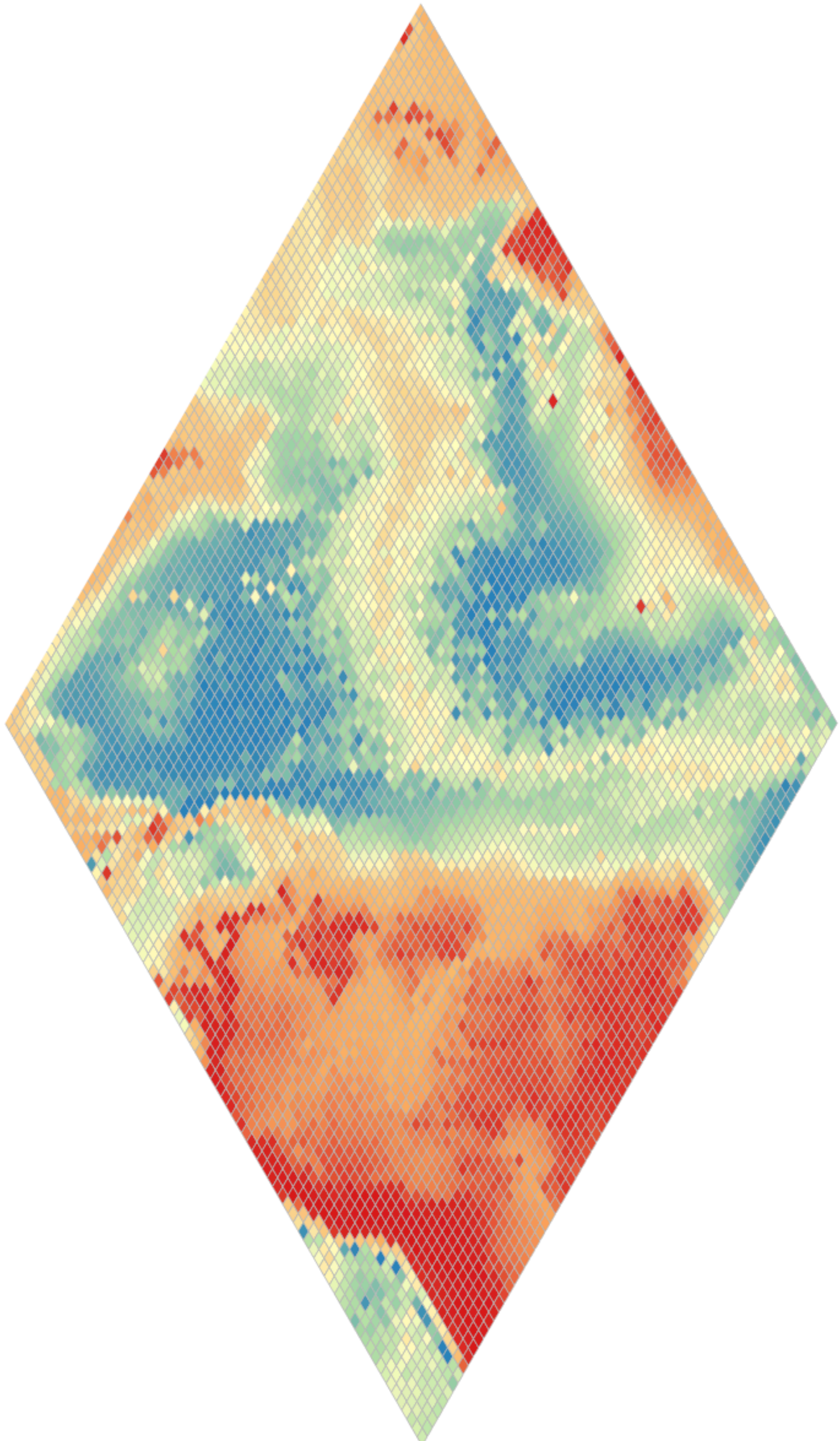


Figure C.57 — Visualization in an ISEA planar projection of a DGGS-JSON response for retrieving data for ISEA9R DGGRS zone A2-0 of GEBCO 2014 bathymetry at a relative depth of 4

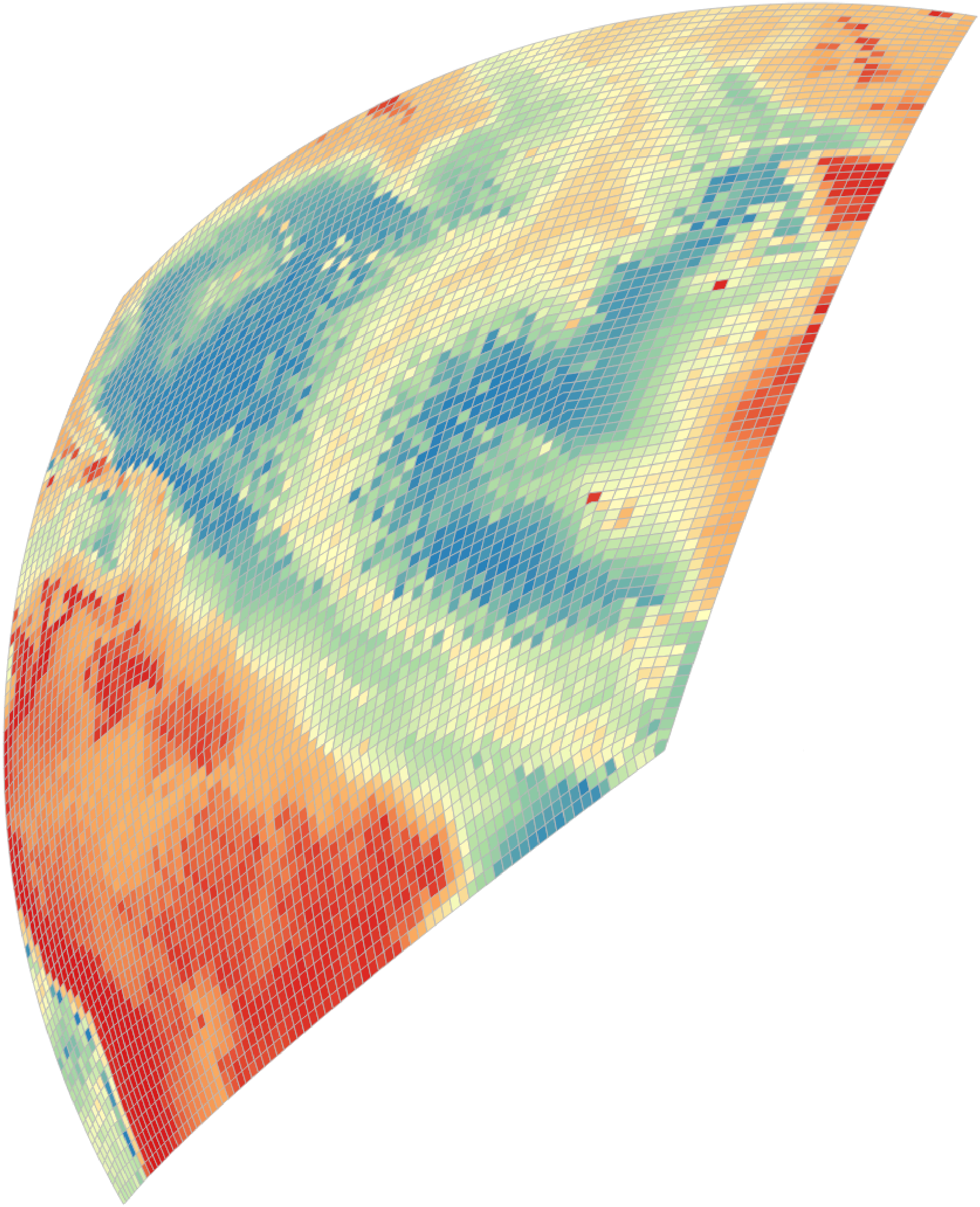


Figure C.58 — Visualization in an orthographic projection of a DGGS-JSON response for retrieving data for ISEA9R DGGRS zone A2-0 of GEBCO 2014 bathymetry at a relative depth of 4

C.8.2.2. GeoTIFF Examples

The following illustrates the GeoTIFF response for requesting data from the same A2-0 ISEA9R zone at same relative depth of 4 in three different CRSs: 5×6 ISEA rotated and sheared space, ISEA planar projection, and CRS84.

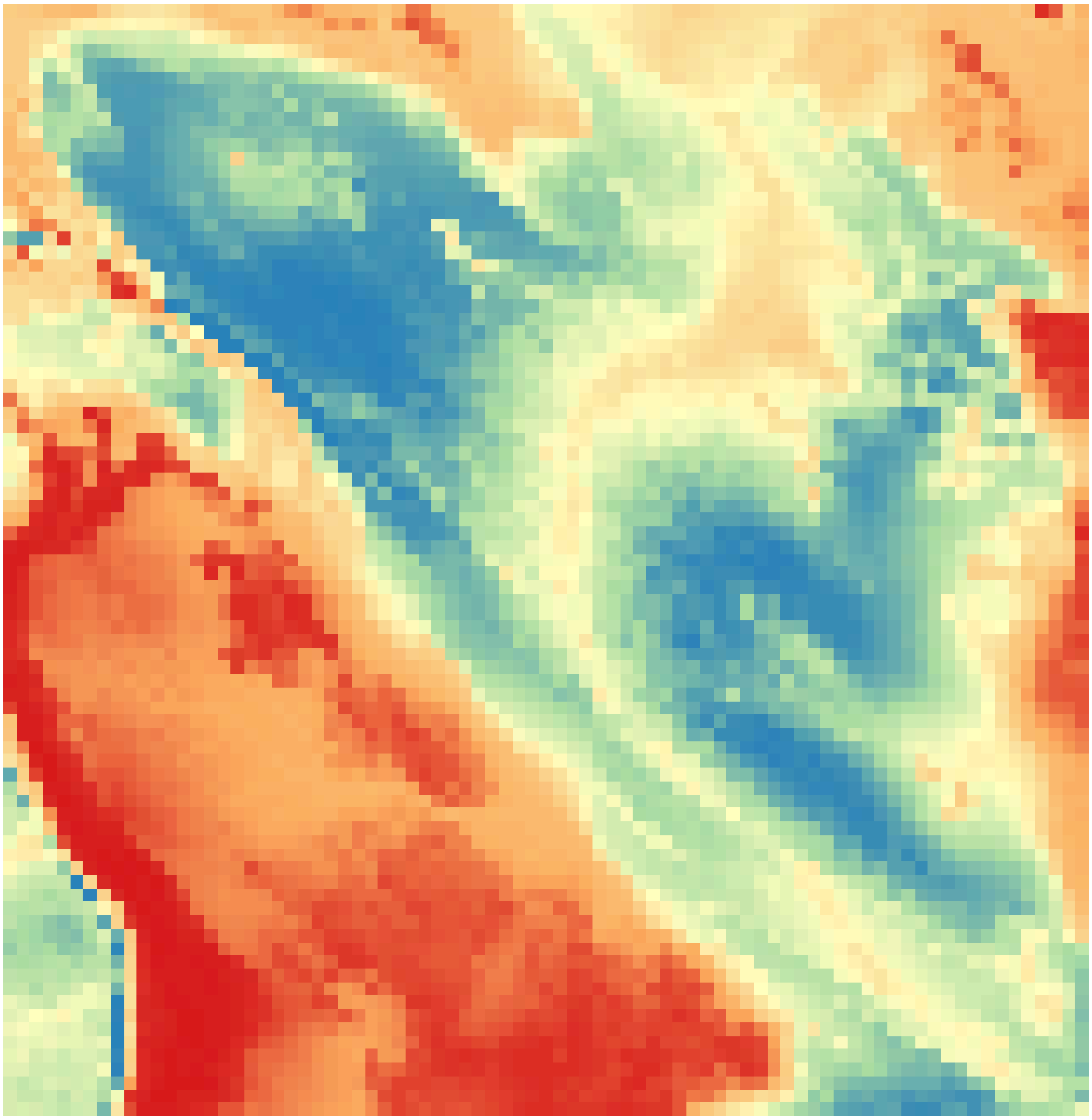


Figure C.59 — Visualization in the ISEA 5x6 rotated and shared space of a GeoTIFF response for retrieving data for ISEA9R DGGRS zone A2-0 of GEBCO 2014 bathymetry at a relative depth of 4

Because ISEA9R zone edges are fully aligned with the axes of the ISEA 5×6 rotated and sheared space and the rhombic zones are square in this *space, a GeoTIFF representation provides a one-to-one mapping of zones to rectilinear gridded coverage cells, as can be seen by comparing this previous visualization of the GeoTIFF response with the visualization of the DGGS-JSON responses. For the same reason, an ISEA9R 2D Tile Matrix Set can be defined in this 5×6 space, which supports implementing the OGC API — Tiles Standard as an alternative or complementary access mechanism to implementing the Zone Data Retrieval requirements class of this DGGS API.*

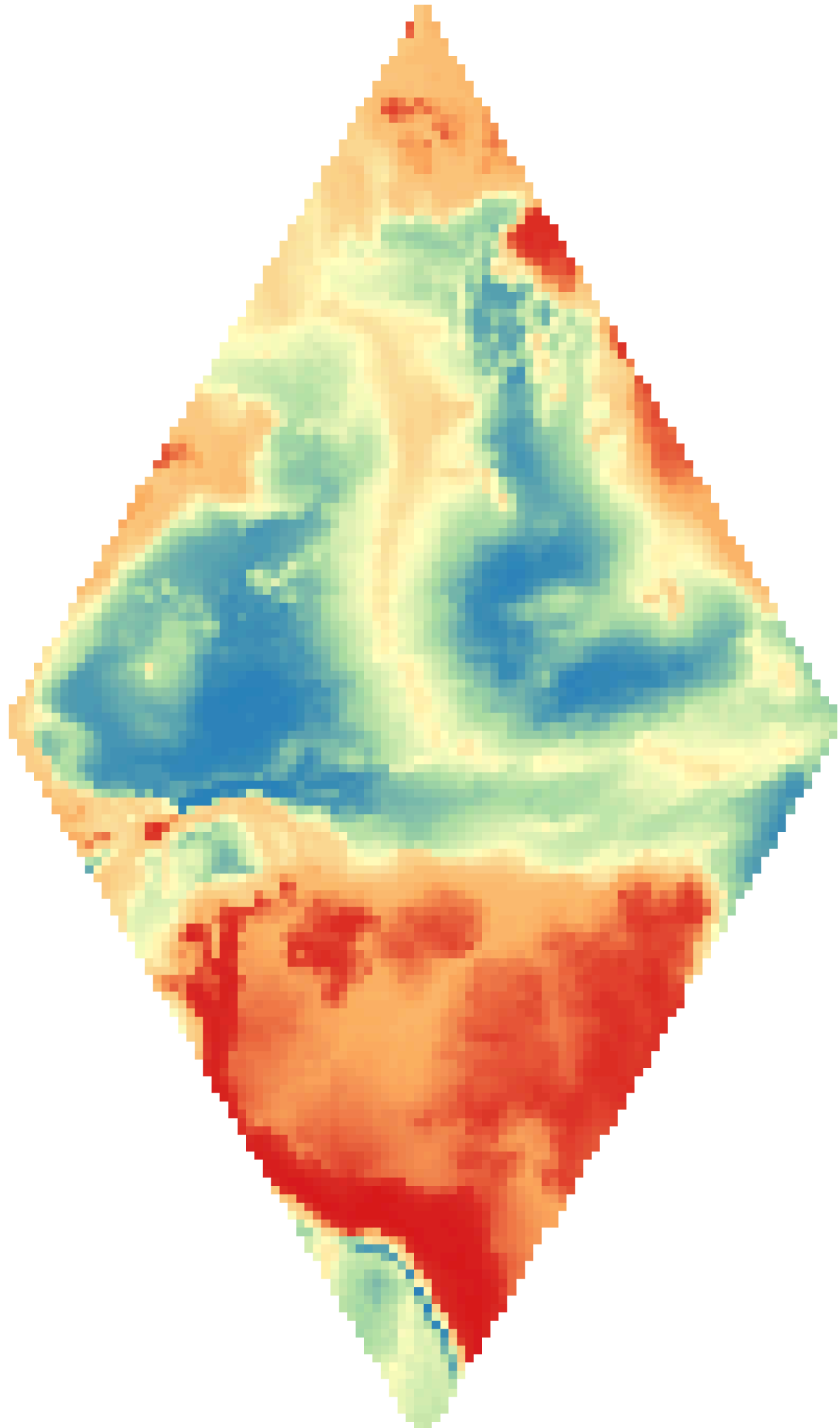


Figure C.60 — Visualization of an ISEA planar projection GeoTIFF response for retrieving data for ISEA9R DGGRS zone A2-0 of GEBCO 2014 bathymetry at a relative depth of 4

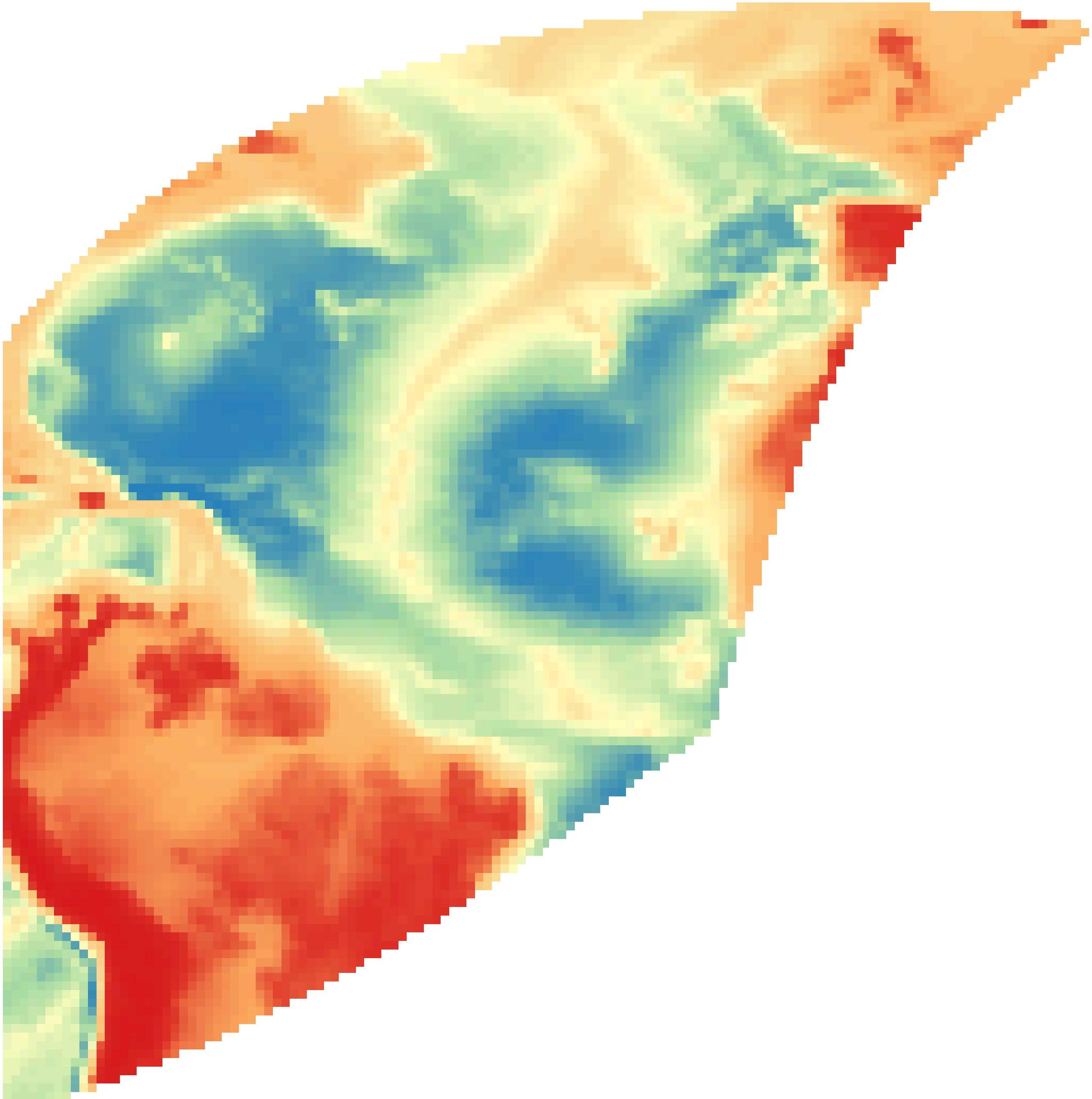


Figure C.61 — Visualization of a CRS84 GeoTIFF response for retrieving data for ISEA9R DGGRS zone A2-0 of GEBCO 2014 bathymetry at a relative depth of 4

> *In these two previous visualizations, the server returned a response in the ISEA planar projection and CRS84, respectively. If the affine transformation from the ISEA 5×6 rotated and sheared space is done on the client side*

instead, the rhombic shape of the individual sub-zones would be preserved, as in the corresponding visualizations of the DGGS-JSON representation.

C.8.2.3. GeoJSON Examples

Using centroid points for the geometry of each zone (`geometry=zone-centroid`), the GeoJSON response and associated visualization would look like the following:


```
{
   "type" : "FeatureCollection",
   "features" : [ {
      "type" : "Feature",
      "id" : 1,
      "geometry" : {
         "type" : "Point",
         "coordinates" : [-78.3336155911744, 31.7735879687561]
      },
      "properties" : {
         "Elevation" : -684.1801215563401,
         "zoneID" : "E2-0"
      }
   }, {
      "type" : "Feature",
      "id" : 2,
   ...
```


Listing

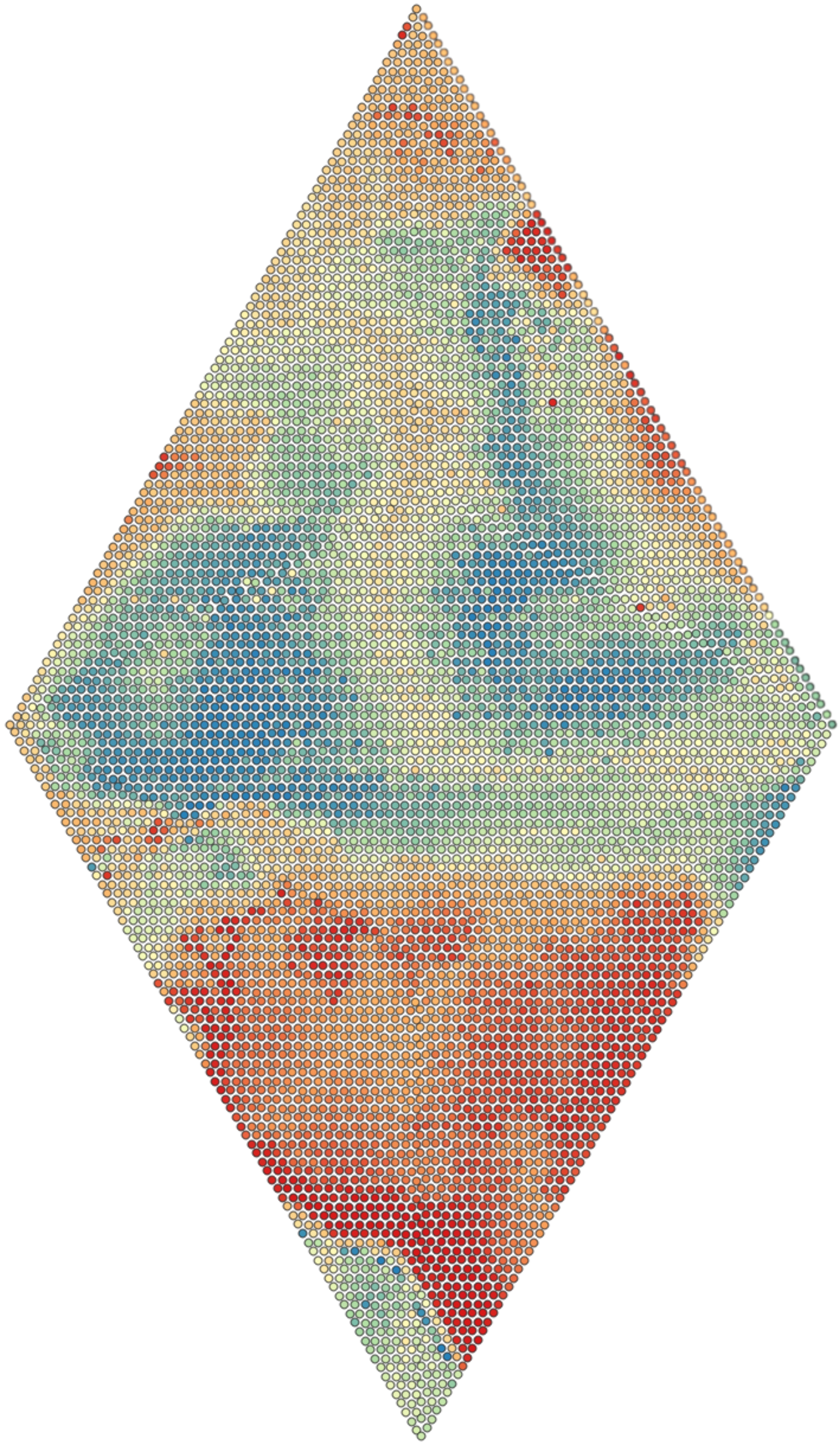


Figure C.62 — Visualization of a GeoJSON point features response for data retrieved from ISEA9R DGGRS zone A2-0 of GEBCO 2014 bathymetry at a relative depth of 4

The following GeoJSON feature collection is requested from an OpenStreetMap dataset of Ottawa for ISEA9R zone I0-1EEBFA7, and illustrated as visualized in different projections.

https://maps.gnosis.earth/ogcapi/collections/osm:ottawa:roads/dggs/ISEA9R/zones/I0-1EEBFA7/data


```
{
   "type": "FeatureCollection",
   "features": [ {
      "type": "Feature",
      "id": 2305843009253710926,
      "geometry": {
         "type": "LineString",
         "coordinates": [
            [ -75.69465583, 45.42431289 ],
            [ -75.69500353, 45.42453159 ],
            [ -75.69543003, 45.4247375 ]
         ]
      },
      "properties": {
         "highway": "path",
         "name": "Rideau Canal Western Pathway",
         "surface": "asphalt",
         "bicycle": "yes",
         "foot": "yes",
         "tunnel": "yes",
         "width": "6"
      }
   }, {
      "type": "Feature",
      "id": 2305843009253711182,
      "geometry": {
         "type": "LineString",
         "coordinates":
   ...
```


Listing

Figure C.63 — Visualization in a world Mercator projection of a GeoJSON feature collection with mixed geometry types for data retrieved from ISEA9R DGGRS zone I0-1EEBFA7 of OpenStreetMap (c) OpenStreetMap contributors

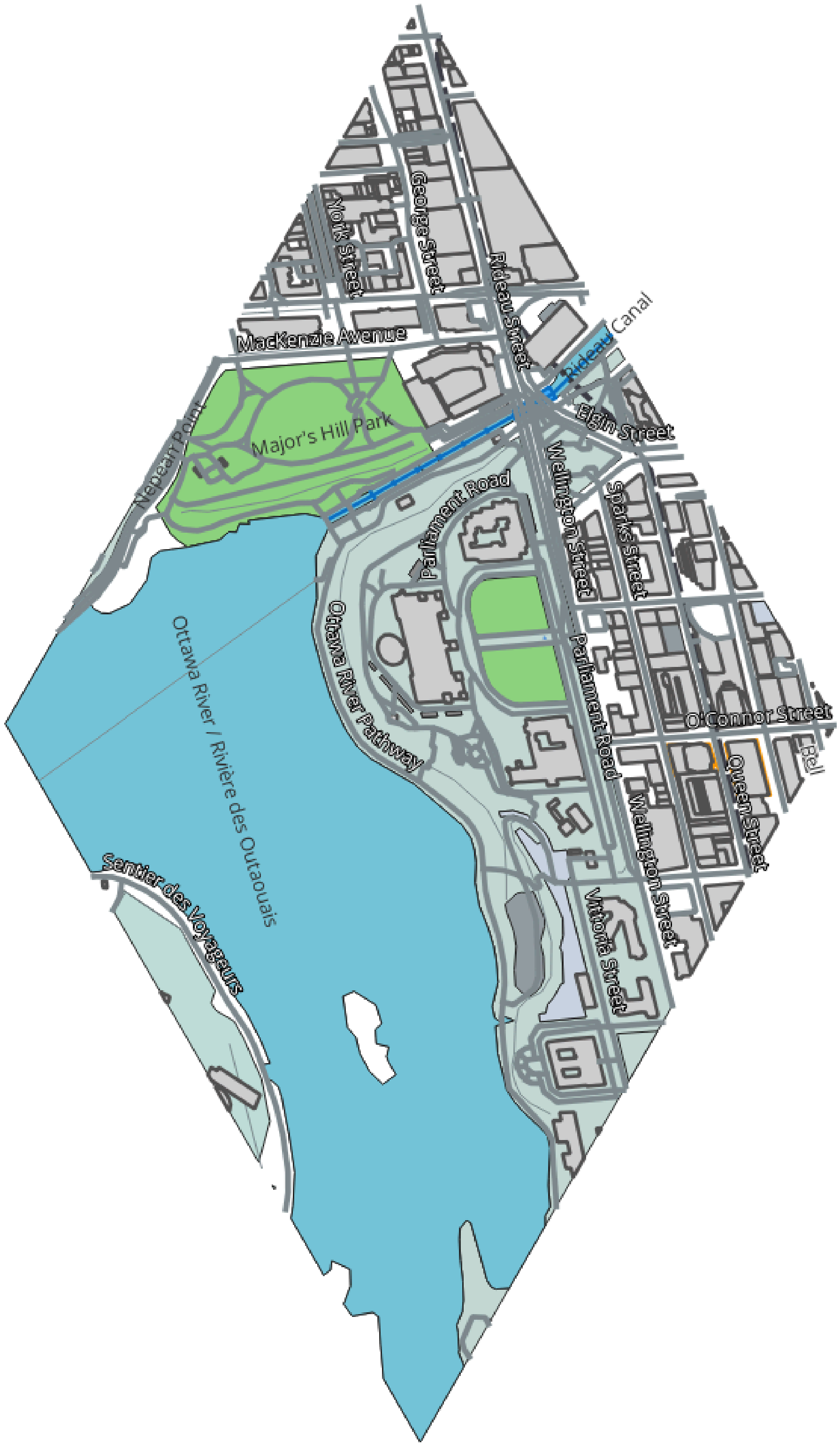


Figure C.64 — Visualization in an ISEA planar projection of a GeoJSON feature collection with mixed geometry types for data retrieved from ISEA9R DGGRS zone I0-1EEBFA7 of OpenStreetMap (c) OpenStreetMap contributors

Figure C.65 — Visualization in ISEA 5x6 rotated and sheared space of a GeoJSON feature collection with mixed geometry types for data retrieved from ISEA9R DGGRS zone I0-1EEBFA7 of OpenStreetMap (c) OpenStreetMap contributors

```
{
   "type": "FeatureCollection",
   "features": [ {
      "type": "Feature",
      "id": 1,
      "properties": {
```

```
            "scalerank": 0,
            "featurecla": "Bathymetry"
         },
         "geometry": {
            "type": "MultiPolygon",
            "coordinates": [
               [
                  [ [ -176.462294193695101, 51.573747730189602 ],
                    [ -176.299702455578142, 51.743306963471383 ],
                    [ -176.29463468229261, 51.738291979490903 ],
                    [ -176.292808119410552, 51.75049674056649 ],
                    [ -176.271367699388492, 51.77285594050052 ],
                    ...
```

Listing

The following GeoJSON polygon contours are requested from Natural Earth bathymetry for ISEA9R zone A9-0.

https://maps.gnosis.earth/ogcapi/collections/NaturalEarth:physical:bathymetry:ne_10m_bathymetry_L_0/dggs/ISEA9R/zones/A9-0/data

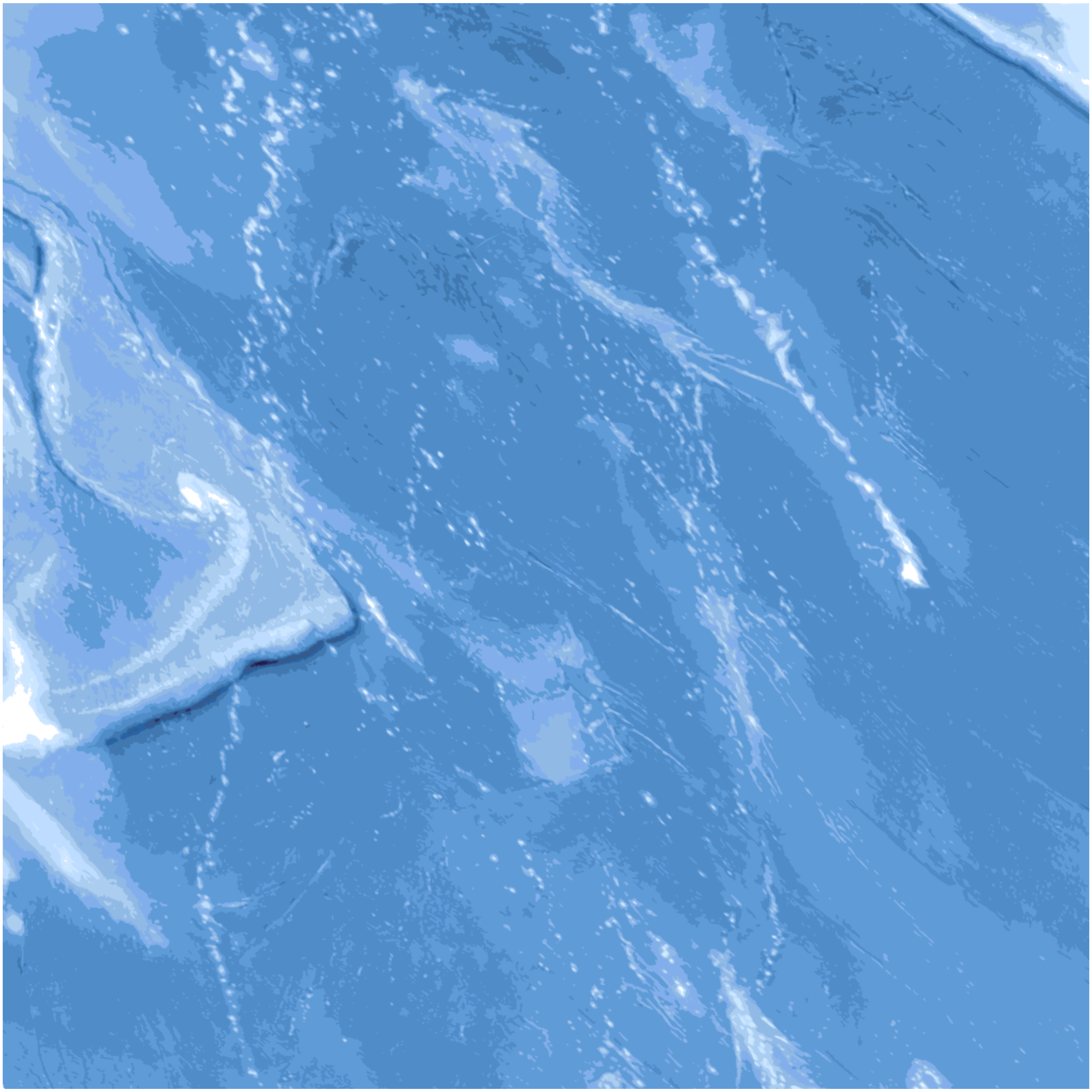


Figure C.66 — Visualization in the ISEA 5x6 rotated and sheared space of GeoJSON polygon contour features for data retrieved from ISEA9R DGGRS zone A9-0 of Natural Earth bathymetry contours.

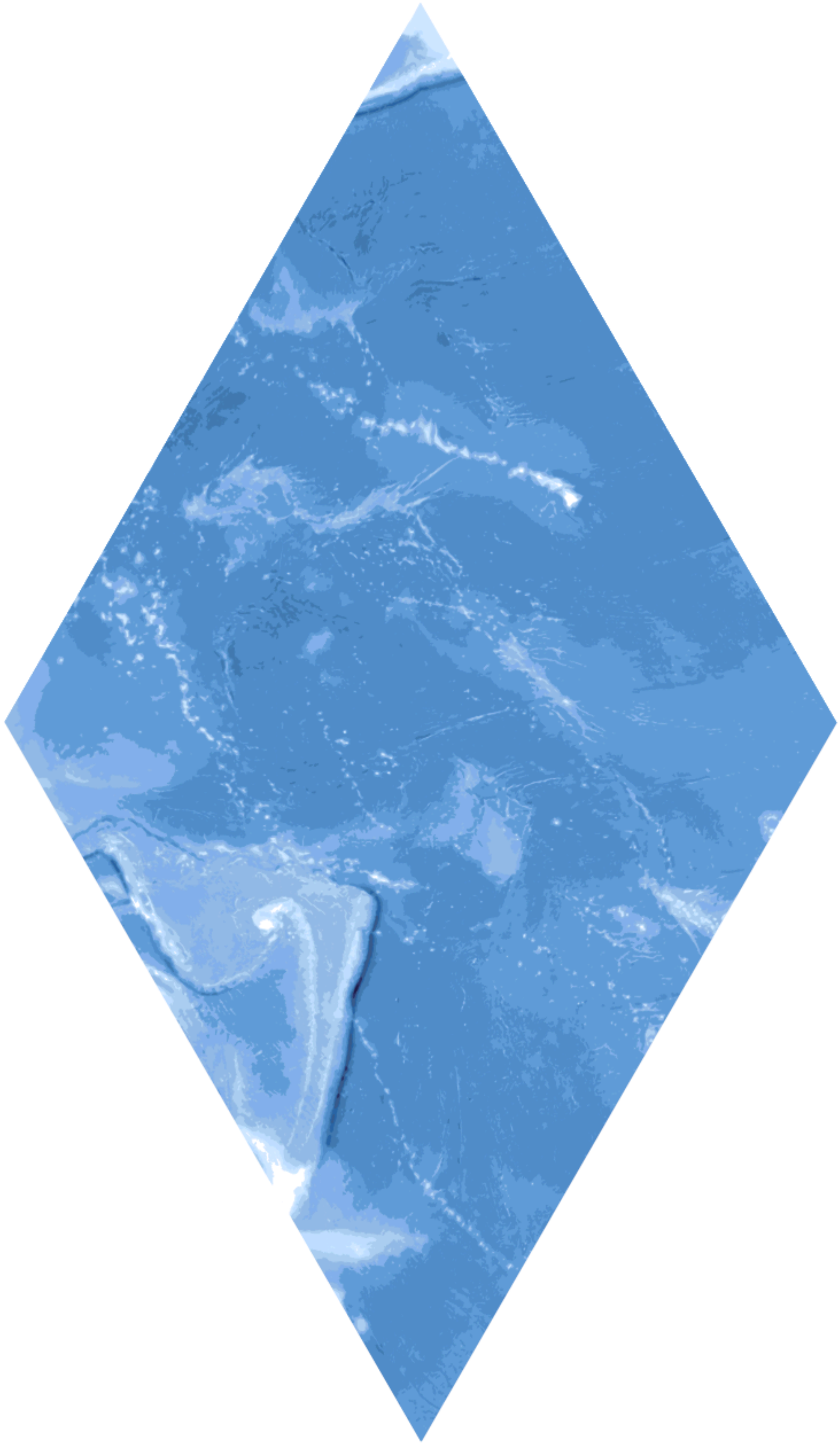


Figure C.67 — Visualization in an ISEA planar projection of GeoJSON polygon contour features for data retrieved from ISEA9R DGGRS zone A9-0 of Natural Earth bathymetry contours.

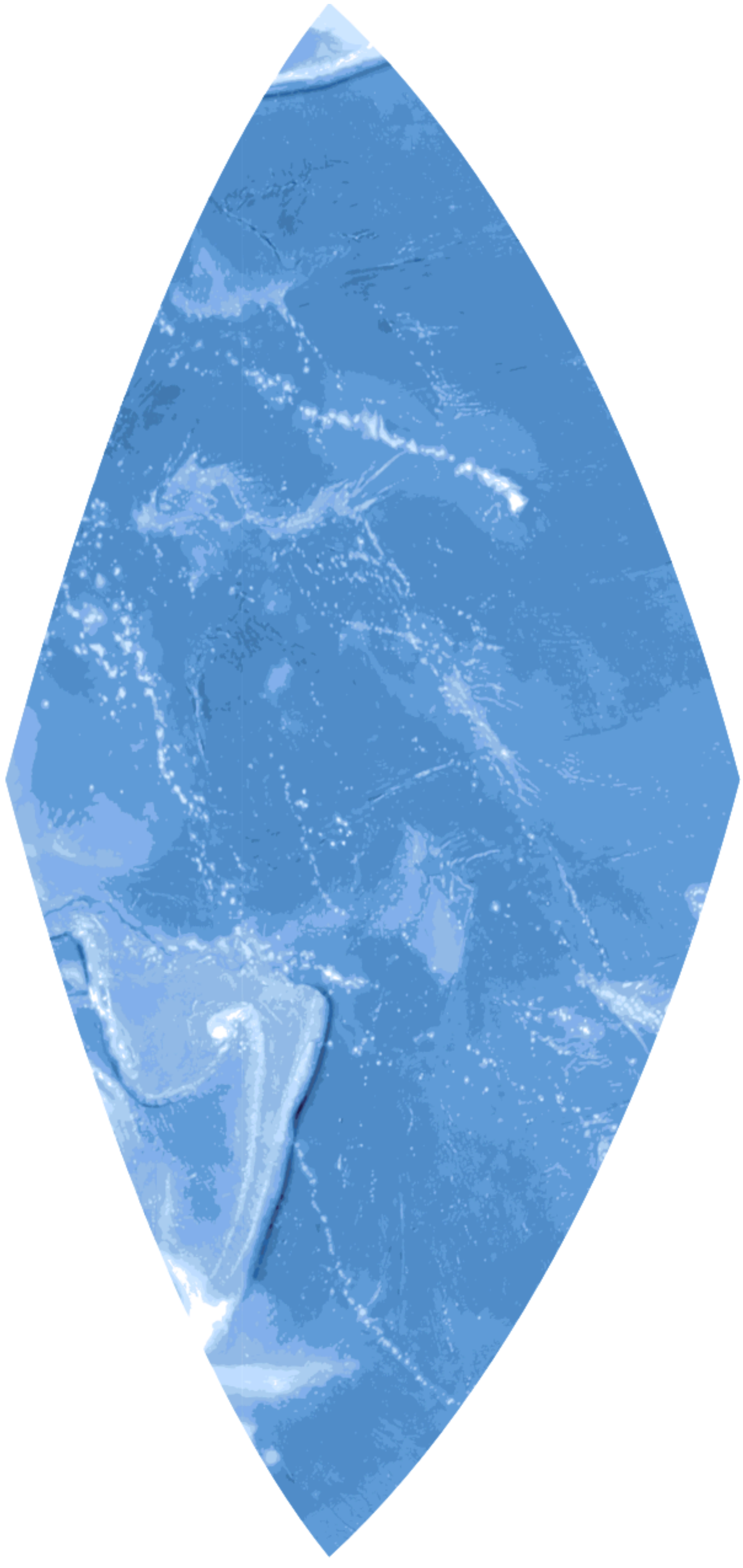


Figure C.68 — Visualization in a Mollweide projection of GeoJSON polygon contour features for data retrieved from ISEA9R DGGRS zone A9-0 of Natural Earth bathymetry contours.

C.8.3. Retrieving data from a GNOSIS Global Grid DGGRS

Examples in this section retrieve data for rectangular zones of a GNOSIS Global Grid DGGRS, with edges aligned to the geographic latitude and longitude axes.

C.8.3.1. DGGS-JSON Examples

The following request is for data from the 0-1-3 level 0 GNOSIS Global Grid zone in DGGS-JSON at a relative depth of 7:

https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/GNOSISGlobalGrid/zones/0-1-3/data?zone-depth=7

```
{
   "dggrs" : "https://www.opengis.net/def/dggrs/OGC/1.0/GNOSISGlobalGrid",
   "zoneId" : "0-1-3",
   "depths" : [ 7 ],
   "values" : {
      "Elevation" : [
         {
            "depth" : 7,
            "shape" : { "count" : 10923, "subZones" : 10923 },
            "data" : [ -4062.7933481641721, -4645.8313990445513, ...,
2878.8939861844092 ]
         }
      ]
   }
}
```

Listing

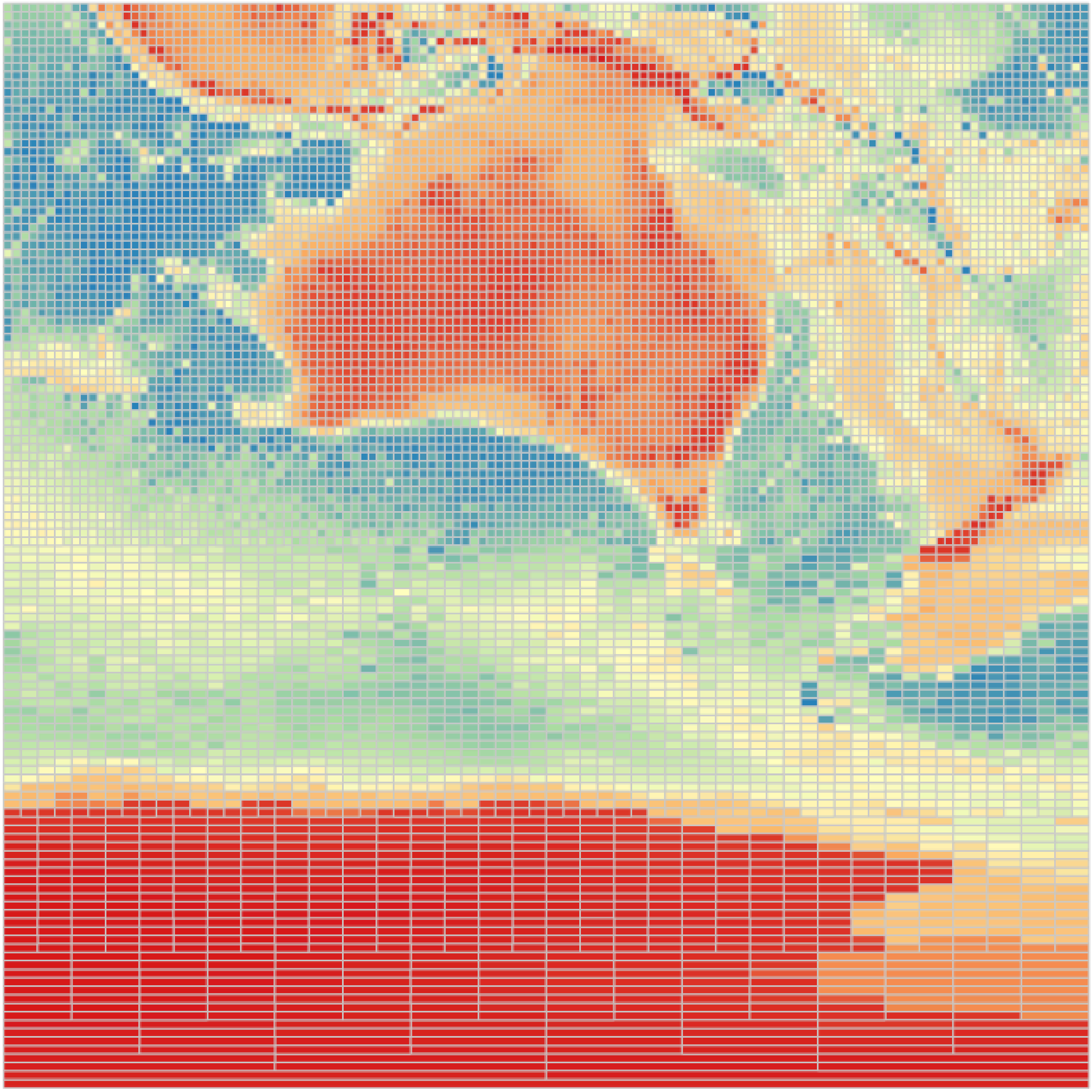


Figure C.69 — Visualization of CRS84 DGGS-JSON response for retrieving data for GNOSIS Global Grid DGGRS zone 0-1-3 of GEBCO 2014 bathymetry at a relative depth of 7

Notice how there are fewer zones per rows approaching the poles, until there is a single zone at the final polar row. While not achieving the 1% error budget allowed for a DGGH to qualify as being equal area, the use of the Variable Width 2D Tile Matrix Set concept allows the GNOSIS Global Grid to maintain a divergence in area from mean zone surface area below 50%

up to a high refinement level, with zones aligned to the geodetic latitude and longitude axes. This in turn greatly facilitates data integration from other discrete global grid hierarchies as well as arbitrary data sources, as no re-projection is needed for WGS84 coordinates.

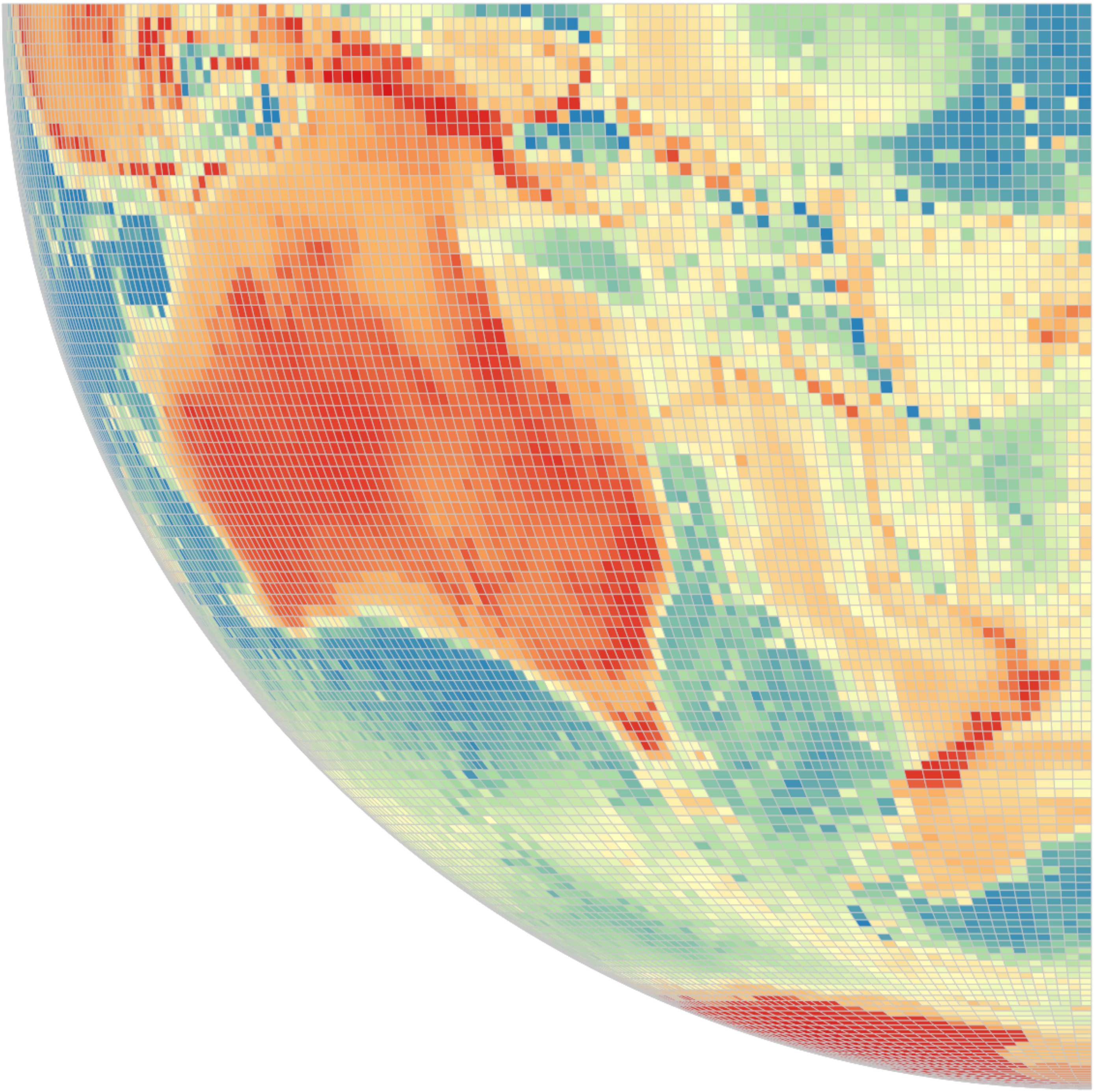


Figure C.70 — Visualization in an orthographic projection of a DGGS-JSON response for retrieving data for GNOSIS Global Grid DGGRS zone 0-1-3 of GEBCO 2014 bathymetry at a relative depth of 7

C.8.3.2. GeoTIFF Examples

The following illustrates the GeoTIFF response for requesting data from the same 0-1-3 GNOSIS Global Grid zone at same relative depth of 7, visualized below in both its native CRS84 and an orthographic projection.

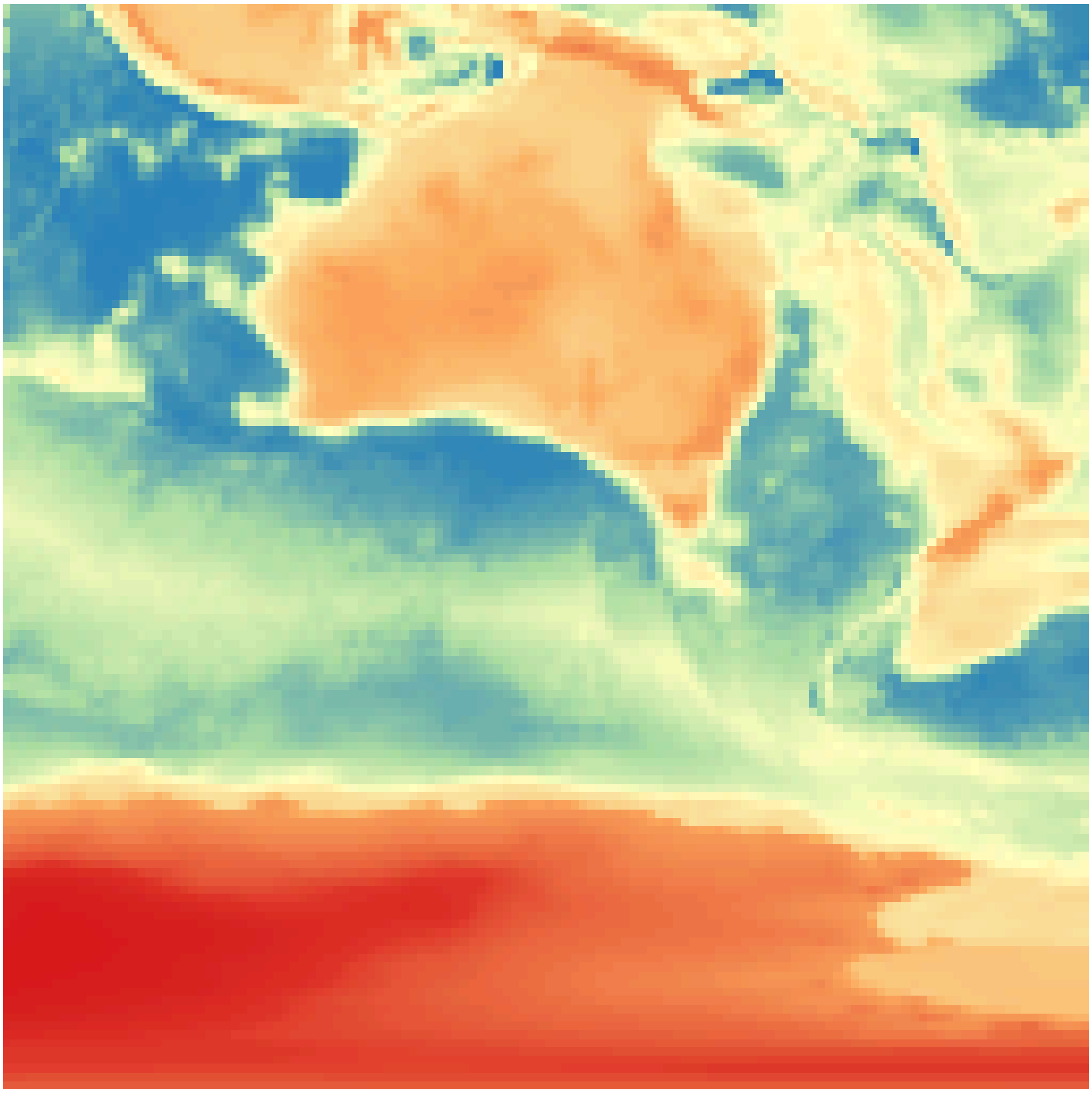


Figure C.71 — Visualization of CRS84 GeoTIFF response for retrieving data for GNOSIS Global Grid DGGRS zone 0-1-3 of GEBCO 2014 bathymetry at a relative depth of 7

> *As can be seen by comparing this previous visualization of the GeoTIFF response with the visualization of the DGGS-JSON responses, because the GNOSIS Global Grid zone edges are fully aligned with geographic latitude and longitude, a GeoTIFF representation provides a one-to-one mapping of*

zones to rectilinear gridded coverage cells, with the exception of the GeoTIFF representation having a fixed width, requiring to duplicate the values of zones approaching the pole.

A registered GNOSIS Global Grid 2D Tile Matrix Set is defined which supports implementing the OGC API — Tiles Standard as an alternative or complementary access mechanism to implementing the Zone Data Retrieval requirements class of this DGGS API.

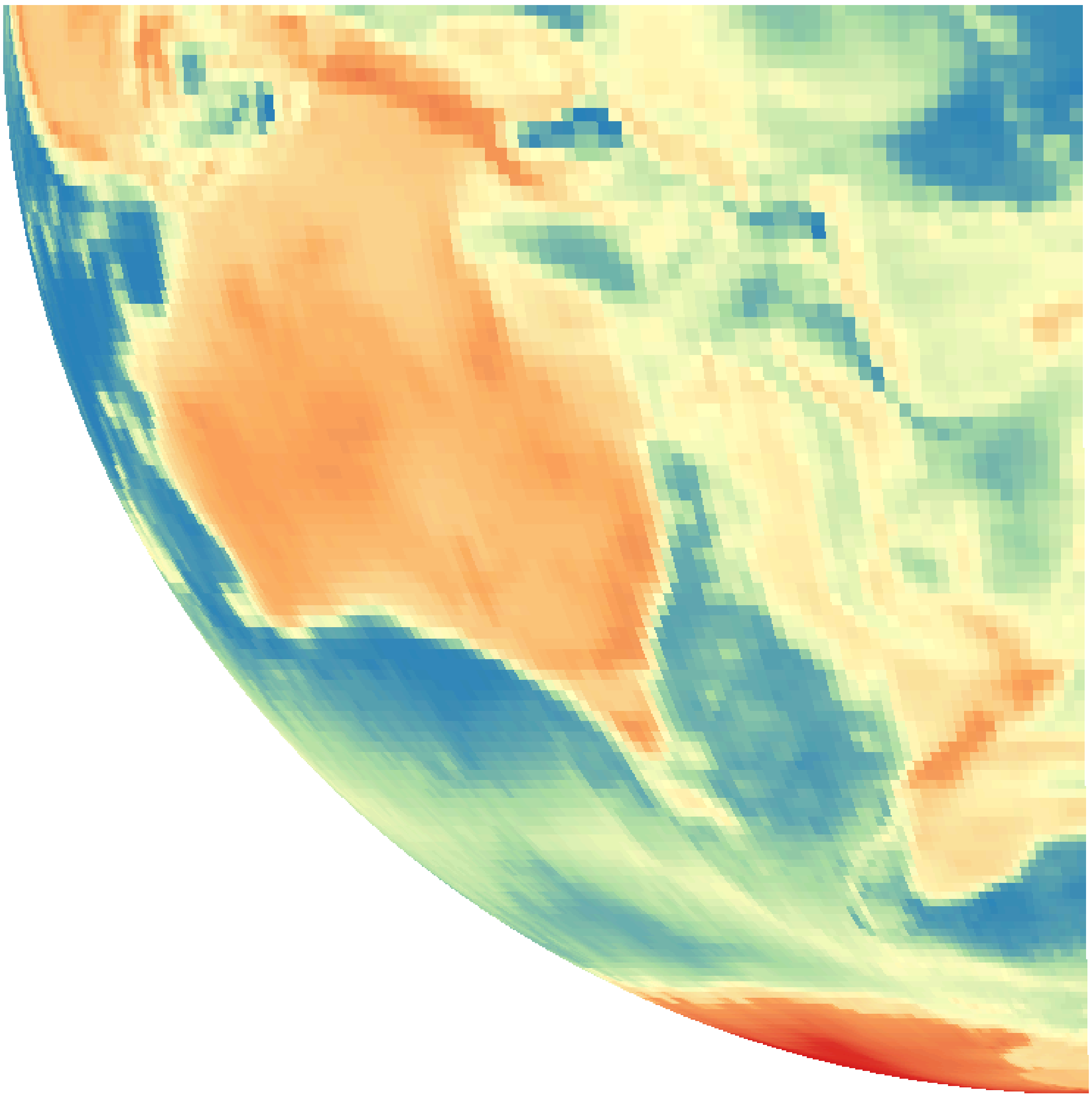


Figure C.72 — Visualization in an orthographic projection of GeoTIFF response for retrieving data for GNOSIS Global Grid DGGRS zone 0-1-3 of GEBCO 2014 bathymetry at a relative depth of 7

C.8.3.3. GeoJSON Examples

Using centroid points for the geometry of each zone (`geometry=zone-centroid`), the GeoJSON response and associated visualization would look like the following:


```
{
   "type" : "FeatureCollection",
   "features" : [ {
      "type" : "Feature",
      "id" : 1,
      "geometry" : {
         "type" : "Point",
         "coordinates" : [90.3515625, -0.3515625]
      },
      "properties" : {
         "Elevation" : -4062.7933481641721,
         "zoneID" : "7-80-180"
      }
   }, {
      "type" : "Feature",
      "id" : 2,
      ...
```


Listing

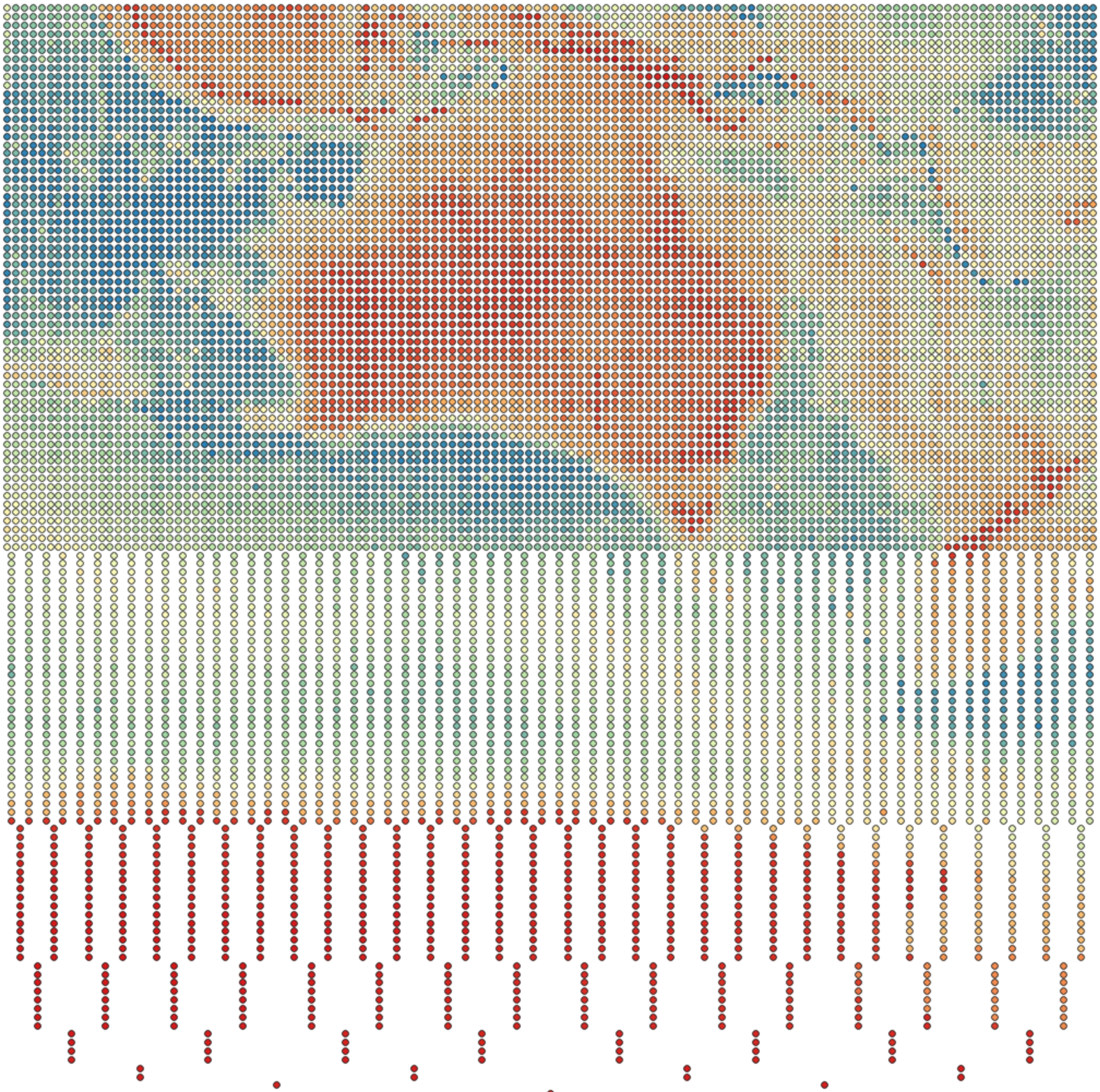


Figure C.73 — Visualization of a GeoJSON point features response for data retrieved from GNOSIS Global Grid DGGRS zone 0-1-3 of GEBCO 2014 bathymetry at a relative depth of 7

The fewer zones approaching the poles based on the Variable Width 2D Tile Matrix Set concept are especially noticeable in these visualizations of the sub-zone centroids as points, illustrating how the GNOSIS Global Grid *achieves a rough approximation of equal area despite zone edges being aligned to geographic latitude and longitude.*

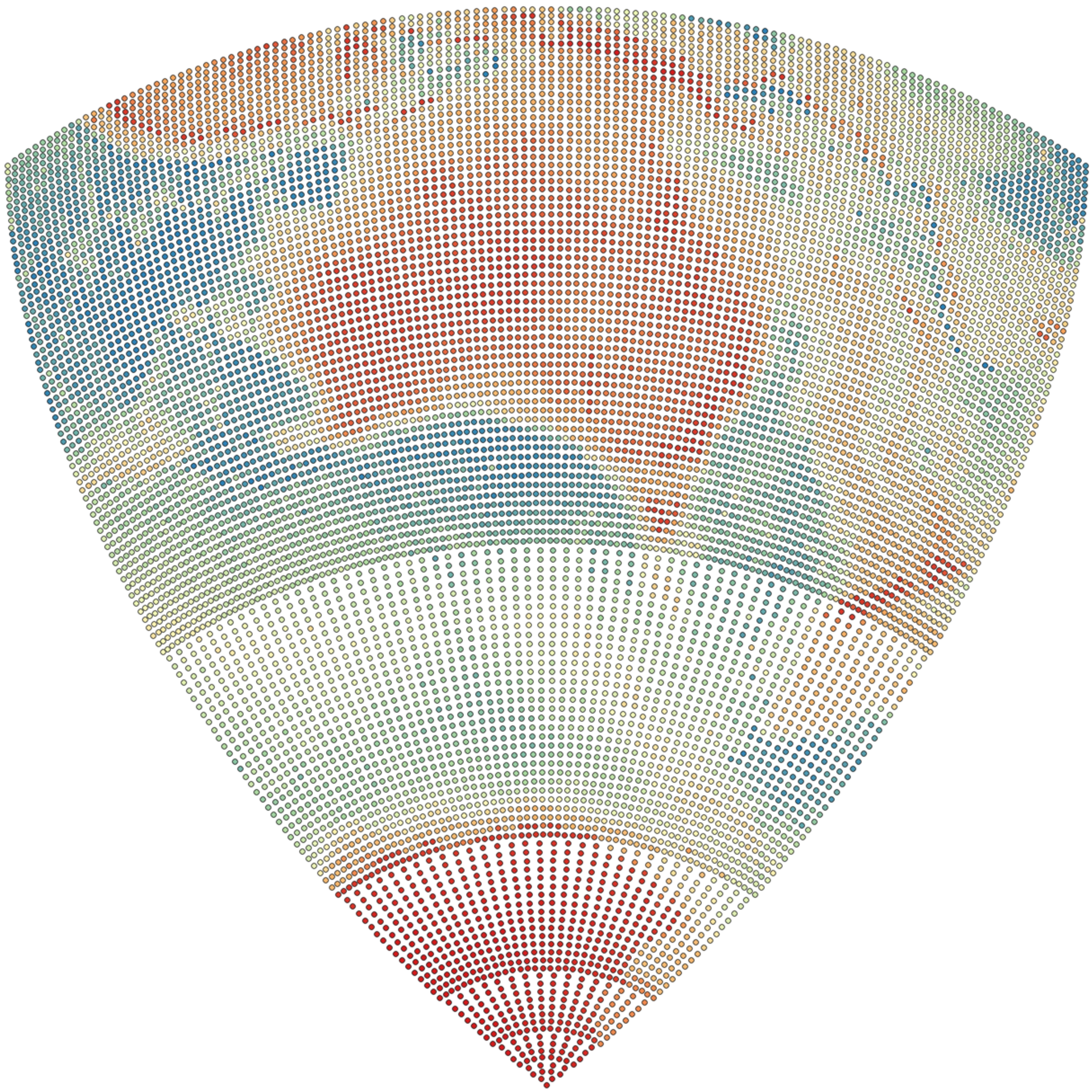


Figure C.74 — Visualization in an orthographic projection of a GeoJSON point features response for data retrieved from GNOSIS Global Grid DGGRS zone 0-1-3 of GEBCO 2014 bathymetry at a relative depth of 7

The following GeoJSON feature collection is requested from an OpenStreetMap dataset of Singapore for GNOSIS Global Grid level 12 zone C-FC5-3276:

https://maps.gnosis.earth/ogcapi/collections/osm:singapore:roads/dggs/GNOSISGlobalGrid/zones/C-FC5-3276/data

The map below visualizes that zone together with the zone immediately East of it, C-FC5-3277:

https://maps.gnosis.earth/ogcapi/collections/osm:singapore:roads/dggs/GNOSISGlobalGrid/zones/C-FC5-3277/data


```
{
   "type" : "FeatureCollection",
   "features" : [ {
      "type" : "Feature",
      "id" : 2305843009223282082,
      "geometry" : {
         "type" : "LineString",
         "coordinates" :
            [ [103.842845400531, 1.2916332189689], [103.8427734375,
1.2916700596819] ]
      },
      "properties" : {
         "highway" : "residential",
         "name" : "Unity Street",
         "surface" : "asphalt",
         "lanes" : "2",
         "turn:lanes:forward" : "through",
         "lanes:backward" : "1",
         "lanes:forward" : "1",
         "turn:lanes:backward" : "left"
      }
   }, {
      "type" : "Feature",
      "id" : 2305843009223284260,
      ...
```


Listing

> *Notice the artificial edge drawn with a grey stroke splitting the Flower Dome in two, around the middle of the map. With the artificial edge information of DGGS-JSON-FG, a client could avoid such visualization artifacts, and could more easily recombine geometry split across separate zones.*

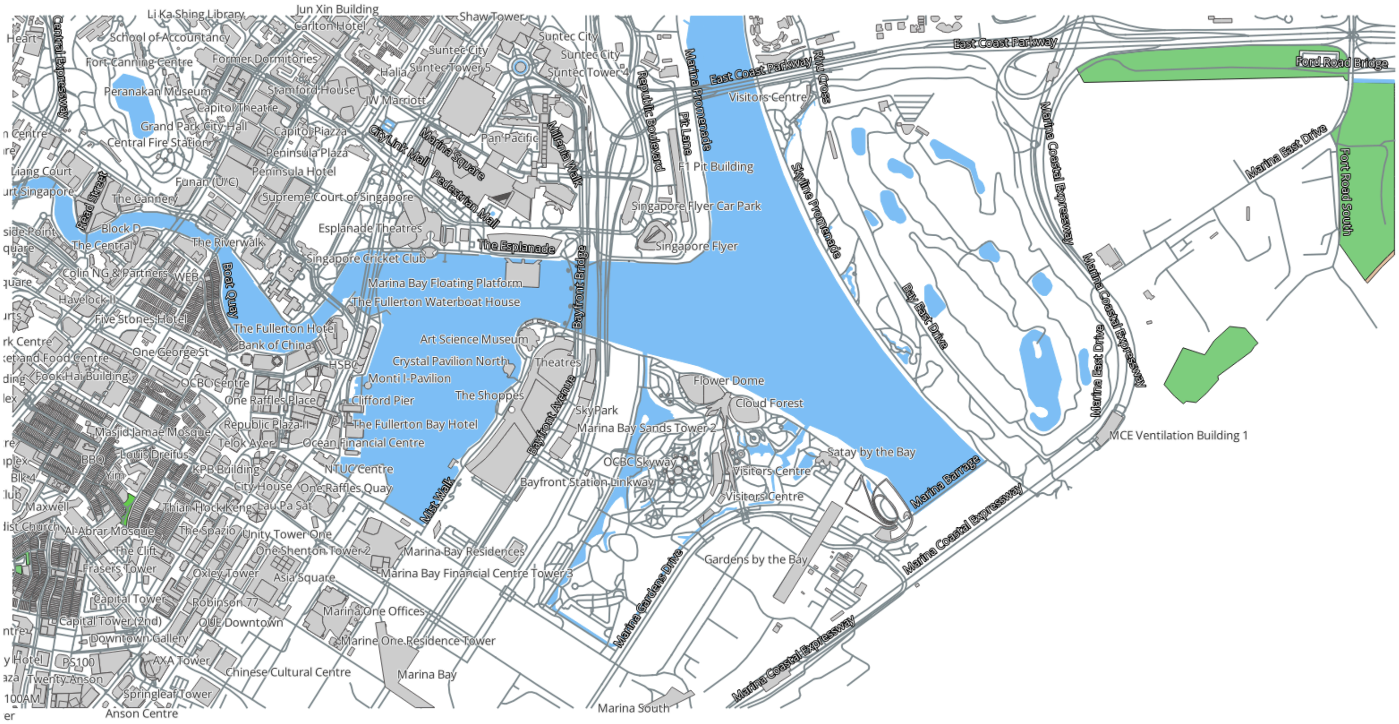


Figure C.75 — Visualization of a GeoJSON feature collection with mixed geometry types for data retrieved from GNOSIS Global Grid DGGRS zones C-FC5-3276 and C-FC5-3277 of OpenStreetMap (c) OpenStreetMap contributors

The following GeoJSON polygon contours are requested from Natural Earth bathymetry for GNOSIS Global Grid level 0 zone 1-1-7.

https://maps.gnosis.earth/ogcapi/collections/NaturalEarth:physical:bathymetry:ne_10m_bathymetry_L_0/dggs/GNOSISGlobalGrid/zones/1-1-7/data


```
{
   "type" : "FeatureCollection",
   "features" : [ {
      "type" : "Feature",
      "id" : 1,
      "geometry" : {
         "type" : "Polygon",
         "coordinates" : [
            [ [136.0693439846182, 33.8470804339864],
              [136.0242801226881, 33.75],
              [135.9791304248306, 33.6527478941586],
              [135.9583581303791, 33.6329197949094],
              [135.9327790239883, 33.6112033052555],
              ...
```


Listing

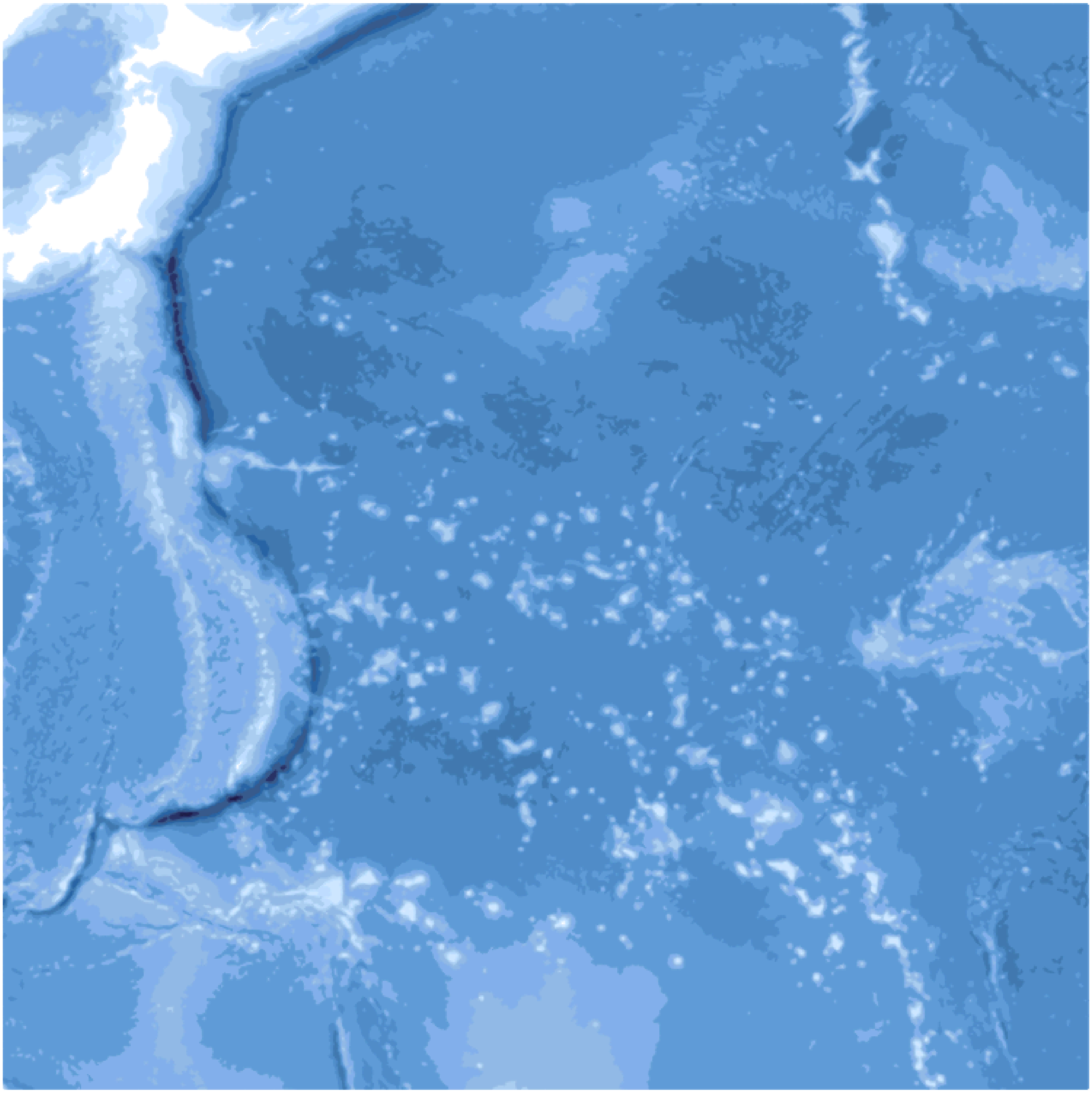


Figure C.76 — Visualization of GeoJSON polygon contour features for data retrieved from GNOSIS Global Grid DGGRS zone 1-1-7 of Natural Earth bathymetry contours.

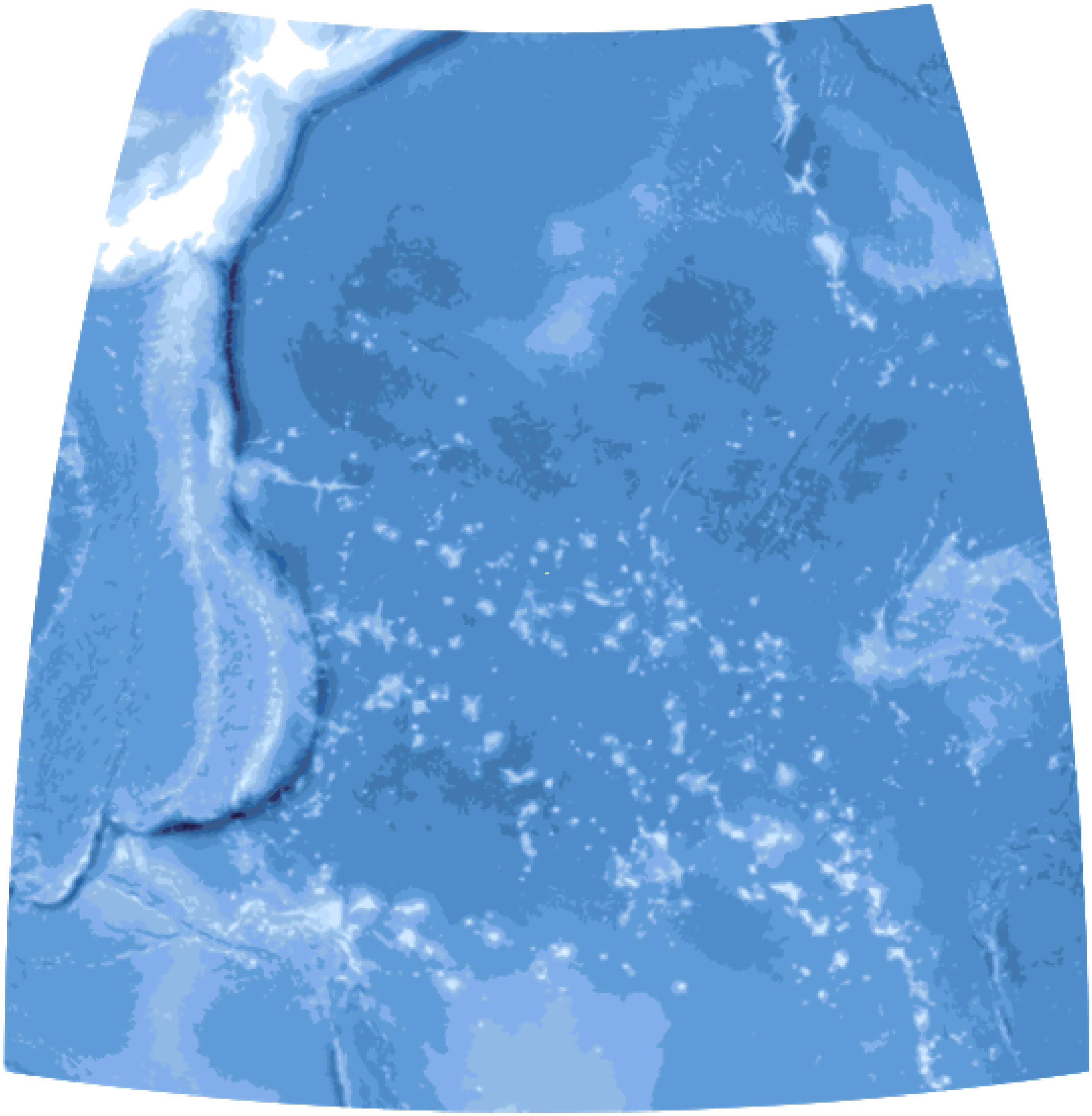


Figure C.77 — Visualization in an orthographic projection of GeoJSON polygon contour features for data retrieved from GNOSIS Global Grid DGGRS zone 1-1-7 of Natural Earth bathymetry contours.

C.9. Requesting data at a particular relative depth

Implementations supporting the `zone-depth` query parameter defined in the "Data Custom Depths" requirements class allows clients to request data at a particular depth.

For example, the two following requests and visualized responses illustrate retrieving zones at relative depths of 4 (91 sub-zone values) and 9 (19,927 sub-zone values), respectively, for ISEA3H zone *H4-A825C-A*:

https://maps.gnosis.earth/ogcapi/collections/sentinel2-l2a/dggs/ISEA3H/zones/H4-A825C-A/data?zone-depth=4&datetime=2020-07-31&properties=B08

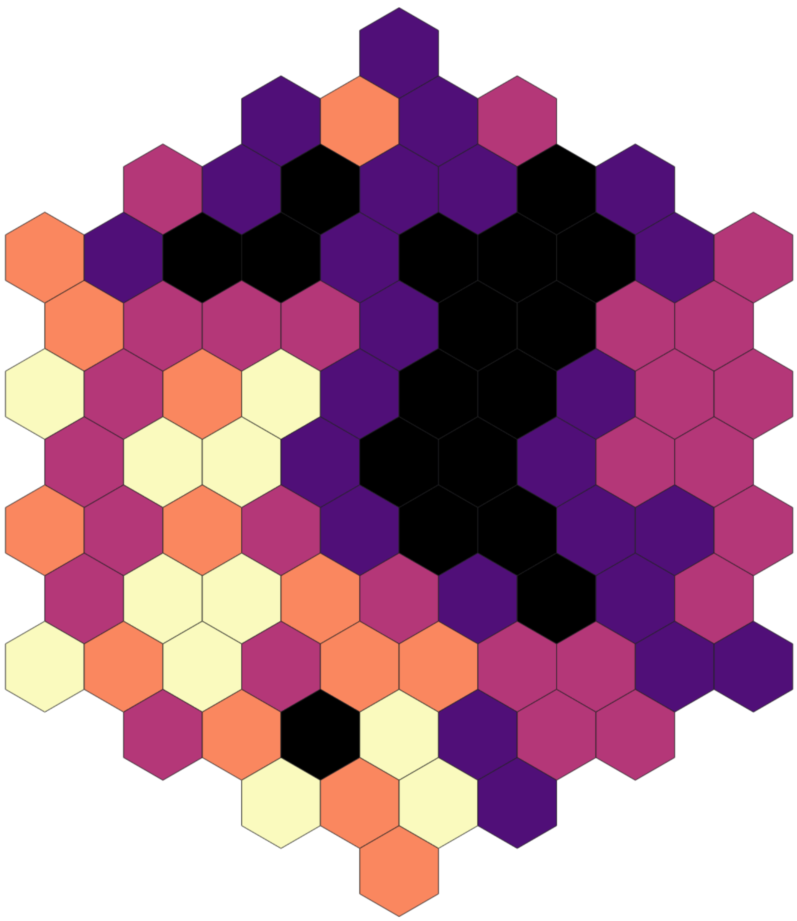


Figure C.78 — Visualization of near-infrared band (`B08`) of ISEA3H zone H4-A825C-A at relative depth of 4 from Copernicus SENTINEL-2 operated by ESA

https://maps.gnosis.earth/ogcapi/collections/sentinel2-l2a/dggs/ISEA3H/zones/H4-A825C-A/data?zone-depth=9&datetime=2020-07-31&properties=B08

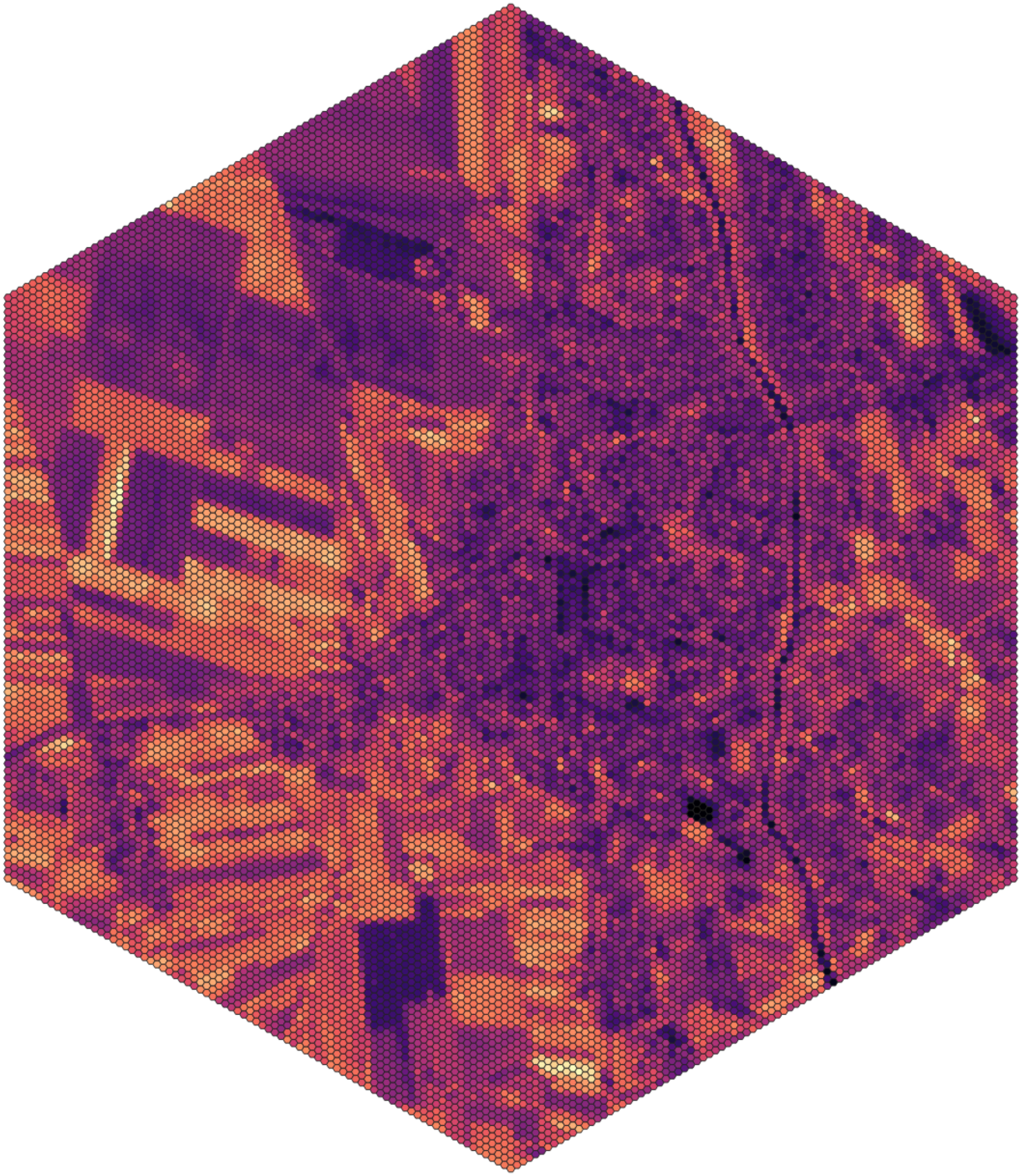


Figure C.79 — Visualization of near-infrared band (`B08`) of ISEA3H zone H4-A825C-A at relative depth of 9 from Copernicus SENTINEL-2 operated by ESA

C.10. Subsetting and filtering zone data

In addition to the basic retrieval of zone data, implementations can support additional query parameters for the …`/zones/{zoneId}/data` resources to return only a portion of the data associated with the zone. The `subset`, `datetime` and `properties` query parameters are defined in the Data Subsetting requirements class, while the `filter` query parameter is defined in the Filtering Zone Data with CQL2 requirements class.

C.10.1. Temporal subsetting

It is possible to retrieve only a portion of a temporal datasets using one of two query parameters: `datetime`, or `subset` with the `time` dimension. The temporal subset can be either a slice specified using a single time instant, or a trim using a lower and upper bound.

The following examples retrieve data from a sentinel-2 collection for the 28th of April, May and June 2022, at relative depth of 12, for ISEA3H zone E7-17D1-A in New Zealand. Since the sentinel-2 dataset does not have imagery for the whole globe at every instant, the server will return the latest imagery available before the specified time instant.

https://maps.gnosis.earth/ogcapi/collections/sentinel2-l2a/dggs/ISEA3H/zones/E7-17D1-A/data?zone-depth=12&datetime=2022-04-28

https://maps.gnosis.earth/ogcapi/collections/sentinel2-l2a/dggs/ISEA3H/zones/E7-17D1-A/data?zone-depth=12&datetime=2022-05-28

https://maps.gnosis.earth/ogcapi/collections/sentinel2-l2a/dggs/ISEA3H/zones/E7-17D1-A/data?zone-depth=12&datetime=2022-06-28

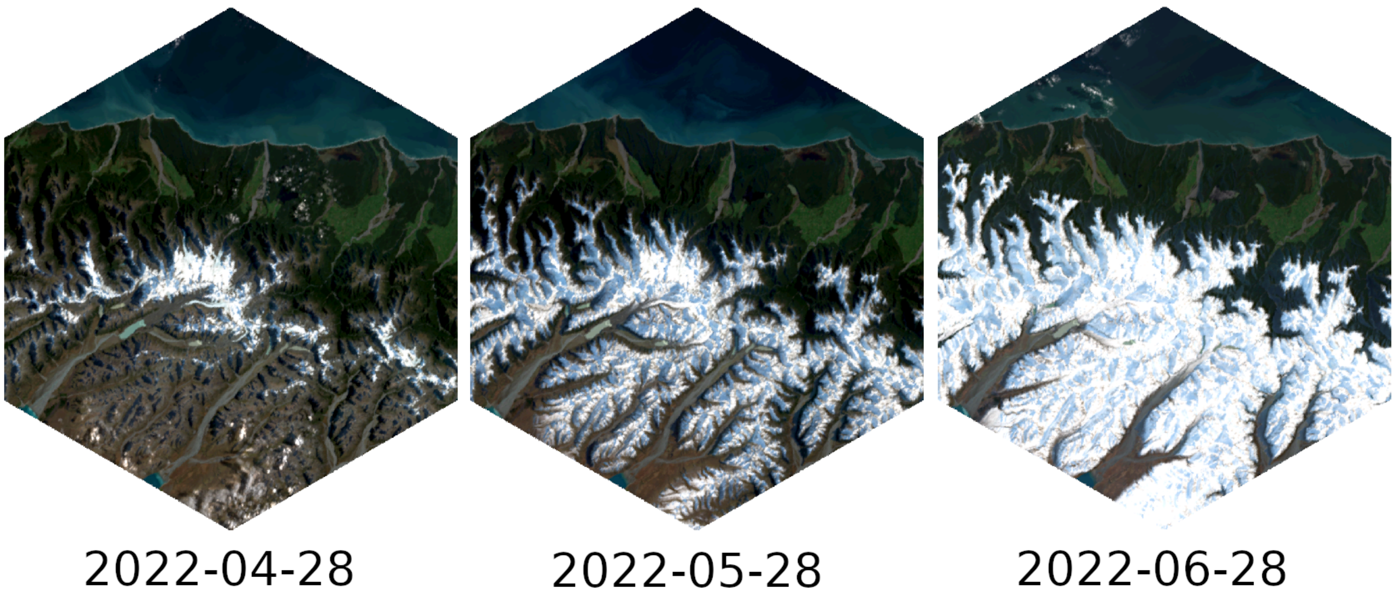


Figure C.80 — Visualization of different temporal slices for ISEA3H zone E7-17D1-A (New Zealand) from Copernicus SENTINEL-2 operated by ESA

> *Although the examples illustrated here are all for rasterized data, temporal subsetting (whether using the `datetime` or `subset=time(…)` syntax) can also be applied to feature collections with vector geometry with a primary temporal property.*

The following examples retrieve data from a sentinel-2 collection for the 15th of February and May 2021, and for the 28th of October 2022, at relative depth of 12, for ISEA3H zone G7-67252-D in K'gari, Australia.

https://maps.gnosis.earth/ogcapi/collections/sentinel2-l2a/dggs/ISEA3H/zones/G7-67252-D/data?zone-depth=12&datetime=2021-02-15

https://maps.gnosis.earth/ogcapi/collections/sentinel2-l2a/dggs/ISEA3H/zones/G7-67252-D/data?zone-depth=12&datetime=2021-05-15

https://maps.gnosis.earth/ogcapi/collections/sentinel2-l2a/dggs/ISEA3H/zones/G7-67252-D/data?zone-depth=12&datetime=2022-10-28

Figure C.81 — Visualization of different temporal slices for ISEA3H zone G7-67252-D (K'gari, Australia) from Copernicus SENTINEL-2 operated by ESA

The following examples retrieve data from a daily CMIP5 climate dataset for the time interval between October 6th to October 10th, 2022 for ISEA3H zone A1-0-E, at a relative depth of 8.

https://maps.gnosis.earth/ogcapi/collections/climate:cmip5:singlePressure/dggs/ISEA3H/zones/A1-0-E/data?zone-depth=8&datetime=2022-10-06/2022-10-10

The following JSON data illustrates an extract of a DGGS-JSON response for this request for the precipitation and wind speed variables. In addition to the sub-zones, this DGGS-JSON sample has temporal dimension for the 7 days of the response.


```
{
   "dggrs" : "[ogc-dggrs:ISEA3H]",
   "zoneId" : "A1-0-E",
   "depths" : [ 8 ],
   "schema":
   {
      "$schema" : "https://json-schema.org/draft/2020-12/schema",
      "$id" : "https://maps.gnosis.earth/ogcapi/collections/climate:cmip5:
singlePressure/schema",
      "title" : "Climate variables",
      "type": "object",
      "properties":
      {
         ...
```

```
         "pr": {
            "type": "number",
            "title": "Mean precipitation flux",
            "description": "Amount of water per unit area and time",
            "x-ogc-propertySeq" : 2,
            "x-ogc-unit": "kg/m2/s",
            "x-ogc-definition": "https://qudt.org/vocab/quantitykind/Flux"
         },
         ...
         "sfcWind": {
            "type": "number",
            "title": "Wind speed",
            "description": "Magnitude of the two-dimensional horizontal air
velocity near the surface",
            "x-ogc-propertySeq" : 6,
            "x-ogc-unit": "m/s",
            "x-ogc-definition": "https://qudt.org/vocab/quantitykind/Speed"
         },
         ...
      }
   },
   "dimensions" : [ {
      "name": "time",
      "interval" : [ "2022-10-06", "2022-10-10" ],
      "grid" : { "cellsCount" : 5, "resolution" : "P1D" }
   } ],
   "values" : {
      "pr" : [ {
         "depth" : 8,
         "shape" : { "count" : 33215, "subZones" : 6643, "dimensions" : {
"time" : 5 } },
         "data" : [
            0.0003729638805, 0.0004226717055, 0.0007104640621, 0.0008059472461,
 0.0000912571239,
            0.0005371499461, 0.0005135724622, 0.0008003884381, 0.0007072870373,
 0.0000603493833,
            ...,
            0.0000608985125, 0.0000827684982, 0.0000371927567, 0.0000246966925,
 0.0000095615553
         ]
      } ],
      "sfcWind" : [ {
         "depth" : 8,
         "shape" : { "count" : 33215, "subZones" : 6643, "dimensions" : {
"time" : 5 } },
         "data" : [
            5.1973060025736, 7.541365875696, 10.7034694149118,
13.1711771530757, 9.8653782764459,
            6.7253713877706, 7.922387440188, 10.7766137018728,
12.3413589506021, 9.2283374128277,
            ...,
            2.4033131921971, 2.5349241545049, 2.4223932801007, 2.3289998661706,
 2.7221016029128
         ]
      } ],
      ...
```


Listing

The following example requests a single day, October 9, 2022, from the same dataset.

https://maps.gnosis.earth/ogcapi/collections/climate:cmip5:singlePressure/dggs/ISEA3H/zones/A1-0-E/data?zone-depth=8&datetime=2022-10-09

The DGGS-JSON extract from the response for this single slice of time does not include an additional time dimension.

```
{
   "dggrs" : "[ogc-dggrs:ISEA3H]",
   "zoneId" : "A1-0-E",
   "depths" : [ 8 ],
   "values" : {
      "pr" : [ {
         "depth" : 8,
         "shape" : { "count" : 6643, "subZones" : 6643 },
         "data" : [ 0.0008059472461, 0.0007072870373, ..., 0.0000246966925 ]
      } ],
      "sfcWind" : [ {
         "depth" : 8,
         "shape" : { "count" : 6643, "subZones" : 6643 },
         "data" : [ 13.1711771530757, 12.3413589506021, ..., 2.3289998661706 ]
      } ],
      ...
```

Listing

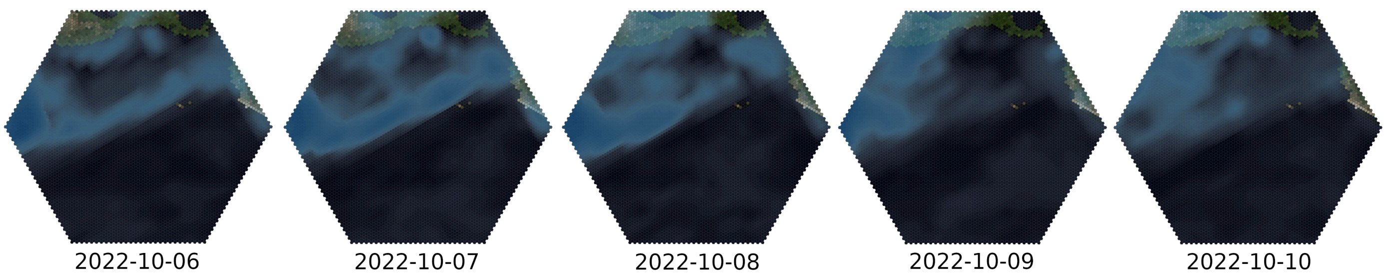


Figure C.82 — Visualization of precipitations from 2022-10-06 to 2022-10-10 for ISEA3H zone A1-0-E from Copernicus Climate Data Store CMIP5 daily data on single levels

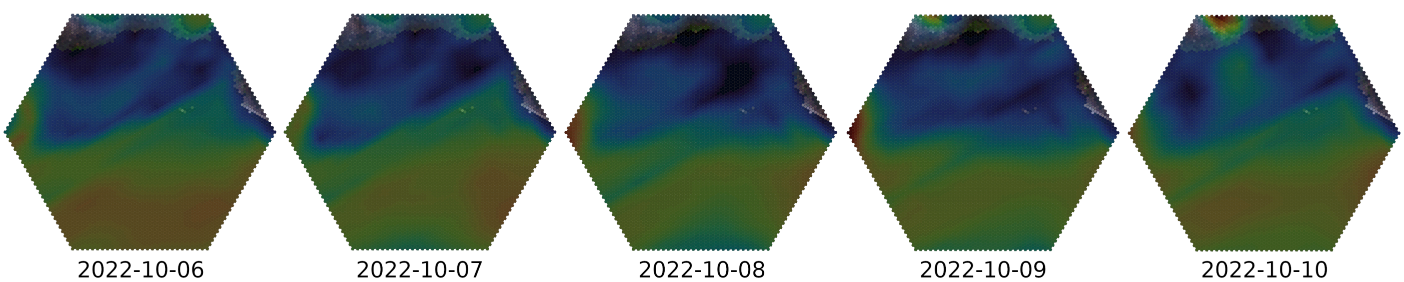


Figure C.83 — Visualization of wind speed from 2022-10-06 to 2022-10-10 for ISEA3H zone A1-0-E from Copernicus Climate Data Store CMIP5 daily data on single levels

> *Temporal subsetting for a slice can also be achieved using the equivalent `subset=time("2022-10-06":"2022-10-10")` syntax for trimming, and `subset=time("2022-10-06")` for slicing.*

C.10.2. Subsetting arbitrary dimensions

In addition to spatial and temporal dimensions, datasets may be comprised of additional dimensions, such as atmospheric pressure. These dimensions can also be subset in a similar manner using the `subset` query parameter.

The following request is for relative humidity from an ERA5 dataset at 11:00 PM on April 6th, 2023 at pressure levels between 750 hPa and 900 hPa (inclusively).

https://maps.gnosis.earth/ogcapi/collections/climate:era5:relativeHumidity/dggs/ISEA3H/zones/A4-0-A/data?zone-depth=8&datetime=2023-04-06T23:00:00&subset=pressure(750:900)

The partial DGGS-JSON response below contains an additional `pressure` dimension, with 7 of the 37 pressure levels available from the collection.


```
{
   "dggrs" : "[ogc-dggrs:ISEA3H]",
   "zoneId" : "A4-0-A",
   "depths" : [ 8 ],
   "schema":
   {
      "$schema" : "https://json-schema.org/draft/2020-12/schema",
      "$id" : "https://maps.gnosis.earth/ogcapi/collections/climate:era5:
relativeHumidity/schema",
      "title" : "ERA5 Relative humidity",
      "type": "object",
      "properties":
      {
         "r": {
            "type": "number",
            "title": "Relative humidity",
            "description": "Water vapour pressure as a percentage of the value
at which the air becomes saturated",
            "x-ogc-propertySeq" : 1,
            "x-ogc-unit": "%",
            "x-ogc-definition": "https://qudt.org/vocab/quantitykind/
RelativeHumidity"
         }
      }
   },
   "dimensions" : [ {
      "name": "pressure",
      "title": "Atmospheric pressure",
      "x-ogc-unit": "hPa",
      "x-ogc-definition": "https://qudt.org/vocab/quantitykind/
AtmosphericPressure"
      "interval" : [ 750, 900 ],
      "grid" : { "cellsCount" : 7, "coordinates" : [ 750, 775, 800, 825, 850,
875, 900 ] }
   } ],
   "values" : {
      "r" : [ {
         "depth" : 8,
         "shape" : { "count" : 38752, "subZones" : 5536, "dimensions" : {
"pressure" : 7 } },
         "data" : [
```

```
            83.0640563964844, 84.7935791015625, 89.8955307006836,
94.8348999023438, 96.1357727050781, 92.2027587890625, 88.4226226806641,
            50.4869918823242, 62.970329284668, 71.4487152099609,
77.9791564941406, 79.6762161254883, 82.0418395996094, 84.0924987792969,
            ...,
            10.35546875, 6.8243680000305, 11.2714557647705, 28.403600692749,
73.5849151611328, 87.5920486450195, 86.5950088500977
         ]
      } ]
   }
}
```


Listing

The following request is for the 900 hPa slice only. The corresponding DGGS-JSON response below does not include a pressure dimension.

https://maps.gnosis.earth/ogcapi/collections/climate:era5:relativeHumidity/dggs/ISEA3H/zones/A4-0-A/data?zone-depth=8&datetime=2023-04-06T23:00:00&subset=pressure(900)


```
{
   "dggrs" : "[ogc-dggrs:ISEA3H]",
   "zoneId" : "A4-0-A",
   "depths" : [ 8 ],
   "values" : {
      "r" : [ {
         "depth" : 8,
         "shape" : { "count" : 5536, "subZones" : 5536 },
         "data" : [ 88.4226226806641, 84.0924987792969, ..., 86.5950088500977 ]
      } ]
   }
}
```


Listing

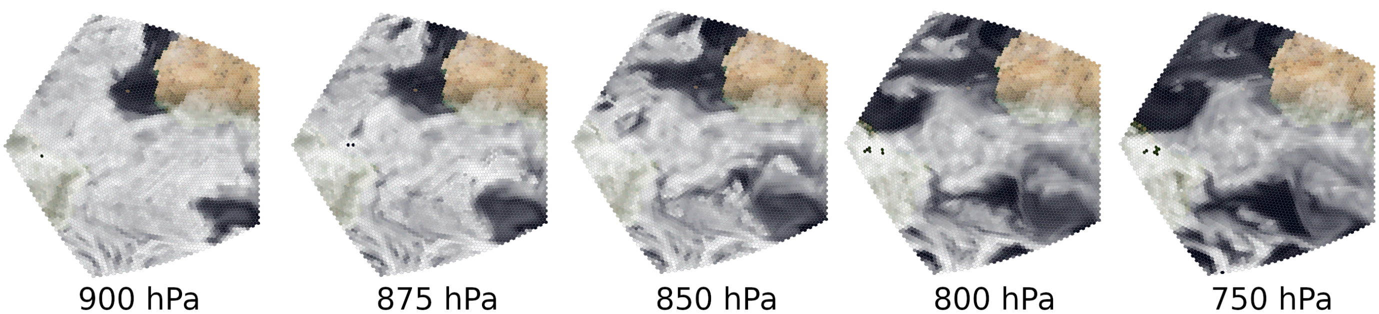


Figure C.84 — Visualization of relative humidity for ISEA3H zone A4-0-A from ERA5 hourly data on pressure levels from 1940 to present from Copernicus Climate Data Store

C.10.3. Field selection

Some data has multiple properties for a feature, or fields for a given cell or position, such as the precipitation and windspeed variables in the CMIP5 dataset illustrated above. It is possible to retrieve only some of these available fields using the `properties` query parameter. The following request retrieves only the red (`B04`), green (`B03`) and blue (`B02`) bands from a sentinel-2 dataset for ISEA3H zone G7-67252-D at a relative depth of 8.

https://maps.gnosis.earth/ogcapi/collections/sentinel2-l2a/dggs/ISEA3H/zones/G7-67252-D/data?zone-depth=8&datetime=2022-10-28&properties=B04,B03,B02

The following is a partial DGGS-JSON response for this previous request. This collection has twelve other fields available (which are not returned by default, in this case).


```
{
   "dggrs" : "[ogc-dggrs:ISEA3H]",
   "zoneId" : "G7-67252-D",
   "depths" : [ 8 ],
   "values" : {
      "B02" : [ {
         "depth" : 8,
         "shape" : { "count" : 6643, "subZones" : 6643 },
         "data" : [ 655.101806640625, 744.5323486328125, ..., 2609.4609375 ]
      } ],
      "B03" : [ {
         "depth" : 8,
         "shape" : { "count" : 6643, "subZones" : 6643 },
         "data" : [ 511.2943725585938, 653.875732421875, ..., 2645.6328125 ]
      } ],
      "B04" : [ {
         "depth" : 8,
         "shape" : { "count" : 6643, "subZones" : 6643 },
         "data" : [ 223.4036102294922, 236.1580047607422, ..., 2563.61083984375
 ]
      } ]
   }
}
```


Listing

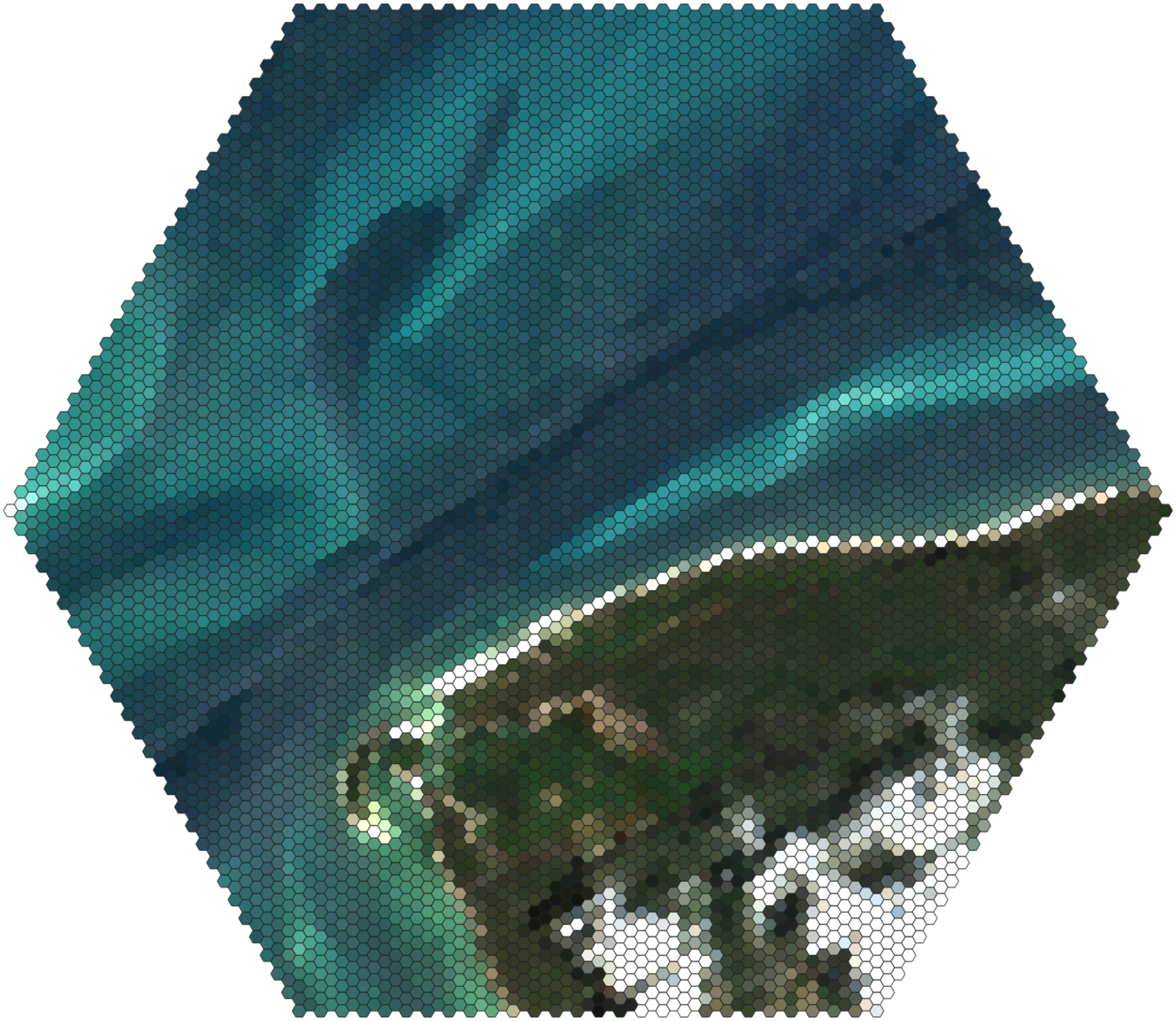


Figure C.85 — Visualization of different B04 (red), B03 (green) and B02 (blue) bands for ISEA3H zone G7-67252-D (K'gari, Australia) from Copernicus SENTINEL-2 operated by ESA

The following request retrieves only the Scene Classification Layer (`SCL`) for the same zone. This field has a lower resolution than the other bands, and could be returned at a lower depth than the other fields if requesting multiple fields at a `zone-depth` beyond the resolution of the SCL field.

https://maps.gnosis.earth/ogcapi/collections/sentinel2-l2a/dggs/ISEA3H/zones/G7-67252-D/data?zone-depth=8&datetime=2022-10-28&properties=SCL

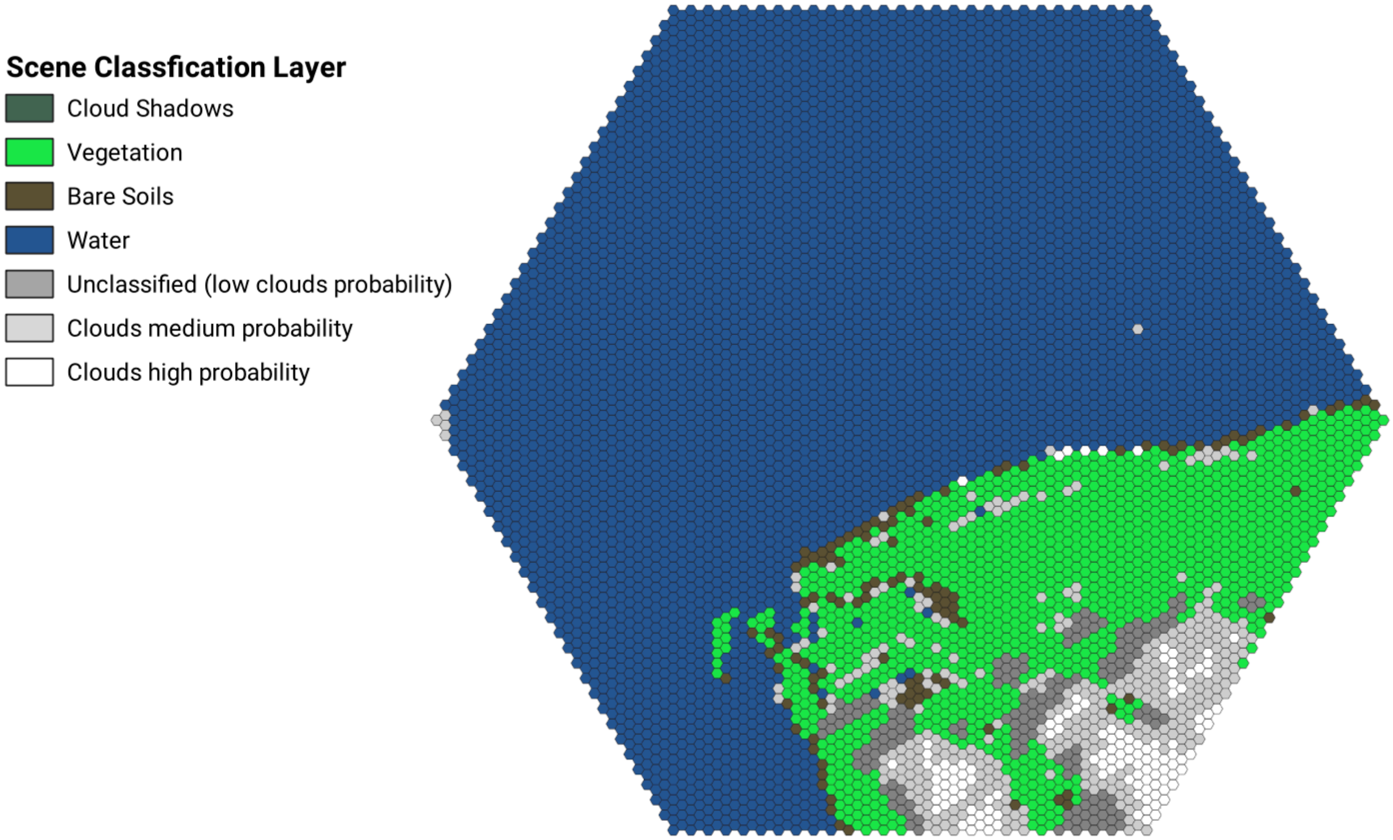


Figure C.86 — Visualization of SCL (Scene Classification Layer) field for ISEA3H zone G7-67252-D (K'gari, Australia) from Copernicus SENTINEL-2 operated by ESA

The following request retrieves only the near-infrared (`B08`) band for the same zone.

https://maps.gnosis.earth/ogcapi/collections/sentinel2-l2a/dggs/ISEA3H/zones/G7-67252-D/data?zone-depth=8&datetime=2022-10-28&properties=B08

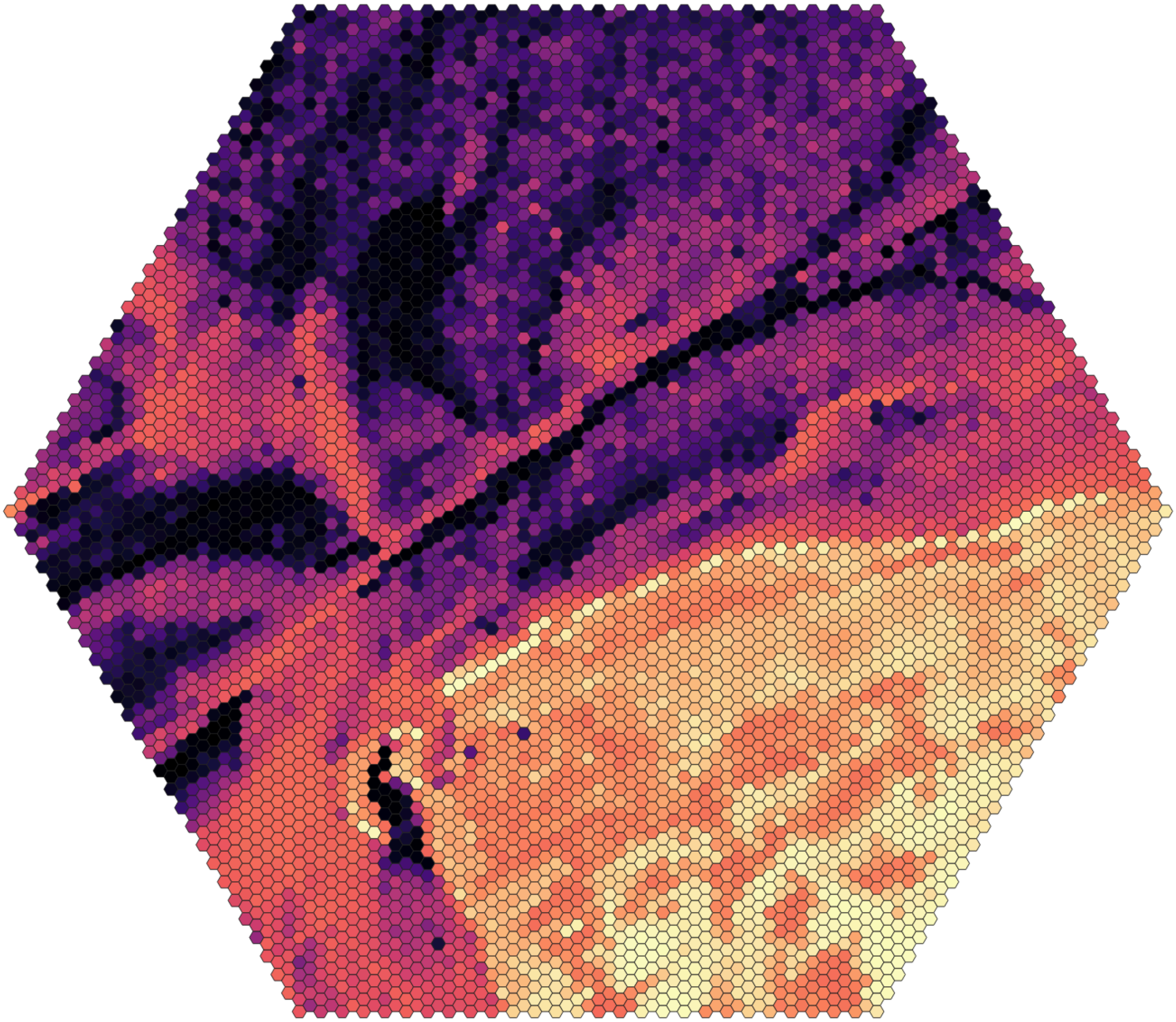


Figure C.87 — Visualization of B08 (near-infrared) band for ISEA3H zone G7-67252-D (K'gari, Australia) from Copernicus SENTINEL-2 operated by ESA

> *Although the examples illustrated here are all for rasterized data, field selection can also be applied to feature collection outputs so select which properties to include.*

C.10.4. Filtering zone data using CQL2 expressions

The `filter` query parameter can be used to only include data matching a CQL2 expression, which can reference the queryables of the collections, such as the feature properties or fields.

In a DGGS-JSON output, sub-zones which are not a match get replaced by a `null` value. In a GeoTIFF output, pixel values get replaced by a NODATA value.

Although the examples illustrated here are all for rasterized data, a CQL2 expression filter can also be applied to feature collections to select which feature to include in the output.

The following request retrieves bathymetry data for ISEA3H zone A8-0-F at a relative depth of 8 without a filter.

https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA3H/zones/A8-0-F/data?zone-depth=8

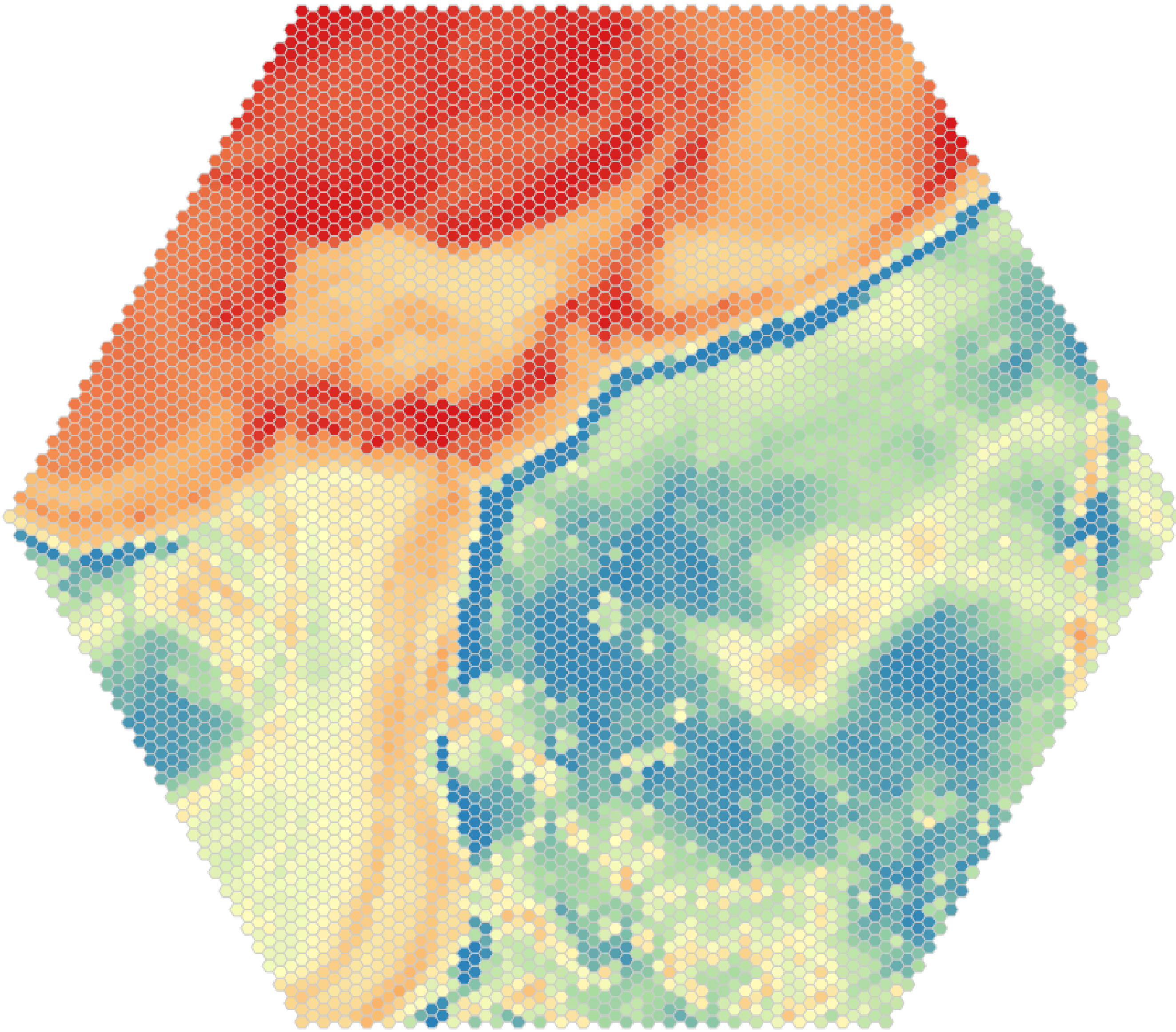


Figure C.88 — Visualization of ISEA3H zone A8-0-F from GEBCO 2014 bathymetry without a filter

The following request retrieves bathymetry data for the same zone with a filter for sea depths greater than 4000 meters.

https://maps.gnosis.earth/ogcapi/collections/gebco/dggs/ISEA3H/zones/A8-0-F/data?zone-depth=8&filter=Elevation < -4000

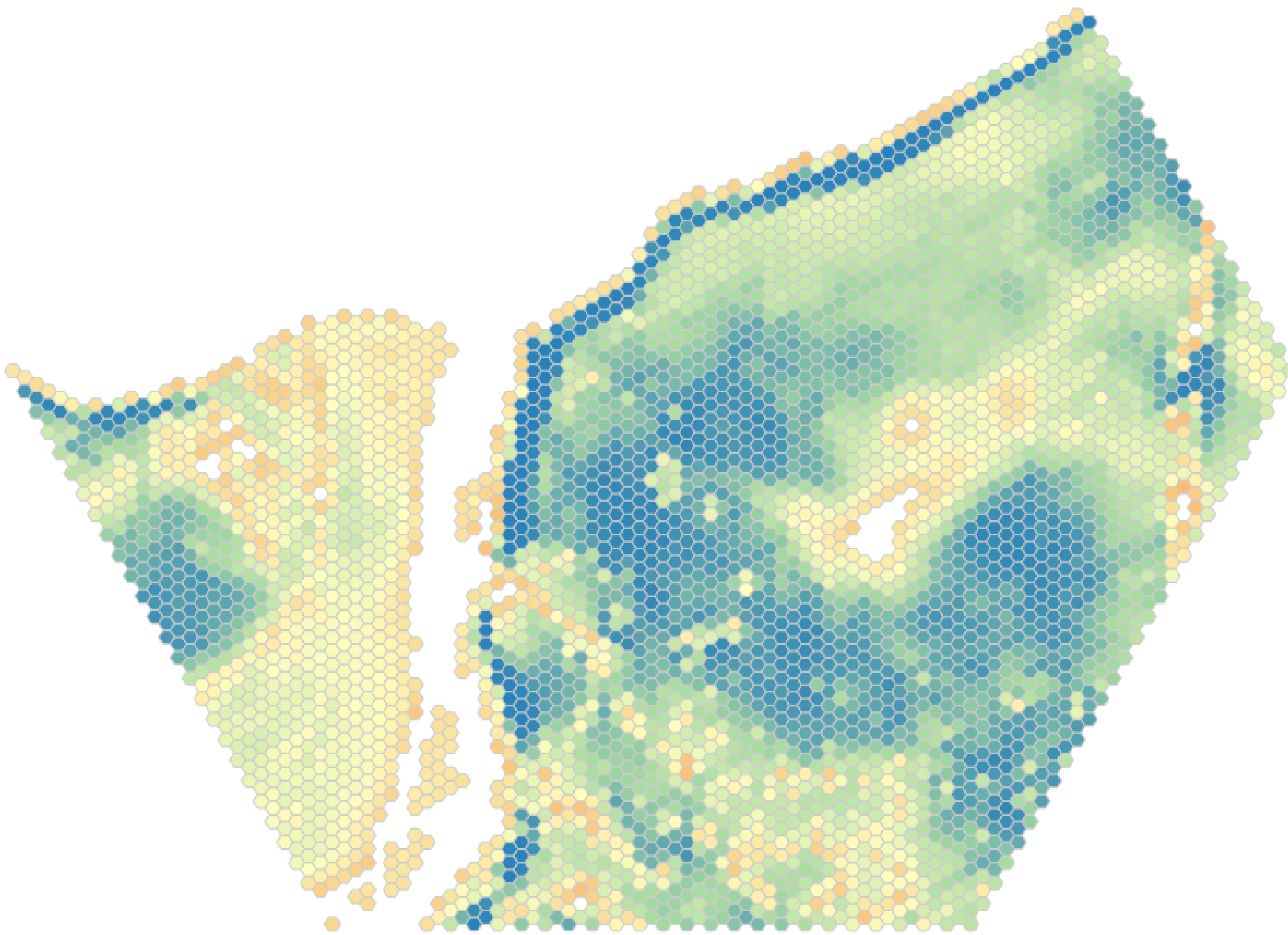


Figure C.89 — Filtering of ISEA3H zone A8-0-F from GEBCO 2014 bathymetry for depths 4000 meters below sea level

When retrieving a temporal slice with a filter, the server may fill in the filtered out sub-zones with earlier imagery not excluded by the filter. This can be used for example to replace cloudy sub-zones with the closest earlier cloud-free imagery. The following requests retrieve sentinel-2 data for zone G7-67252-D between August 1, 2022 and August 31, 2022 at a relative depth of 12 with:

no filter:

https://maps.gnosis.earth/ogcapi/collections/sentinel2-l2a/dggs/ISEA3H/zones/G7-67252-D/data?zone-depth=12&datetime=2022-08-01/2022-08-31

filtering only high probability clouds (SCL: 9):

https://maps.gnosis.earth/ogcapi/collections/sentinel2-l2a/dggs/ISEA3H/zones/G7-67252-D/data?zone-depth=12&datetime=2022-08-01/2022-08-31&filter=SCL <> 9

also filtering out medium probability clouds (SCL: 8):

https://maps.gnosis.earth/ogcapi/collections/sentinel2-l2a/dggs/ISEA3H/zones/G7-67252-D/data?zone-depth=12&datetime=2022-08-01/2022-08-31&filter=SCL <> 8 and SCL <> 9

also filtering out cirrus clouds (SCL: 10):

https://maps.gnosis.earth/ogcapi/collections/sentinel2-l2a/dggs/ISEA3H/zones/G7-67252-D/data?zone-depth=12&datetime=2022-08-01/2022-08-31&filter=SCL not in (8, 9, 10)

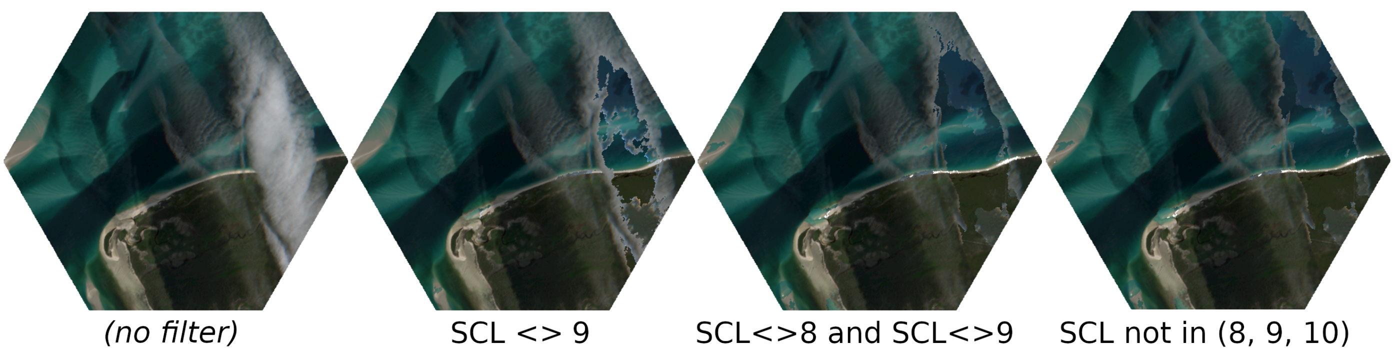


Figure C.90 — Filtering out high probability clouds, medium probability clouds, and cirrus clouds for ISEA3H zone G7-67252-D (K'gari, Australia) from Copernicus SENTINEL-2 operated by ESA

The following request specifies a more complex filter using CQL2 arithmetic expressions to filter for a normalized difference vegetation index (NDVI) greater than 0.5, computed using the near-infrared (`B08`) and red bands (`B04`) using the expression `(B08-B04)/(B08+B04)`:

https://maps.gnosis.earth/ogcapi/collections/sentinel2-l2a/dggs/ISEA3H/zones/G4-74A6D-E/data?zone-depth=9&filter=(B08-B04)/(B08+B04)>0.5

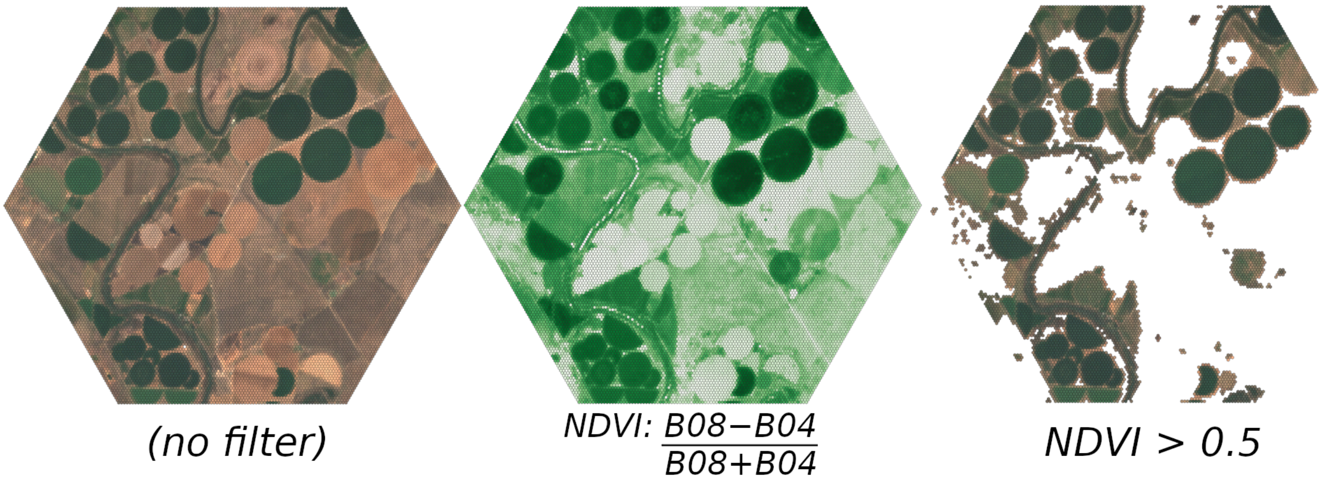


Figure C.91 — Filtering based on a minimum normalized difference vegetation index (NDVI) for ISEA3H zone G4-74A6D-E from Copernicus SENTINEL-2 operated by ESA

D

ANNEX D (INFORMATIVE) REVISION HISTORY

D ANNEX D (INFORMATIVE) REVISION HISTORY

DATE	RELEAS	EDITOR	PRIMARY CLAUSES MODIFIED	DESCRIPTION
2021-05-17	0.1	Matthew Purss	all	initial version
2022-07-22	0.3	Jérôme St-Louis	all	Renamed Part 1 to core (316d37f) Initial set up of conformance classes clauses (b76073c)
2022-07-26	0.4	Gobe Hobona	all	Fixes problem that was breaking auto-build (a13c2a8)
2022-10-04	0.5	Jérôme St-Louis	7-Data Retrieval	Initial take at Data Retrieval Conformance Class (b67f290)
2022-11-11	0.10	Jérôme St-Louis	all	Split multi-zone retrieval into separate conformance class (06a4260). Initial progress on Zone Query conformance class (a208502) Added preface, abstract, Jérôme added as editor (71e76c6) Clarification regarding resolution (c43bfad) Clarifications (cee3685)
2023-02-17	0.11	Jérôme St-Louis	7-Data Retrieval	Added `zone-depth` parameter (2b8fa58)
2024-06-18	0.12	Jérôme St-Louis	all	Draft ready for presentation at Montréal Members Meeting
2024-08-24	1.0.0rc1	Jérôme St-Louis	all	Complete Zone Data examples. New Zarr requirements class. Clarify DGGS/DGGH terminology and related DGGRS schema adjustments. New DGGS-(UB)JSON-FG encoding for efficient vector format with coordinates quantized to sub-zones. `filter` parameter for zone data. Initial draft ready for OAB review.

DATE	RELEAS	EDITOR	PRIMARY CLAUSES MODIFIED	DESCRIPTION
2024-09-20	1.0.0rc2	Carl Reed, Jérôme St-Louis	all	Applied suggested edits from Carl Reed's review. Draft ready for public comment period.
2024-10-16	1.0.0rc3	Jérôme St-Louis	all	Shift ISEA orientation 0.05° West for ISEA3H and ISEA9R DGGRS to avoid vertices on land (using geodetic/authalic latitude conversion). Other fixes and improvements.
2025-02-13	1.0.0rc4	Jérôme St-Louis, Matthew Purss, Michael Jendryke	all	Completed addressing issues raised by Carl. Addressed issues received during RFC period. Completed Abstract Test Suite. Merged JSON Zone List encoding requirements classes into Zone Query. Merged GeoJSON/JSON-FG requirements classes. Added `profile` query parameter and defined profiles to distinguish encoding variations using same media type. Created a new overview section with some content from the examples annex, and with a new table illustrating the paths, responses and parameters, as suggested by João. Additional fixes and improvements. Draft ready for final TC vote and publication.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Robert Thomas, Sara Saeedi: OGC 23-010, *Towards a Federated Marine SDI: Connecting Land and Sea to Protect the Arctic Environment Engineering Report*. Open Geospatial Consortium (2023). http://www.opengis.net/doc/PER/FMSDI3.

[2] Robert Gibb, Byron Cochrane, Matthew Purss: OGC 20-039r2, *OGC Testbed-16: DGGS and DGGS API Engineering Report*. Open Geospatial Consortium (2021). http://www.opengis.net/doc/PER/t16-D017.

[3] Jérôme Jacovella-St-Louis, Flexible real-time data processing and visualization workflows emerging from OGC API modules (Draft), OGC 21-033 (2025). https://maps.gnosis.earth/moaw/DiscussionPaper-Draft3.pdf

[4] Universal Binary JSON Specification. https://ubjson.org/

[5] Michael F. Goodchild, Yang Shiren, A hierarchical spatial data structure for global geographic information systems, CVGIP: Graphical Models and Image Processing, Volume 54, Issue 1, Pages 31-44, ISSN 1049-9652 (1992). https://doi.org/10.1016/1049-9652(92)90032-S

[6] John P. Snyder, An Equal-Area Map Projection for Polyhedral Globes, Cartographica, 29(1),10-21. urn:doi:10.3138/27H7-8K88-4882-1752 (1992). https://doi.org/10.3138/27H7-8K88-4882-1752

[7] Kevin Sahr, Denis White, A. Jon Kimerling, Geodesic Discrete Global Grid Systems, Cartography and Geographic Information Science, Vol. 30, No. 2, pp. 121-134 (2003). https://cs.sou.edu/~sahrk/dgg/pubs/gdggs03.pdf

[8] Ali Mahdavi Amiri, Faramarz Samavati, Perry Peterson, Categorization and Conversions for Indexing Methods of Discrete Global Grid Systems, ISPRS International Journal of Geo-Information 4, no. 1: 320-336 doi: 10.3390/ijgi4010320 (2015). https://www.mdpi.com/2220-9964/4/1/320

[9] Robert Gibb, The rHEALPix Discrete Global Grid System. IOP Conference Series: Earth and Environmental Science. 34. 012012. 10.1088/1755-1315/34/1/012012 (2016). https://dx.doi.org/10.1088/1755-1315/34/1/012012

[10] Perry R. Peterson, Idan Shatz, An Application Development Framework for Open Geospatial Consortium Discrete Global Grid System Standard, 2019 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Yokohama, Japan, pp. 5668-5670, doi: 10.1109/IGARSS.2019.8898394 (2019). https://ieeexplore.ieee.org/document/8898394