

OGC® DOCUMENT: 24-039

External identifier of this OGC® document: <http://www.opengis.net/doc/PER/t20-D012>



Open
Geospatial
Consortium

OGC TESTBED 20 COVERAGE FORMAT SELECTION REPORT

ENGINEERING REPORT

DRAFT

Submission Date: 2024-12-31

Approval Date: 2024-12-31

Publication Date: 2025-01-30

Editor: Joan Masó Pau, Núria Julià Selvas

Notice: This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is *not an official position* of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard.

Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements. However, the discussions in this document could very well lead to the definition of an OGC Standard.

License Agreement

Use of this document is subject to the license agreement at <https://www.ogc.org/license>

Copyright notice

Copyright © 2025 Open Geospatial Consortium

To obtain additional rights of use, visit <https://www.ogc.org/legal>

Note

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

CONTENTS

I. OVERVIEW	vi
II. EXECUTIVE SUMMARY	vii
III. KEYWORDS	vii
IV. CONTRIBUTORS	vii
V. FUTURE OUTLOOK	viii
VI. VALUE PROPOSITION	viii
1. INTRODUCTION	2
1.1. About this Report	2
1.2. Aims	2
1.3. Objectives	2
2. USE CASES AND RECOMMENDATIONS FOR COVERAGE SELECTION	4
2.1. Introduction	4
2.2. Panchromatic or grey scale image	5
2.3. Photographic landscape (RGB image neither georeferenceable nor georectified)	6
2.4. Multiband satellite image (georeferenceable)	6
2.5. Hyperspectral satellite image	7
2.6. Orthophotography (georectified)	7
2.7. Stereogram image	8
2.8. Temperature image (continuous cell property)	8
2.9. Land cover classification image (categorical cell property)	8
2.10. Weather variables images (surface temperature, relative humidity...)	9
2.11. Weather forecast	9
2.12. Atmosphere characterization (height)	12
2.13. Atmosphere evolution (height + time)	12
2.14. Weather forecast runs (modeling ensemble)	13
2.15. Crop yield prediction as a function of level of fertilizers and species and plant variety (modeling outputs depending on several input parameter combination)	15
2.16. Browsing (pan, zoom, clip...) through high resolution and large extent image in the Web or in the Cloud	15
2.17. Confidential or private image (metadata)	18
2.18. Sparse images	19
2.19. Planetary image (extraterrestrial CRS)	20
2.20. Satellite images time series	20

2.21. Burst Photography	21
2.22. Videos from a camera with a fixed position	21
2.23. Videos from a moving camera	22
2.24. Videos from multiple cameras	22
2.25. Videos with a depth sensor	23
2.26. Videos with reconstructed 3D point cloud	23
3. OUTLOOK	25
4. SECURITY, PRIVACY AND ETHICAL CONSIDERATIONS	27
BIBLIOGRAPHY	29
ANNEX A (NORMATIVE) ABBREVIATIONS/ACRONYMS	31
ANNEX B (INFORMATIVE) SILVEREYE TECHNOLOGY – COVERAGE FORMAT	33
B.1. Executive Summary	33
B.2. Tiling	33
ANNEX C (INFORMATIVE) BENCHMARKING – APPROACHES AND CONSIDERATIONS	36
C.1. Benchmarking: Strategy for evaluating tiled images	36
C.2. List of benchmarking procedures implemented	37
ANNEX D (INFORMATIVE) SECTIONS OF THE HEIF TO READ IN A PARTIAL READING TILED IMAGE	40

LIST OF TABLES

Table – Editors and Contributors	vii
--	-----

LIST OF FIGURES

Figure 1 – Still image and its metadata related by ItemReferenceBox. Picture taken from NGA 0076 v0.6 and simplified to show only the accessory metadata	14
Figure 2 – Extracting the needed amount in 3 resolutions. (The red square represents the actual needed data for the screen, and the green square represent the minimum data to be transferred from the server)	17
Figure 3 – Media Configuration and ISM labeling for the XML code example. Picture taken from NGA 0076 v0.6	18
Figure 4 – File structure of a tili image item. Picture taken from OGC 24-040	20

Figure C.1 – Setup to benchmark imagery	37
Figure D.1 – how to read a HEIF file using HTTP range	41



OVERVIEW

This OGC Testbed 20 Report provides an overview of use cases where the HEIF format can be used. For each use case, the different items and properties provided by the HEIF specification or by the proposed GeoHEIF extensions are selected.

The ISO Base Media File Format (ISOBMFF) is a container file format that defines a general structure for files that contain time-based multimedia data such as video and audio. The basic structure is a sequence of “boxes”, where each box structure starts with the same header structure, including a box identifier, then content according to the requirements of that box. Some boxes can be nested. Typically, the body of the outer box is just a sequence of inner boxes.

HEIF is a container file format, defined by Moving Picture Experts Group (MPEG), for storing multimedia files, including images, audio, video, and timed-text stream, that compresses images into smaller files while maintaining high quality. HEIF is an alternative to traditional image formats like JPEG and PNG and is commonly used on Apple devices. The GeoHEIF specification defines HEIF properties to include the georeference information in the file and to define the semantics of dimensions and the cell properties that together enable converting HEIF into a datacube container.

A “MetaBox” (*meta*) is essentially a container within the file structure that holds metadata about the image(s) or image sequences the file contains. This metabox provides information about the items stored within the file, such as their type, coding, and other relevant details.

The *meta* nested boxes include:

- The *ItemInfoBox*, which identifies the various items that are present,
- The *ItemLocationBox*, which identifies the location of the encoded data (often by byte offsets into the file),
- The *ItemPropertiesBox*, which provides descriptive and transformative properties.

The structure of the *ItemPropertiesBox* is broken out showing the *ItemPropertyContainerBox* (*ipco*) and *ItemPropertyAssociationBox* (*ipma*). The item properties are contained in the *ItemPropertyContainerBox*, concatenated one after another. The item properties are associated with the image item by an entry in the *ItemPropertyAssociationBox*. Item properties generally describe how to interpret the encoded data to reconstruct the required image. Transformative properties change the resulting image, and include operations such as mirroring, rotation and scaling. GeoHEIF item properties are used in the same way as other item properties.

This document does not provide more details on how HEIF and GeoHEIF are internally structured. In order to understand the discussion and recommendations in this Report, please read the ISOBMFF, HEIF and GeoHEIF specifications.



EXECUTIVE SUMMARY

The OGC Testbed 20 Coverage Format Selection Report (this document) provides an overview of use cases where the High Efficiency Image File Format (HEIF) can be used. For each use case, the different items and properties provided by the HEIF specification or by the proposed Geographic High Efficiency Image Format (GeoHEIF) extensions are selected. The use cases start with simple ones such as panchromatic or grey scale image and extend beyond photographic images into model outputs such as land cover classifications, atmosphere evolution, and weather forecast runs. Some other use cases consider actions such as map browsing (pan, zoom, clip...), or special cases such as confidentiality or extraterrestrial imagery.

The three annexes in this Report include different points of view and considerations for how to do benchmarking of tiled imagery that can be the basis for a practical benchmarking future exercise.

The use case exercise was limited to still imagery and time series. In the future the set of use cases should be extended to motion imagery.



KEYWORDS

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

HEIF, Image File Format, ISOBMFF, GeoHEIF, use cases, benchmarking



CONTRIBUTORS

All questions regarding this document should be directed to the editors or the contributors:

Table – Editors and Contributors

NAME	ORGANIZATION	ROLE
Joan Masó Pau	University Autonomous of Barcelona, Spain / CREAM	Editor
Núria Julià Selvas	University Autonomous of Barcelona, Spain / CREAM	Editor
Brad Hards	Silvereye Technology	Contributor

NAME	ORGANIZATION	ROLE
Dirk Farin	Dirk Farin Algorithmic Research	Contributor



FUTURE OUTLOOK

In the future, a comparison of the GeoHEIF selected uses cases with other file formats can be conducted thereby extending the scope of the coverage format selection decision process. A future benchmarking exercise could be based on comparing HEIF with other formats and demonstrating the advantages and disadvantages of the format for geospatial data. In OGC Testbed 20, the use case exercise was limited to still imagery, time series and image sequences. In the future the set of use cases should be extended to motion imagery. In addition, how to store the position of the camera should be further elaborated.

Both access to the local drive files as well as to HEIF files accessed through HTTP range header request over the web (file formats optimized for web access are commonly known as cloud optimized formats) were considered and could be implemented and compared in a future OGC Testbed.

During OGC Testbed 20, no properties was proposed for the security information for an image. In OGC Testbed 21 a way to security label the information and methods for data encryption should be considered.



VALUE PROPOSITION

This Report documents the versatility of the HEIF format to support a range of use cases starting from simple ones such as panchromatic or grey scale image up to model outputs such as land cover classifications, atmosphere evolution, and weather forecast. The Report also considers the applicability of the HEIF format for georeferenced images. In some contexts the HEIF format can be a better alternative to GeoTIFF or NetCDF in some contexts.

1

INTRODUCTION

1.1. About this Report

GeoHEIF is a container file format based on HEIF and ISOBMFF that compresses images into smaller files while maintaining high quality and provides an alternative to traditional image formats. The flexibility of the HEIF format supports different methods for storing images and their metadata. However, not all methods will be efficient and interoperable.

This Report provides a list of use cases where HEIF format can be used and how the data and the metadata can be structured. The use cases are based on discussions and testing done in OGC Testbed 20. Some cases describe how to store different image types such as panchromatic or grey scale image and extend beyond photographic images. Others are more applicable to model outputs such as land cover classifications, atmosphere evolution, and weather forecast runs. Some other use cases consider actions such as map browsing (pan, zoom, clip...) and the use of tiles and overview as well as confidentiality or extraterrestrial imagery. Finally, a set of use cases related with images sequences is provided and described.

1.2. Aims

The focus of the OGC Testbed 20 GIMI Coverage format selection activity was to enumerate HEIF applications and select among the characteristics (items and properties) that need to be used from the long list of existing and proposed characteristics to improve the efficiency and the interoperability of the image format.

1.3. Objectives

This Report collects the use cases discussed during the development and benchmarking activities conducted in OGC Testbed 20 that were focused on the HEIF and the GeoHEIF extensions.



2

USE CASES AND RECOMMENDATIONS FOR COVERAGE SELECTION

USE CASES AND RECOMMENDATIONS FOR COVERAGE SELECTION

2.1. Introduction

This section provides an overview of use cases where the HEIF format can be used. For each case, the different items and item properties provided by the HEIF specification or by the proposed GeoHEIF extensions are selected, and an explanation on how to combine the items and the item properties is provided.

This is the list of use cases that are analyzed in this Report:

1. Panchromatic or grey scale image
2. Photographic landscape (RGB image neither georeferenceable nor georectified)
3. Multiband satellite image (georeferenceable)
4. Hyperspectral satellite image
5. Orthophotography (georectified)
6. Stereogram image
7. Temperature image (continuous cell property)
8. Land cover classification (categorical cell property)
9. Weather variables images (surface temperature, relative humidity...)
10. Atmosphere characterization (height)
11. Atmosphere evolution (height + time)
12. Weather forecast runs (Modeling ensemble): metadata, provenance, workflow,...
13. Crop yield prediction as a function of level of fertilizers and species and plant variety (modeling outputs depending on several input parameter combination)
14. Browsing (pan, zoom, clip...) through high resolution and large extent image in the Web or in the Cloud: random access, tiles, uncompressed image,...
15. Confidential or private image: metadata
16. Sparse images
17. Planetary image (extraterrestrial CRS)

18. Satellite images time series
19. Burst Photography (images sequences)
20. Videos from a camera with a fixed position
21. Videos from a moving camera
22. Videos from multiple cameras
23. Videos with a depth sensor
24. Videos with reconstructed 3D point cloud

The following subsections describe each use case. Please note that the cases are based on the capabilities of the GeoHEIF format defined in the OGC Testbed 20 GIMI Specification Report (OGC 24-038) (e.g., datacube) and in the GIMI Lessons Learned and Best Practices Report (OGC 24-040) (e.g., tiling). Many of these “capabilities” are newly defined and were not implemented in current HEIF related libraries (such as libheif) at the time of this Report was written. In particular, the properties `edim` and `pcel` to define extra-dimensions and cell properties were not available. Support for non-integer data types might not exist yet (so currently, cell properties such as temperature and wind speed such be scaled and rounded to integers).

2.2. Panchromatic or grey scale image

Panchromatic or grey scale imagery is one of the simplest cases. The HEIF file has an item containing a **single channel** image.

When using ISO/IEC 23001-17 (uncompressed codec), this is a component type 0 (monochrome). In JPEG-2000, the image is stored as a one channel image. When using video-oriented codecs (h.264, h.265, h.266, AV1) and JPEG, this is a YCbCr image with chroma format 4:0:0 (i.e., luminance without color information).

These various encodings may be abstracted away. For example, in the libheif API, all cases are mapped to images with a `heif_colorspace_monochrome` channel.

Additionally, if the image contains geospatial information, the `ogeo` brand must be included in the `FileTypeInfoBox (ftyp)` to signal that the file structure follows the GeoHEIF. In this case, the coordinate reference system (CRS) is defined using the `mcrs` item property in the `ItemPropertyContainerBox`, and, if the image is georeferenced, the matrix transformation model is defined using the `mtx` item property in the `ItemPropertyContainerBox`.

2.3. Photographic landscape (RGB image neither georeferenceable nor georectified)

Photographic landscape is another simple use case. The HEIF file has an item containing a **three-channel** image (a.k.a. three compound image). This is typically in a format that supports RGB internally, such as JPEG-2000 or ISO/IEC 23001-17, or a format that is readily converted to RGB. For example, video codecs will usually store color images not in the RGB space but first transform those images into a YCbCr color space because that results in better compression ratios. Often, chroma channels are additionally subsampled to a lower resolution. However, images can be stored in native RGB by storing a `nc1x` color profile in the metadata with a `matrix_transform=0`. This stores the green channel in the Y (luminance) channel and red, blue in the two chroma channels without subsampling (chroma 4:4:4). JPEG always uses YCbCr internally.

2.4. Multiband satellite image (georeferenceable)

In the HEIF format, multiband satellite images can be structured in two ways. The HEIF file can have an image item in a format that internally supports multiple bands or can have multiple image items, one item for each band. The `ogeo` brand must be included in the `FileTypeInfoBox (ftyp)` to indicate that the file contains geospatial information and to indicate a GeoHEIF file. In a GeoHEIF file, the images can be georeferenceable by associating them to the CRS in the `mcrs` item property and to a set of **tie-points** in the `tiep` item property, both in the `ItemPropertyContainerBox` and related to the image items in the `ItemPropertyAssociationBox`. The fact that some images share the same associations is an indication that they are part of the same multiband image. The description of the **characteristics of each band** can be added in the property `CellPropertyTypeProperty (pcl)` in the `ItemPropertyContainerBox`. There is a cell property type property for each image item of a different type. The specifics of the description of the band, such as the maximum and minimum frequency of the band, are expected to be found by following the 'definition' uri.

When a compressing video codec is used, it is necessary to store each band in a separate image as described in the greyscale case above because these codecs do not natively support multiple bands. When using JPEG-2000 or ISO/IEC 23001-17, it is possible to store the bands as separate, custom image components within the same image.

NOTE: A hyperspectral image is a different use case where the sensor frequency becomes an extra dimension instead of a cell property.

2.5. Hyperspectral satellite image

Multiband and hyperspectral imaging are two distinct remote sensing technologies that differ primarily in their spectral resolution and number of bands captured. Multiband imaging captures data in a limited number of broader spectral bands, typically 3 to 20. These bands are discrete and separated, focusing on specific wavelengths of interest. Multiband sensors commonly capture visible light (RGB) and a few infrared bands. In contrast, hyperspectral imaging captures hundreds to thousands of narrow, contiguous spectral bands. This results in a much higher spectral resolution, allowing for more precise identification and characterization of materials.

Hyperspectral imagery can be arranged as a datacube. A datacube is a multi-dimensional (“n-D”) array of values designed for efficient data analysis and querying. The data is arranged in dimensions and cell properties. Dimensions can be coordinates, other continuous variables or even enumerations, e.g., categories.

In the case of hyperspectral imagery, the **datacube** is a **3D** case with two spatial dimension and an extra dimension representing the narrow radiometric frequency interval of the spectral bands.

The third dimension is the spectral dimension that is represented by **several image items**, at least one for each of the N radiometric frequency values in the spectral dimension.

The third dimension, spectral, is defined as an **extra-dimension** in a `ExtraDimensionProperty` (`edim`) and the **extra-dimension values** is defined in `ExtraDimensionValueProperty` (`edvl`) as the central frequency of the interval. All images are associated with the **same cell property type property** (`pcel`). This property contains the description of the intensity of light in each radiometric frequency interval.

2.6. Orthophotography (georectified)

Orthorectified imagery is a special case of a georectified image. In addition to the georeference process, orthorectification removes distortions caused by terrain variations, camera angle, and other factors. The resulting orthorectified image appears as if each cell were captured from directly above (vertical).

As in the case above, the HEIF file can have an image item in a format that internally supports multiple bands or can have multiple image items, one item for each band. The `ogeo` brand must be included in the `FileTypeBox` (`ftyp`) to indicate that the file contains geospatial information and to indicate a GeoHEIF file. In a GeoHEIF, the images can be georectified by associating them to the **CRS** in the `mcrs` item property and by defining the **affine transformation matrix** in the `mtx` item property. These are both in the `ItemPropertyContainerBox` and related to the image items in the `ItemPropertyAssociationBox`. If the components of the image are only “colors” the association to a cell property type property (`pcel`) may not be necessary.

2.7. Stereogram image

In a HEIF file, a stereogram image has **two images** with the same timestamp. The `ogeo` brand must be included in the `FileTypeInfoBox (ftyp)`. In GeoHEIF, the images should have the same association to the CRS in the `mcrs` item property and the two images should have their respective association to a set of **tie-points** in the `tiep` item property. The two sets of tie points should define two areas that have a significant overlap.

The fact that it is a stereo image pair is indicated by adding an `EntityGroup` of type `ster` with the first entity ID identifying the left image and the second entity ID referencing the right image. HEIF also includes the item properties `cmIn` and `cmex` for specifying the camera intrinsic and extrinsic parameters that describe the spatial relationship between the two cameras.

2.8. Temperature image (continuous cell property)

In a HEIF file, a temperature image has an image item in a format that supports **floating point** numbers. This is currently only possible with the ISO/IEC 23001-17 codec.

The `ogeo` brand must be included in the `FileTypeInfoBox (ftyp)`. In a GeoHEIF, the image is georectified by associating it to the CRS in the `mcrs` item property and by defining the affine transformation matrix in the `mtx` item property. The description of the temperature cell property (or any other **continuous cell property**) should be added as an association to a cell property type property (`pcel`). The definition URI should point to the definition of temperature such as: <https://qudt.org/vocab/quantitykind/Temperature>. The `unit`, should point to the units of measure of the temperature such as https://qudt.org/vocab/unit/DEG_C and the `unitLang` such as <https://qudt.org/vocab/unit/>.

2.9. Land cover classification image (categorical cell property)

For a land cover land classification image, a HEIF file has an image item, typically with unsigned short integer values (`uint8`). While in theory using video codecs for this image type should be possible, using the ISO/IEC 23001-17 codec is advised because this scheme by definition is lossless. Even though ISO/IEC 23001-17 is often called the 'uncompressed' codec, it in fact supports lossless compression using the 'deflate' or 'brotli' algorithms. Brotli compression is a lossless data compression algorithm that reduces the size of files such as HTML, CSS, and JavaScript. It is primarily used to make web traffic to make it faster and more efficient to transmit information over the internet. Some video codecs that provide lossless compression can be used. However, great care must be taken to set the parameters correctly. This is because these codecs are not primarily designed for lossless operation and this part is often overlooked.

The ogeo brand must be included in the FileTypeInfoBox (ftyp)... In a GeoHEIF file, the description of the **categorical cell property** should be added as an association to a cell property type property (pcel) (e.g., “Land cover”); the definition URI should point to the definition of land cover such as: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Glossary:Land_cover. Each possible land cover class should be provided as an association to a cell property category property (pcat) where the **list** of possible land cover **classes** are listed (e.g., “forest”, “farm”, “water”...).

2.10. Weather variables images (surface temperature, relative humidity...)

Images such as surface temperature are a typical case of a continuous cell properties **multiband** image. As with the **multiband** case above, the HEIF file either has **several image items**: One for each weather variable typically in a format that supports **floating point** numbers (i.e., ISO/IEC 23001-17), or it contains a single ISO/IEC 23001-17 with separate components for the physical parameters. The ogeo brand must be included in the FileTypeInfoBox (ftyp). In a GeoHEIF file, all images are georectified by associating them to the same CRS in the mcrs item property and to the same affine transformation matrix in the mtxf item property. The description of the **continuous cell properties** is different for each cell property.

NOTE: The weather variables images are the variables measured (or stored) in each cell of the grided space and these variables are called **cell properties** in the datacube model defined in OGC 24-038. Another type of properties is related to the items such as mcrs. These properties are defined in the ItemPropertyContainerBox and are **item properties**.

2.11. Weather forecast

The Global Forecast System (GFS) is a global numerical weather prediction system containing a global computer model and variational analysis run by the United States’ National Weather Service (NWS). The model, run four times a day, generates data for dozens of atmospheric and land-soil variables, including temperatures, winds, precipitation, soil moisture, and atmospheric ozone concentration. The system couples four separate models (atmosphere, ocean model, land/soil model, and sea ice) that work together to accurately depict weather conditions. This is an example of the data that comes out of numerical forecasting: <https://www.nco.ncep.noaa.gov/pmb/products/gfs/gfs.t00z.pgrb2.0p25.f003.shtml>

Numerical forecasts utilize two kinds of times:

- The time that the forecast was run (the weather model software a few times a day using observations up to time the run starts).

- The future time being forecasted for. The forecast might predict out for a few days or a week, typically in something like 1 hour increments.

2.11.1. Subcase 1. Temperature – single altitude (“surface”), multiple time predictions across a single forecast run.

As an example, assume that the forecast was performed at 06Z (a.k.a. 0600 UTC) and has 24 predictions for incremental times (0600 UTC, 0700 UTC, 0800 UTC ... 0500 UTC). Typically, the early predictions are in the past by the time the results get published.

This can be modeled as a datacube in 2 ways:

- Each prediction time is an image resulting in 24 image items.
- The set of predictions becomes an image sequence: one track, with 24 frames in it.

The description of the temperature cell property (or any other **continuous cell property**) should be added as an association to a cell property type property (`pcel`). The definition URI should point to the definition of temperature such as: <https://qudt.org/vocab/quantitykind/Temperature>. The unit, should point to the units of measure of the temperature such as https://qudt.org/vocab/unit/DEG_C and the unitLang such as <https://qudt.org/vocab/unit/>.

For the image items case, there are 2 time dimensions:

- The time of the forecast was run: The 06Z run and any other runs that we can store in the same file.
- The set of times of the predictions: in our example the 24 predictions for each run.

In this case it is easy to find all predictions for the exact same hour in the different runs traversing the cube. Both times are defined as an **extra-dimension** in a `ExtraDimensionProperty` (`edim`) and the **values** that this extra-dimension takes are defined in `ExtraDimensionValueProperty` (`edvl`). All images are associated to the **same cell property type property** (`pcel`) which contains the description of the array of the atmosphere parameters measured.

For the image sequence case, the track provides a grouping for each run.

2.11.1.1. Consideration on other ways to encode the time

Instead of the existing timestamping mechanisms commonly used, the new International Atomic Time (TAI) timestamps (in ISO 23001-17 Amd 1, copied into Annex D of the GIMI spec) offer an alternative. There is also a way to specify “fuzzier” timestamps in ISO/IEC 23000-10 (Surveillance). For the forecast run identification, creation time for the item could be used (i.e., `crтт` descriptive property) or the image sequence track (i.e., the `tkhd` box that contains information related to the track properties as defined in the ISOBMFF standard). Alternatively, if there is more metadata for the run (e.g., the software version, or the forecast center), the time

could be provided as general metadata (using file level or track level MetaBox as appropriate) and associated to the track or item using `cdsc` (content describes) referencing if needed.

The grouping for image items could be done using an entity group (ISO/IEC 23008-12 Section 6.8), like the way an image burst, stereo pairs, or slideshow could be done. A grouping type is defined and shared for every item in the group.

2.11.2. Subcase 2. Temperature – single time prediction from a single forecast run – multiple altitudes.

Similar to subcase 1, but instead of 24 hourly surface predictions, specific time (say 1200Z), and altitude changes are of interest. The GFS output has 50+ temperature layers corresponding to different altitudes (e.g., “surface”, “2 m above ground”, “80 m above ground”, “100 m above ground”, “1829 m above mean sea level”, “2743 m above mean sea level”, “tropopause”, “max wind”, “1000 mb”, “975 mb”, “950 mb”, “925 mb”, “900 mb”, “850 mb” etc).

In this case, the image sequence solution is not useful. Instead, 30 image items are used (based on a selection of those altitudes, noting that they do not have a common reference level) along with an extra dimension for altitude. `edim` is used with `extradimension_type` 1. The `extradimension_count` is 1 (i.e., `i` is always 0), and `extradimension_index_count` is 30 with the 30 altitudes defined.

It is also possible to have an image item with 30 components but in this case, the altitude is not a dimension but a cell parameter.

2.11.3. Subcase 3. Temperature – multiple time predictions, multiple altitudes from a single forecast run.

This is a combination of subcase 1 and subcase 2. This forecast run involves 24 hours of predictions at 30 altitudes. The results could be structured into 30 image sequences, or up to 720 image items (could be less if multi-component tracks or images are used).

2.11.4. Subcase 4. Wind – multiple time predictions, multiple altitudes from a single forecast run.

This is like subcase 3, but the wind measurement has two parts (usually called “U” and “V”), which are orthogonal velocity components (instead, speed and direction could be used). The velocity values are resolved in the “east” and “north” directions. These could be treated as the real and imaginary parts of a complex number or as two **cell parameters** described in `pcel`.

2.11.5. Subcase 5. Multiple weather parameters, multiple time predictions, multiple altitudes from a single forecast run.

This forecast use case is the superposition of the subcase 3 and subcase 4. In this case, several measurements are performed, such as wind, temperature, relative humidity, and the wind vertical velocity. All of these measurements are **cell parameters** described in `pcel`. They can

exist in separate image items or in a multicomponent image item. Each image uses an association to the relevant `pcel`.

2.11.6. Subcase 6. Reduced number of parameters, multiple forecast runs.

In the multiple forecast run use case, more than one weather model (e.g., GFS, ECMWF, UK Met Office) or ensemble forecasts using slightly different initial conditions for the same model are considered. In both cases, the forecast model provides the “same” prediction for temperature and wind at the surface run at the same forecast time that provide predictions for a couple of time steps (say 1200Z and 1800Z). The key aspect is identifying which set of image items goes with which forecast using the relevant associations to the relevant cell properties (`pcel`) or extra-dimensions (`edim`).

2.12. Atmosphere characterization (height)

This is a typical case of a **3D datacube**. We characterize the atmosphere by measuring parameters such as water vapor content, temperature... at several heights. The HEIF file has **several image items**, at least one for each of the N height values in the height dimension. Each image item is a **multiband** image, each component containing an atmospheric parameter (water vapor content, temperature...).

If the 3D datacube need to be tiled for performance reasons, there is a `tiled` image (defined in “OGC Testbed 20: GIMI Lessons Learned and Best Practices Report, Annex B”) with one extra dimension for the height. Each tile is an ISO/IEC 23001-17 image with one component for each atmospheric parameter. Alternatively, one can use one separate `tiled` image for each atmospheric parameter and use images with just one component.

The `ogeo` brand must be included in the `FileTypeBox (ftyp)`. All images are georectified by associating them to the **same CRS** in the `mcrs` item property and to the **same affine transformation matrix** in the `mtx` item property. The third dimension, height, is defined as an **extra-dimension** in a `ExtraDimensionProperty (edim)` and the **values** that this extra-dimension takes are defined in `ExtraDimensionValueProperty (edvl)`. All images are associated to the **same cell property type property (pcel)**, that contains the description of the array of the atmosphere parameters measured.

2.13. Atmosphere evolution (height + time)

Atmospheric characterization is a typical **4D datacube** use case. The atmosphere is characterized by measuring parameters such as water vapor content and temperature at several heights and several times during a month.

The HEIF file has **several image items**, at least one for each of the N height values multiply by the M time values, resulting in at least N*M images. Each image item is a **multiband** image, each component containing an atmospheric parameter (water vapor content, temperature...).

If the 4D datacube need to be tiled for performance reasons, the HEIF file has a tiled `tiled` image (defined in “OGC Testbed 20: GIMI Lessons Learned and Best Practices Report, Annex B”) with two extra dimensions. One extra dimension is the height and the other extra dimension the time. As above, the representation can also be turned inside-out by storing multiple `tiled` 4D datacubes, one for each parameter.

The `ogeo` brand must be included in the `FileTypeBox` (`ftyp`). All images are georectified by associating the images to the **same CRS** in the `mcrs` item property and to the **same affine transformation matrix** in the `mtx` item property. In this case, the file has **two extra-dimensions** defined in `edim`: height and time and **two sets of values**, one for each extra-dimension defined in `edv`. All images are associated with the **same cell property type property** (`pcell`), that contains the description of the array of the atmosphere parameters measured.

2.14. Weather forecast runs (modeling ensemble)

The ensemble case requires a **complex datacube** with several extra-dimensions. In an ensemble weather forecast, several **models** are used to issue the weather prediction. Then there is the **time** when the calculations of the model are done (this created a time series) and the **anticipation** when the forecast predictions apply (another time series). This generated a 5D datacube with three additional dimensions: model, time and anticipation. The ‘model’ dimension is a categorical dimension (in this case the definition of the dimension values is included in `edv` item property), the ‘time’ dimension is continuous, and the ‘anticipation’ dimension is an ordinal dimension.

The HEIF file would have **several image items**, at least one for each combination of model, time and anticipation, resulting at least N*M*O images (where N is the number of models, M the number of times and O is the number of anticipations). Each image item is a **multiband** image, each component containing a predicted parameter (temperature, precipitation, wind direction, wind speed, cloud cover, etc.).

If the datacube need to be tiled for performance reasons, the HEIF file has a tiled `tiled` image (defined in “OGC Testbed 20: GIMI Lessons Learned and Best Practices Report, Annex B”) with several extra dimensions: one for height, time and anticipation.

In this case it is important to define the workflow by including the **metadata** and **provenance** information as a metadata item. To relate each image with its corresponding metadata an Item Reference Box (`iref`) is needed.

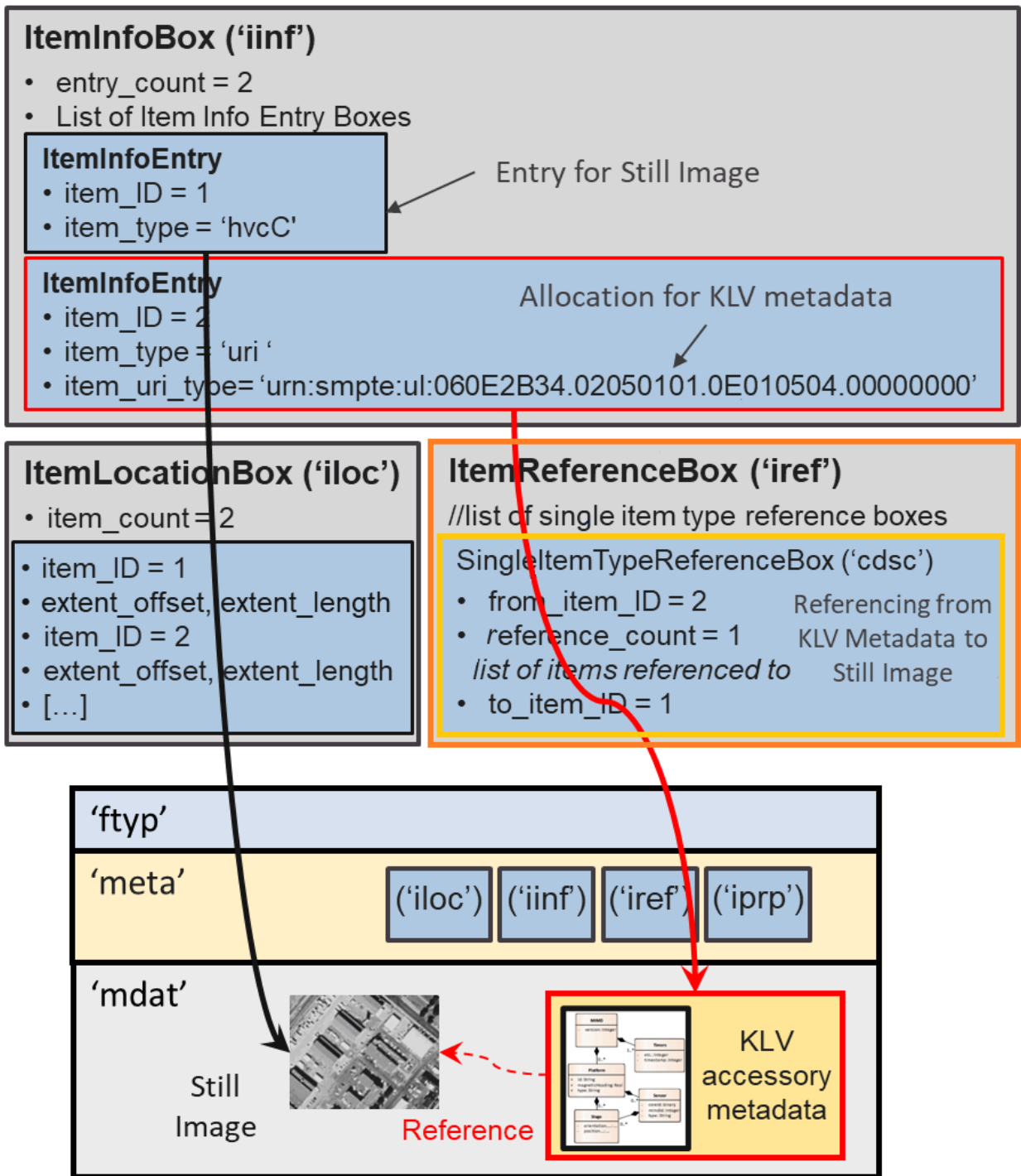


Figure 1 – Still image and its metadata related by ItemReferenceBox. Picture taken from NGA 0076 v0.6 and simplified to show only the accessory metadata

2.15. Crop yield prediction as a function of level of fertilizers and species and plant variety (modeling outputs depending on several input parameter combination)

The results of a model predicting crop yield depend on the initial parameters such as fertilizer quantity, irrigation intensity, and type of crop. Depending on the sophistication of the model, the execution time could be too long for an easy browsing application that explores the effects of adjusting the parameter values. A possible solution to this problem is to execute the model for all possible combinations of parameters. This way, **model parameters** become **extra-dimensions** in a datacube where cell property values are crop yields inside the cell. The practical implementation of this case is done in the same way as described in the weather forecast run. Again, continuous extra dimensions (fertilizer quantity, irrigation intensity, etc.) and categorical dimensions (type of crop, type of fertilizer, etc.) can be used.

If the datacube need to be tiled for performance reasons, the HEIF file has a tiled `tile` image (defined in “OGC Testbed 20: GIMI Lessons Learned and Best Practices Report, Annex B”) with the initial parameters as extra dimensions. If there is no simulation run for a specific combination of parameters, the output image (tile) can be marked as ‘undefined’.

2.16. Browsing (pan, zoom, clip...) through high resolution and large extent image in the Web or in the Cloud

A classical HTTP interaction requesting a resource results in downloading the whole resource. Traditionally, obtaining a part of a results such as a part of large image was implemented by offering imagery through a web service on top of HTTP server. The web service extracts a requested subset of original information hiding the complexities of the data structure. An HTTP Range request offers a different alternative, sending parts of a resource back to a client. In practice, it is possible to read parts of a resource in a different order than the original storage, allowing for a random access. As such, a client can random access the file in the same way as can be done on local drives. In this approach, the details of an optimized internal data structure of the file formats become visible and known to the client.

Cloud-optimized formats are a set of data storage and access approaches that enable more efficient handling of large geospatial datasets for visualization and analysis in cloud environments. The cloud optimized format may or may not be random accessible. However, the typical implementation may use HTTP range request. The key **benefits** of these formats include:

- **Efficient Data Access:** Enable clients to retrieve only the specific portions of a dataset that are needed, rather than downloading the entire file. This enables real-time workflows.

- **Reduced Data Duplication:** By accessing data directly from cloud storage, cloud-optimized formats eliminate the need to copy and cache entire datasets locally.
- **Legacy Compatibility:** Many cloud optimized formats are based on existing geospatial data formats. This enables traditional GIS software to work with the data without modification.

A cloud-optimized format provides a data structure for imagery that enables clients to make **HTTP range** requests (HTTP requests to ranges of bytes) to the server to fetch desired subregions and sub-resolutions on the fly.

There are two common data structure approaches: chunks and tiles. **Chunks** divide the file into a linear sequence of fragments of the same size without taking the geometric characteristic into account. **Tiles** divide the file into regular geometric shapes consisting of a single connected “piece” (topological disc) without “holes” or “lines” (commonly 2D) (definition adapted from OGC Abstract Spec on Tiling of 2D Euclidean space, OGC 19-014r3). Tiles are usually complemented by overviews (a lower resolution version of the same data, also divided into tiles of the same size).

The screen visualization use case can be considered a particular one, as the amount of data required to fill the screen is lower than a typical size of a GIS or remote sensing image. In this case, there are **two extreme cases**:

- Requesting the **whole area** of the image (a general view) will select a sub-resolution that covers the area of the screen (some times referred as the viewport) with a pixel size (defined as the size of the pixel in the reality) smaller but closer to the pixel size of the screen,
- Requesting a **deep zoom** will select the original resolution but only for a small area that covers the area of the screen, sometimes exceeding only for a small proportion.

In both extremes, the performance is conditioned by the ability to select the right ranges of bytes from the original image. Since the tiled image pyramid is pre-generated, the amount of data transmitted will depend on the size of the **tiles** used in the original tiling, and how closes the sub-resolutions in the original file are from the actual resolution required by the visualization. The amount of data transmitted will also depend on the efficiency of the compression of the original tiles.

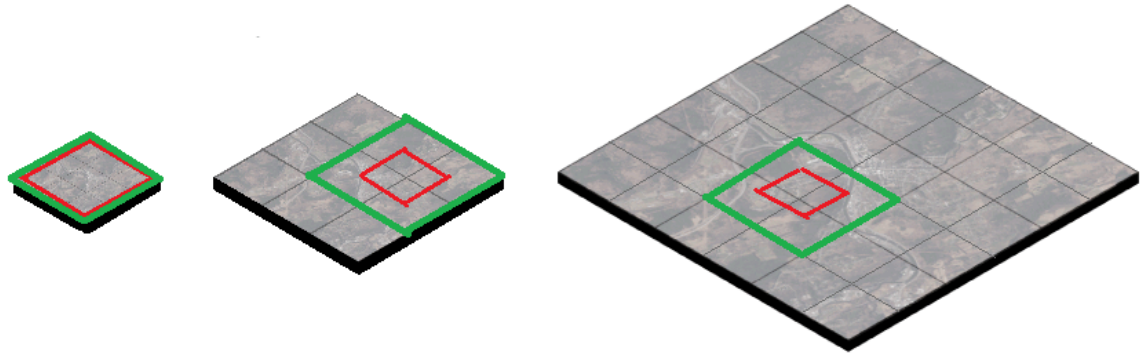


Figure 2 – Extracting the needed amount in 3 resolutions. (The red square represents the actual needed data for the screen, and the green square represent the minimum data to be transferred from the server)

A user viewing the image may do frequent pan and zoom operations. When a strategy for caching previous responses is implemented, the performance will also be conditioned by how much of the already transmitted data can be recycled, thereby reducing the amount of new information transmitted. This is conditioned by the original structure of the tiling and sub-resolutions.

The HEIF format provides a tiling mechanism using the `grid` image format (ISO 14496-12:2022 Section 6.6.2.3), that saves a tiled image as a collection of small images. Each tile is stored in the HEIF file as a separate image item, which are then combined into a large grid image. This format has some limitations: The number of tiles in each dimension (i, j) cannot exceed 256. In addition, every source image used as part of the grid derived image must be referenced to the grid image. Further, the number of references is limited to 65535, so at most one dimension can be 256. Assuming typical tile sizing (e.g., 256 pixels wide and 256 pixels high), this limitation is likely to be encountered in the geospatial domain. For example, when representing a long strip from a line-scan sensor, or a mosaic of a composite image. Note that this limitation does not occur in TIFF, or in the derived Cloud Optimized GeoTIFF (COG) format.

In “OGC Testbed 20: GIMI Lessons Learned and Best Practices Report, Annex B” a solution to support larger images that extends the HEIF format and provides a `tiled` image type was proposed as an alternative to `grid`.

Also, the ISO/IEC 23001-17 lossless image format has built-in support for efficient tiling and can be used when lossless data or non-integer datatypes are required.

Also, as part of Testbed 20, Silvereye Technology evaluated options for Coverage selection from HEIF files related to this use case described in Annex B.

Testing if the tile approach is a good strategy is not easy as it involves several HTTP range requests and a good implementation of a caching or buffer mechanism. Some details on how to conduct benchmarking on tiled images are explain in Annex C.1 and Annex D.

2.17. Confidential or private image (metadata)

During Testbed 20 no item property was proposed to implement the security information for an image.

The GIMI profile, NGA 0076 v0.6, specifies a way to label the security content based on the use of an Information Security Markings (ISM.XML) document and UUID's, ISM.XML provides a way to label each item.

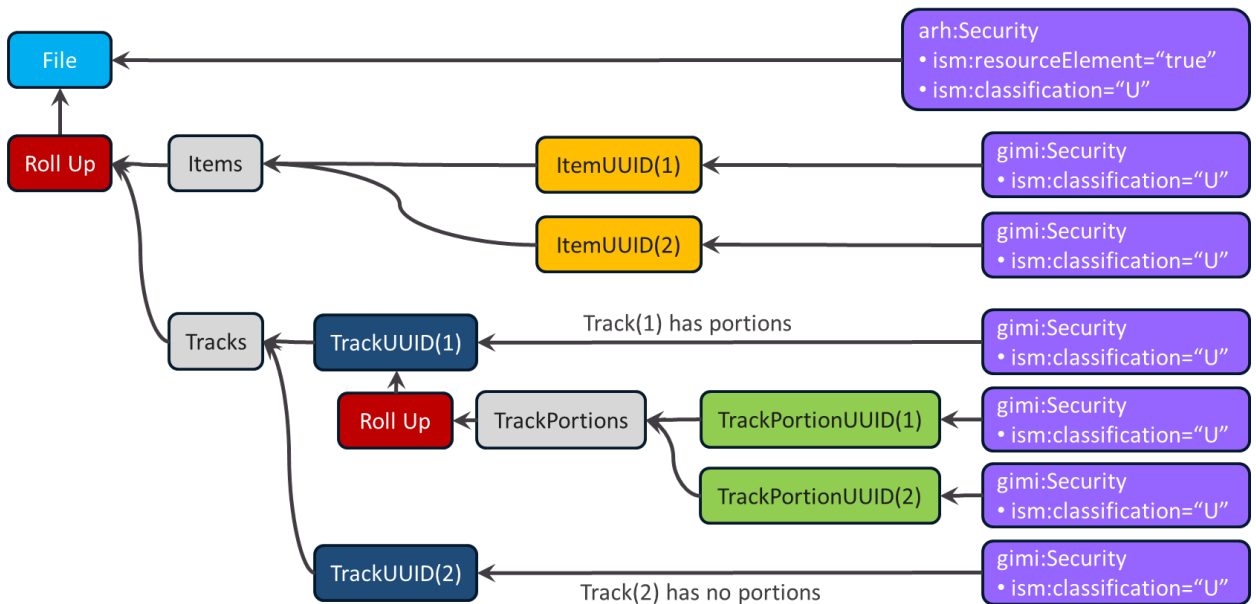


Figure 3 – Media Configuration and ISM labeling for the XML code example. Picture taken from NGA 0076 v0.6

```
<?xml-model href="../../../ISM/Schematron/ISM/ISM_XML.sch" type="application/xml"
schematypens="http://purl.oclc.org/dsdl/schematron"?>
<GIMISecurity xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:us:mil:nga:stnd:0076:ism"
xmlns:ism="urn:us:gov:ic:ism"
xmlns:gimi="urn:us:mil:nga:stnd:0076:ism"
xmlns:arh="urn:us:gov:ic:arh"
xsi:schemaLocation="urn:us:mil:nga:stnd:0076:ism Sample_imagery_
ism.xsd"
ism:DESVersion="202405"
ism:ISMCATCESVersion="202405"
gimi:GIMISecVer="1">
<File>
<arh:Security ism:compliesWith="USGov USIC"
ism:resourceElement="true"
ism:createDate="2006-05-04"
ism:classification="U"
ism:ownerProducer="USA"/>
<Items>
<Item gimi:itemID="ee4ea7a9-27a1-522f-9e6d-182618b2609e">
<gimi:Security ism:classification="U"
```

```

        ism:ownerProducer="USA"/>
    </Item>
    <Item gimi:itemID="e03d1743-5664-586e-916f-9fa731d0315f">
        <gimi:Security ism:classification="U"
            ism:ownerProducer="USA"/>
    </Item>
</Items>
<Tracks>
    <Track gimi:trackID="60f00d81-52c3-5c3c-9258-5f0e07a63076">
        <gimi:Security ism:classification="U"
            ism:ownerProducer="USA"/>
        <TrackPortions>
            <TrackPortion gimi:trackportionID="d83b7576-4fc0-59e3-9303-
9d1a9830c0ed">
                <gimi:Security ism:classification="U"
                    ism:ownerProducer="USA"/>
            </TrackPortion>
            <TrackPortion gimi:trackportionID="2245b213-9d44-5c8f-8124-
645df7a65110">
                <gimi:Security ism:classification="U"
                    ism:ownerProducer="USA"/>
            </TrackPortion>
        </TrackPortions>
    </Track>
    <Track gimi:trackID="8b40ebf8-c278-5d7a-88c6-95aa89b61deb">
        <gimi:Security ism:classification="U"
            ism:ownerProducer="USA"/>
    </Track>
</Tracks>
</File>
</GIMISecurity>

```

Listing 1 – Example of ISM.XML Instance File taken from NGA 0076 v0.6

2.18. Sparse images

Images that cover a very large spatial area may only contain small areas of interest. For example, imagery of oceans may show mostly water without objects such as small islands, ships, or sea mammals. Providing high level images of the large area with detailed imagery only where an object has been detected may be useful.

The sparse images case can be addressed using the `tiled` image item described in the “OGC Testbed 20: GIMI Lessons Learned and Best Practices Report, Annex B”. This supports the implementation of tile space with empty tiles. In this case an empty tile is a tile that is enumerated in the `TiledImageOffsetTable` with a `tile_start_offset` 0 indicating the tile content is not encoded. All tile images should be compressed to optimize the space.

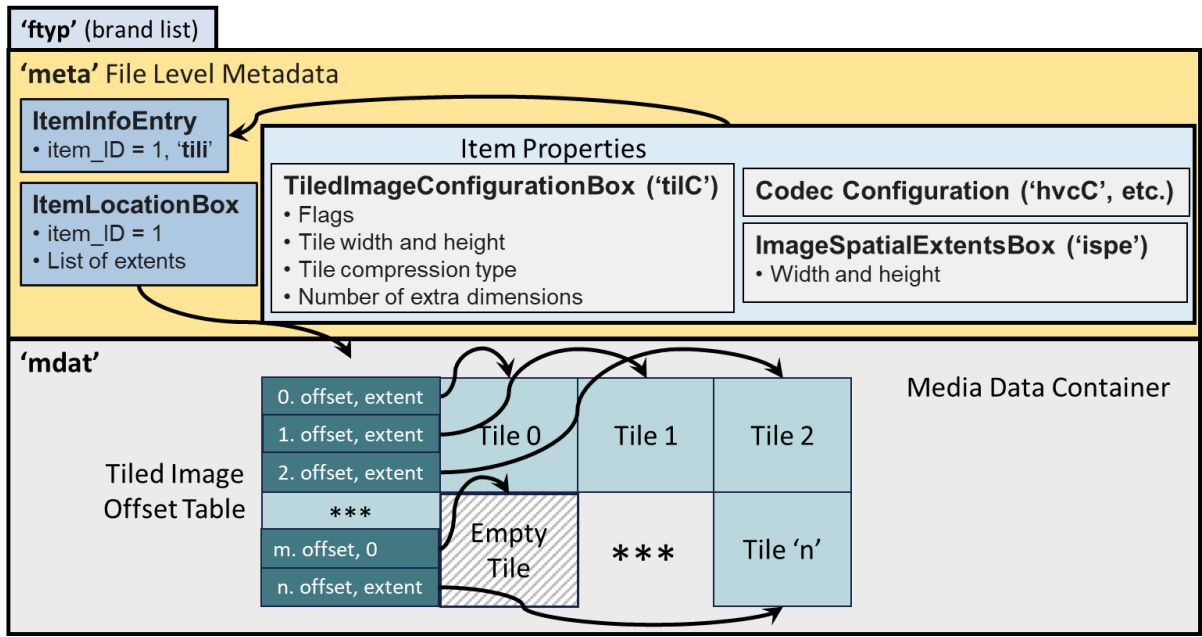


Figure 4 – File structure of a tili image item. Picture taken from OGC 24-040

2.19. Planetary image (extraterrestrial CRS)

For images referenced to a planetary reference system, the `mcrs` item property provides ways to reference or describe the CRS. In the GeoHEIF format, the CRS is not only a number (such as in the case of GeoTIFF) but a UTF8 text preceded by a numeric `crsEncoding`. It is likely that relevant organizations create registries for reference systems on other planets and celestial objects. When there is an URL that links to a registry that provides a definition of the planetary CRS, a 4CC `crsu` code is used. When there is not CRS URI, but the CRS can be described as a combination of datum, projection and ellipsoid, a `wkt2` code is used. Finally, when the image is a picture taken by a sensor, an Engineering CRS may be necessary. In this case, the Engineering CRS is centered on the spacecraft's camera and the coordinate system class is restricted to Cartesian (3D), Spherical (3D), and Spherical (2D). See more details in in OGC Testbed 19 Extraterrestrial GeoTIFF Engineering Report [OGC 23-028]. Currently the definition of an Engineering CRS is not specified in the GeoHEIF format.

2.20. Satellite images time series

When a series of images are taken from an earth observation platform as it tracks over the planet, the images potentially form an overlapping mosaic where each image has a different imaging time.

The serie of images can be stored preferably in an image sequence. The respective absolute capture time for each image can be stored as TAI timestamp, which is assigned to each image sample as a Sample Auxiliary Information (SAI). TAI timestamps are defined in ISO/IEC 23001-17 Amendment 1. In the GeoHEIF format, each image can also be georeferenceable or georectified by associating it with a CRS in a `mcrs` property that is assigned to the track's sample description box. This CRS is constant for the whole track or at least a sample cluster (time range of the track). The camera position matrix can be stored in a property similar to an **affine transformation matrix** (`mtxf`) data packet. The property needs to be defined in futures activities, and that is assigned to each image as sample auxiliary information.

Alternatively, this case can be considered as a **3D datacube** where the extra-dimension is the **image time**. In the GeoHEIF format, each image can be georeferenceable or georectified by associating it to a CRS in the `mcrs` item property and to a set of **tie-points** in the `tiep` item property or to a **affine transformation matrix** in the `mtxf` item property respectively, both in the `ItemPropertyContainerBox` and related to the image item in the `ItemPropertyAssociationBox`. In this case each image has a different `tiep` or `mtxf` item property, and typically all images have the same `mcrs`. The **image time** extra-dimension is implemented as 'edim' item property and the set of times that this dimension can take is implemented as `edvl`.

2.21. Burst Photography

In HEIF, a burst of images can be more efficiently coded in an image sequences. HEIF provides additional constraints that enable fast random access to these images. For example, predictive coding can be restricted to have just one reference frame. Thus, each frame in the burst can be extracted by decoding at most two images (the reference image and the desired image that uses the reference image for prediction, resulting in a composed image). It is also possible to store the image burst together with a composed image and relate the image burst and the composed image together. For example, a single file could contain a sequence of exposure-bracketed input images and the resulting HDR image.

2.22. Videos from a camera with a fixed position

The video with a fixed position use case could be a surveillance camera, a fixed camera used for traffic monitoring by taking images or video streams, or satellite images or orthophoto time series.

In a GeoHEIF file, these images can be encoded as image sequences with the camera CRS `mcrs` and the transformation matrix `mtxf` stored in the sample description box of the track. Note that, different from the "Satellite images time series" use case, the `mtxf` is stored in the sample description. This is because the camera is fixed and therefore not necessary to transmit a new transformation matrix for each frame.

In a HEIF file, these images can be encoded as image sequences using the MovieBox (moov) with a TrackBox (trak). In the TrackBox, the MediaBox (mdia) includes a MediaInformationBox (media) with timescale and the duration of the sequence and other boxes such as the SampleTableBox (stbl) and the SampleDescriptionBox (stsd) where there is a description of what type of samples are in the images, which codec is used. There are also other boxes, such as DecodingTimeToSampleBox (stts) which provides the duration of each frame, or the SampleSizeBox (stsz) which provides the size of each frame.

2.23. Videos from a moving camera

This videos from a moving camera use case could be images or video tracks from a camera installed on a helicopter for traffic monitoring, or images or video tracks from a drone.

Similar to the Satellite images timeseries, the camera position and pose can be stored with a fixed CRS `mcrs` in the sample description box and the transformation matrix `mtx` as sample auxiliary information attached to each frame. If the high-precision TAI timestamps are not needed, they can be omitted and the sample timing of the ISOBMFF format can be used, which is suitable for fixed frame-rate video cameras.

In a HEIF file, if additional metadata is required, this can be added as a separate metadata track. The content of the data is application specific and can include various sensor data (like acceleration sensors or temperature). This results in two track boxes (trak): a track for the video and a track for the metadata. The metadata track includes a reference to a video track using a `cdsc` (Content Description) reference to denote that the metadata track is further describing the referenced visual track. This allows metadata to be synchronized with the video (or visual) track. Both tracks can include a DecodingTimeToSampleBox (stts) that provides the duration of each frame. The duration can be different for the metadata frames and for the visual frames.

2.24. Videos from multiple cameras

The videos from multiple cameras use case covers videos when the same event is being tracked from multiple cameras or when it is desired to create a 3D movie with stereographic cameras.

A HEIF file can contain multiple video tracks, one for each camera's samples. Even though the video is stored in separate tracks, they can use the same time base, such that the camera images stay synchronized. This can be achieved even if the cameras run at different framerates.

The camera positions can be stored in a way similar to the CRS and the **affine transformation matrix** is stored for the still images, using a `mcrs` and `mtx` as described in previous use cases (and defined in GeoHEIF), or using the `cmi`, `cmex` boxes (camera internal matrix, camera external matrix) defined in the HEIF standard. The latter is specifically designed for stereoscopic cameras, but can also be used in other use cases when only the relative pose of the cameras in relation to each other is important.

2.25. Videos with a depth sensor

A depth image, also known as a depth map or range image, is a visual representation where each pixel encodes the distance from a sensor or camera to points in a scene. Unlike regular color images, the value at each pixel in a depth image corresponds to the relative or absolute depth (distance) of the scene's surfaces from a specific viewpoint. The depth images usually are accompanied with "depth representation information" (stored, for example, as H.265 SEI messages). This information defines the near and far plane of the depth data and whether the distances are sampled linearly or stored as the disparity between left and right image.

A depth image can be obtained directly by special cameras with a time-of-flight sensor or by constructing it with an algorithm that use color images from stereo camera pair. These depth images can be stored in a separate video track. The image resolution of this depth image can (and often will) be different from the main color image track. The depth track should use a track reference to the color track to indicate that it contains auxiliary depth data instead of visual content.

As depth images have different characteristics than color images, typical compression artifacts that are acceptable for color images can be undesired for depth images. This should be considered when choosing the right video codec for the depth data. For example, there is a special coding mode for depth images in H.265, but it is not widely supported. Another choice is to use a lossless codec or a codec where the maximum error can be controlled (e.g., JPEG2000).

2.26. Videos with reconstructed 3D point cloud

Some cameras reconstruct a sparse 3D point cloud that accompanies the 2D color image. Since the format of this 3D point cloud varies, storing this as custom sample auxiliary information, attached to each image is recommended. There is no standardized four-character code to identify these yet and a proprietary code has to be used.

3

OUTLOOK

In the future, an additional comparison of the GeoHEIF selected use cases with other file formats can be conducted. This work could include optimal selection of specific tools and features from the ISO Base Media File Format and Image File Format family of standards for application to geospatial use cases. Further benchmarking of performance and usability could form part of this comparison.

The use case exercise documented in this OGC report was limited to still imagery and time series. Sequence images and motion imagery were introduced but more experimentation and definition of more properties to detail camera position is needed.

Both access to the local files as well as to HEIF files accessed through HTTP range-requests over the web (file formats optimized for web access are commonly known as cloud optimized formats) have been considered and could be implemented and compared in a next OGC Testbed. The name COHEIF (Cloud Optimized GeoHEIF) has been suggested.

While limited labeling for security requirements was investigated in the implementation of GIMI, future activities could investigate broader security considerations, including implementing Common Encryption as specified in ISO/IEC 23001-7.



4

SECURITY, PRIVACY AND ETHICAL CONSIDERATIONS

4

SECURITY, PRIVACY AND ETHICAL CONSIDERATIONS

This document introduces some considerations on how to mark the confidentiality level of the information. This Report does not discuss encryption of the content of the HEIF file or mechanisms to ensure integrity. These could be the focus of future work.



BIBLIOGRAPHY





BIBLIOGRAPHY

- [1] Carl Reed: OGC 19-014r3, *Topic 22 – Core Tiling Conceptual and Logical Models for 2D Euclidean Space*. Open Geospatial Consortium (2020). <http://www.opengis.net/doc/AS/2D-tiles/1.0>.
- [2] Joan Maso: OGC 21-026, *OGC Cloud Optimized GeoTIFF Standard*. Open Geospatial Consortium (2023). <http://www.opengis.net/doc/is/COG/1.0.0>.
- [3] Michael Leedah: OGC 23-028, *OGC Testbed 19 Extraterrestrial GeoTIFF Engineering Report*. Open Geospatial Consortium (2024). <http://www.opengis.net/doc/PER/T19-D002>.
- [4] ISO/IEC: ISO/IEC 14496-12:2022, *Information technology – Coding of audio-visual objects – Part 12: ISO base media file format*. International Organization for Standardization, International Electrotechnical Commission, Geneva (2022). <https://www.iso.org/standard/83102.html>.
- [5] ISO/IEC: ISO/IEC 23008-12:2022, *Information technology – High efficiency coding and media delivery in heterogeneous environments – Part 12: Image File Format*. International Organization for Standardization, International Electrotechnical Commission, Geneva (2022). <https://www.iso.org/standard/83650.html>.
- [6] OGC Testbed 20 GEOINT Imagery Media for ISR (GIMI) Specification Report
- [7] OGC Testbed 20 GIMI Lessons Learned and Best Practices Report
- [8] GEOINT Imagery Media for ISR (GIMI) Profile of ISOBMFF



ANNEX A (NORMATIVE) ABBREVIATIONS/ACRONYMS



ANNEX A (NORMATIVE) ABBREVIATIONS/ACRONYMS

COG	Cloud Optimized GeoTIFF
CRS	Coordinate Reference System
GeoHEIF	Geographic High Efficiency Image File Format
GFS	Global Forecast System
GIS	Geographic Information System
HD	High Definition
HEIF	High Efficiency Image File Format
ISOBMFF	ISO Base Media File Format
MPEG	Moving Picture Experts Group
NWS	National Weather Service
TAI	International Atomic Time
TIFF	Tagged Image File Format



B

ANNEX B (INFORMATIVE) SILVEREYE TECHNOLOGY – COVERAGE FORMAT

B

ANNEX B (INFORMATIVE) SILVEREYE TECHNOLOGY – COVERAGE FORMAT

B.1. Executive Summary

There are a range of coverage selection methods that can be supported by HEIF (and hence by GIMI). These include:

- download of the entire file
- client side byte range requests (e.g., with HTTP(S)).
- server side APIs such as OGC API – Tiles or OGC API – Coverage

For a server that understands the underlying HEIF file structure, there are strategies that enables the server to return HEIF tiles without needing to first decode any part of the compressed data first. This can improve server performance and avoid re-encoding artifacts when using lossy compression types.

B.2. Tiling

As part of Testbed 20, Silvereye Technology evaluated options for Coverage selection from HEIF files. In addition to the options for the client to download the whole file, or parts of the file using HTTP byte range requests (as for the Cloud Optimized GeoTIFF concept), HEIF and hence GIMI can also be used with a server side API implementation of OGC Standards such as OGC API – Tiles or OGC API – Coverages – Part 1: Core.

When used with a server side API endpoint, a HEIF file that incorporates tiling or overviews (a.k.a. sub-resolutions), or both tiling and overviews, can be more efficient than one that does not. The use of tiling and overviews in a HEIF file enables the client to select the information required to respond to the query without decoding the whole file.

As a special case, the server may advertise a tiling scheme that matches the underlying tile structure of the HEIF file. In this case, the response to the query can be a HEIF file that is

constructed without needing to decode the data on the server side. Instead, a server that understands the HEIF structure can identify the component image item that corresponds to the query and then extract, along with the associated properties, the required compressed data for that image item, along with the associated properties. These can then be rebuilt into a conformant HEIF file without decompressing the data first.

Where a coverage request is aligned to the coordinate reference system, but the tiles do not have the exact spatial extent required, the server can potentially return a superset of the data required, and associate a Clean Aperture (c1ap) transformative item property that trims the resulting image to the requested spatial extent. The Clean Aperture property instructs a HEIF reader that the crop operation must be performed before rendering the final image. In this case, the tile(s) can again be generated without decoding. Note that this can not provide image warping or reprojection.



ANNEX C (INFORMATIVE) BENCHMARKING – APPROACHES AND CONSIDERATIONS



ANNEX C (INFORMATIVE) BENCHMARKING – APPROACHES AND CONSIDERATIONS

C.1. Benchmarking: Strategy for evaluating tiled images

In order to execute a representative benchmark, a strategy based on simulating the actions of a user browsing and panning was suggested. From the required content, the minimum data needed can be calculated to be transferred from the server and the corresponding HTTP range requests can be formulated using CURL with the appropriate headers.

```
C:\temp>curl -v -X GET -H "range: bytes=1-8" http://joanma.uab.cat/temp/2023_
NDWI_Cataluna.tif -o kk.bin

> GET /temp/2023_NDWI_Cataluna.tif HTTP/1.1
> Host: joanma.uab.cat
> User-Agent: curl/8.4.0
> Accept: */*
> range: bytes=1-8
>
< HTTP/1.1 206 Partial Content
< Content-Type: image/tiff
< Last-Modified: Tue, 16 Jul 2024 11:23:21 GMT
< Accept-Ranges: bytes
< Content-Length: 8
< Content-Range: bytes 1-8/1857384822
<
{ [8 bytes data]
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
             %                   %         Dload  Upload  Total  Spent    Left   Speed
100      8   100      8     0     0    283      0  --:--:--  --:--:--  --:--:--    296
```

Listing C.1

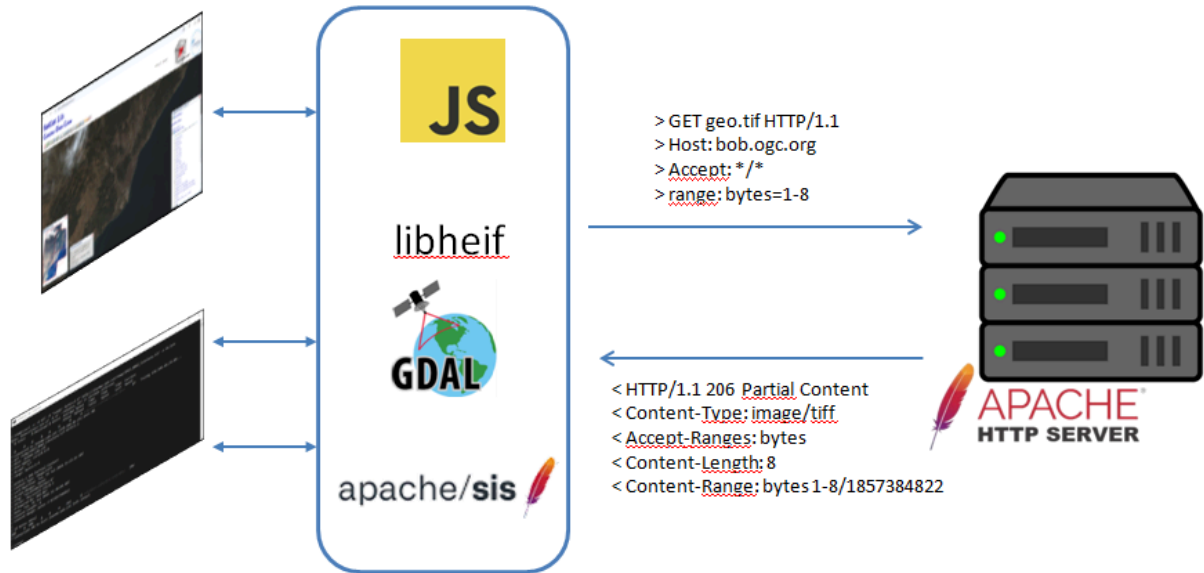


Figure C.1 – Setup to benchmark imagery

This task assumed:

- speed,
- compression ratio of received data
- % of extra data transmitted,
- % of recycled previous cached data in sequential requests (note that headers are retrieved only once)

As an example, the current implementation of libheif first fetches a small portion at the file start to read the size of the Metabox and then requests the whole Metabox data. Then whenever the application asks libheif to decode a tile, libheif will call back into the application, requesting the required file range.

Performance can depend on the way we arrange the requests but also on the way the image was prepared in the first place (tile size etc., different multiband interleave approaches e.g., by band, by tile, by row, by pixel or tile arrangements (by line or “zigzag”), compression algorithm).

C.2. List of benchmarking procedures implemented

- Benchmark of a serial access: Sequential use case, in which the conversion does not need to be accurate between user-requested coordinates and pixels and HTTP range requests.

- Benchmark of a random access: Extracting a region of interest (ROI) from a big image. ROI may not align with tile boundaries.
- Benchmark of a random access with overviews: Access to scale-in / scale-out version of same image.
- Benchmark of full access: Downloading the entire image, and, uploading and ingesting the entire image.
- Implicit steps in any test: Extraction of metadata for an entire image. For example, for accessing the tile information, or for knowing if the image is compressed and uncompressed.



D

ANNEX D (INFORMATIVE) SECTIONS OF THE HEIF TO READ IN A PARTIAL READING TILED IMAGE

D

ANNEX D (INFORMATIVE) SECTIONS OF THE HEIF TO READ IN A PARTIAL READING TILED IMAGE

The HEIF format (and the ISO Base Media File Format from which HEIF derives) uses a set of nested “box” structures. The FileTypeBox is always the first box. After that, boxes may occur in any order.

A typical HEIF file will consist of a FileTypeBox (ftyp), a MetaBox (meta), and a MediaDataBox (mdat), and may include other top-level boxes. The encoded image data is almost always provided in the MediaDataBox, as potentially unordered sequences of bits.

To decode the image data, the reader software first checks the brands in the FileTypeBox to ensure that there is at least one brand that the reader software is compatible with. If the software is compatible, then it must locate the MetaBox.

The MetaBox identifies the primary image, the type of image encoding (e.g., H.264, AV1 or uncompressed) for each image, the location of the data (e.g., a set of extents in the MediaDataBox), and associated image properties and images (e.g., a separately encoded alpha plane).

When reading part of an image that is remote (e.g., on a cloud service accessed over HTTPS), having the initial fetch of data include both the FileTypeBox and MetaBox is useful. If this is not the case, one or more additional round trips are needed to find the MetaBox data. In addition, for a file using overview (pyramid) structures, having the initial fetch of data also include at least the top level (most zoomed out) tiles in the pyramid is useful.

While the optimal arrangement of a file is at least somewhat use-case specific, the order of boxes described in ISO 14496-12:2022 Section 6.3.4 provides good general guidance. In addition, any overview tiles should be ordered from lowest resolution (most zoomed out) to highest resolution (most zoomed in). It should be noted that this ordering optimizes for reading performance and may require additional effort by producers (writers). In particular, significant additional memory may be required to store the file prior to serialization, or use of temporary files and subsequent copying of data.

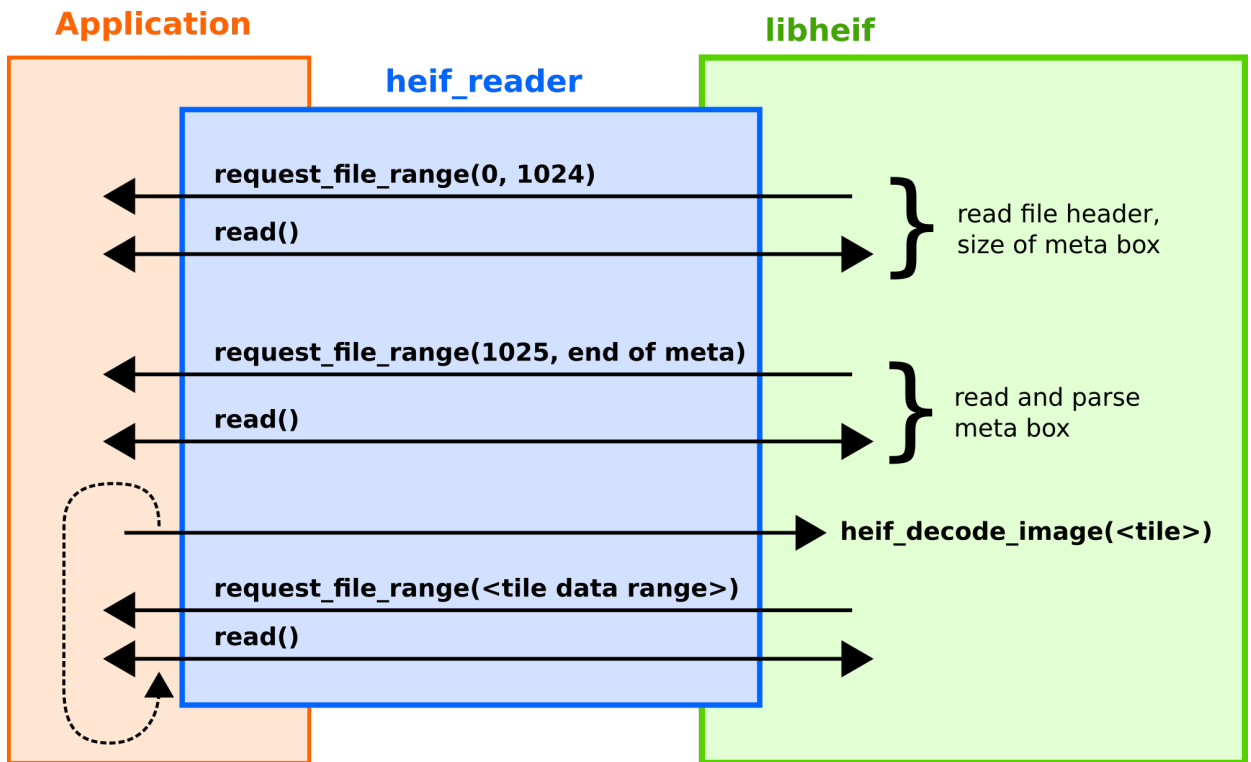


Figure D.1 – how to read a HEIF file using HTTP range

Normally in a geospatial application, geographic areas are transformed into tile indices and tile indices to pixel zones and that are related to HTTP byte ranges. More information can be found in: “OGC Testbed 20: GIMI Lessons Learned and Best Practices Report, Annex B”. The transformation of the geographic areas into tiles needs the georeference information specified in the ‘mcrs’ and in ‘mtxf’, see OGC 24-038.