ESA EOPF 101

Table of contents

Welcome

This online book is the go-to resource to learn everything about the EOPF Sample Service by ESA.

To be continued…

Part I

About EOPF

1 Introduction to the EOPF

1.0.1 What is the Earth Observation Processor Framework (EOPF)?

The Earth Observation Processor Framework (EOPF) is an initiative led by the European Space Agency (ESA) designed to modernise and harmonise data from the Copernicus Sentinel Missions.

With the upcoming Copernicus Expansion missions in 2028, the amount of data produced daily will significantly increase. EOPF is ESA's solution to organise Sentinel data in a way that works seamlessly with modern cloud technology. This will make it easier to find, access, and process the information you need. The new approach provides user-friendly access, simplifies maintenance, and helps keep costs down, guaranteeing reliable access to Sentinel data in the long run.

The Sentinel-1, Sentinel-2, and Sentinel-3 missions are the first to be updated with this new system.

1.0.2 The EOPF Data Model

The EOPF data model has been defined by following a set of principles:

- **Open standards:** Following common and community approved data standards ensure sustainability and user uptake.
- **Interoperability:** Harmonised with a clear and organised structure that describes the data itself.
- **Cloud optimisation:** Designed for efficient access and handling in cloud environments.
- **Conversion flexibility:** Providing tools to adjust the data for different applications.

Under EOPF, there are four key areas of activities: (i) EOPF product structure, (ii) EOPF metadata structure, (iii) EOPF encoding structure and (iv) EOPF Processor Framework

1.0.2.1 EOPF product structure

As part of the EOPF, ESA is actively working on a common data structure for Sentinel data products, with the aim to define a common meta-model that can be used across all Sentinel and other EO missions. This approach ensures that data from several missions is consistent.

The EOPF product structure consists of the following components:

- **Measurements:** The actual sensor readings (like how much light is reflected or the temperature), at different levels of detail.
- **Quality indicators:** Details that help understand how reliable the measurements are.
- **Conditions:** Information about the environment or technical aspects when the data was collected.
- **Attributes:** Global metadata, such as when it was acquired and the sensor's orbit.

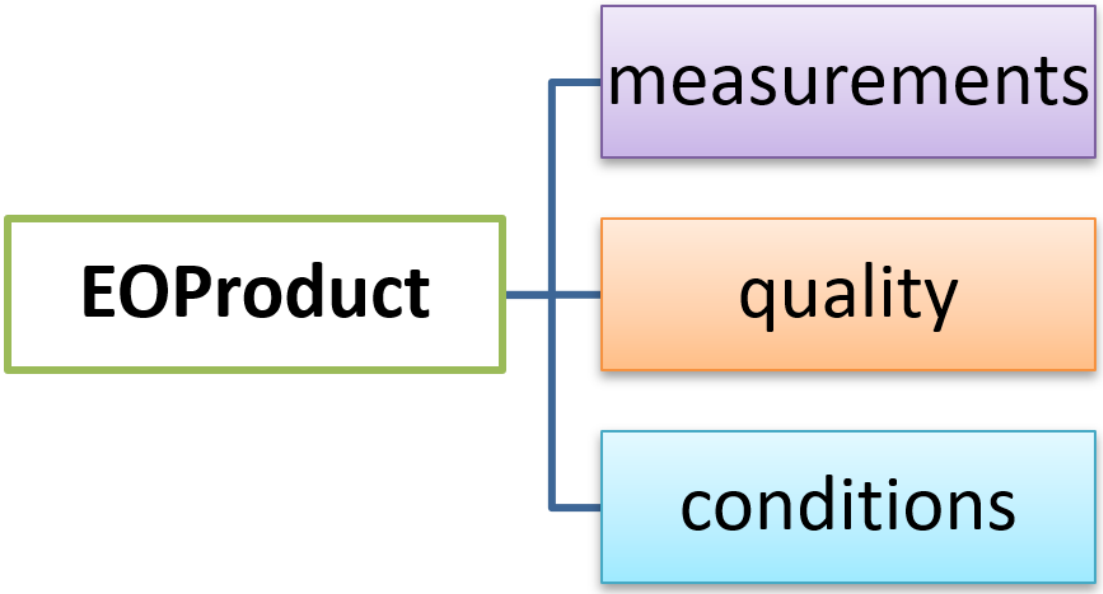

Figure 1.1: EOPF product structure

> **Note**
>
> Learn more about the EOPF Zarr product structure here.

1.0.2.2 EOPF metadata structure

Metadata provide all relevant information required to uniquely describing each Sentinel product. The EOPF metadata structure will be organised into:

- **Discovery Metadata**: Following the metadata structure defined by the SpatioTemporal Asset Catalogue (STAC), which helps to keep things consistent across different missions.
- **Processing History Metadata**: keeping a record of how the data has been processed.
- **Other Metadata**: Information like the status of the sensor and details about the satellite's orbit.

> **Note**
>
> EOPF and STAC: Learn more about EOPF and STAC here.

1.0.2.3 EOPF encoding structure

An encoding structure can be seen as the specific method used to package and store data and its associated metadata in a digital format. Building on the consistent data structure and clear metadata, the new storage system must be capable of handling various aspects of current Sentinel data (such as manifest files and tile structures from the SAFE format) while remaining fully compatible with cloud environments.

ESA chose `.zarr` as encoding format as it allows for instant access to data, efficient processing of massive amounts of data, and seamless integration with other datasets. The EOPF Zarr data format allows you to work with data from multiple missions more effectively.

> **Note**
>
> Learn more about the EOPF Zarr format here. And learn more about cloud-optimised geospatial data formats in general in the Cloud-Optimised Geospatial Data Formats Guide

1.0.2.4 EOPF processor framework

The way Sentinel data is processed is being updated to take advantage of modern cloud computing. This will make the processing faster and more efficient, while ensuring the scientific quality and accuracy of the Sentinel data remains the same.

> **Note**
>
> To learn more about the EOPF processor framework, visit https://eopf.copernicus.eu/eopf/

1.0.3 What's next?

In the following chapters, we will introduce the datasets being made available under EOPF and will provide you practical examples to work with the new EOPF Zarr data format.

2 About Cloud Optimised Formats

2.0.1 Why do we need to cloud-optimise geospatial data formats?

The volume of EO data has grown exponentially in recent years. The Copernicus programme alone generates ~16TB daily from the Sentinel missions. Traditional file formats, like SAFE (where each file can be hundreds of megabytes), are optimised for efficient archiving and distributing data. This means that we often download the data from an entire overpass, even if we only need to access a small part of it, for example, if we want to do an analysis of the area of a single city over a decade.

With growing data volumes, this becomes a challenge. To picture the different nature of challenges we come across, let us compare a traditional local workflow with a cloud-based workflow:

- **Traditional local workflow**: When working locally, we download much more data than we need, and we are constrained by the compute and storage capacity of the local system. However, an advantage working locally is that data and compute are close together, meaning that there is not much delay in accessing the data.
- **Cloud-based workflow**: Cloud environments overcome limitations local workflows have. A cloud environment offers limitless storage and compute capacity. On the contrary, data storage, compute, and you the destination are far apart. There is an additional time for data to travel between the storage location, processing resources and us. This time is referred to as **data latency**.

> **Note**
>
> **Data latency** refers to the time it takes for data to be transmitted or processed from cloud storage to your computer. In local workflows, data latency is minimal, whereas in cloud-based workflows, data latency needs to be optimised.

Local workflows are similar to placing an order at the nearby pizzeria. It is quick since the 'data' (pizza) is easily accessible, but we can only choose from what they have on hand and their menu. The local alternatives limit our options. On the other hand, cloud-based workflow offers almost limitless choices and access to a wide range of speciality ingredients or distinctive styles. This makes it similar to being able to order a pizza from any pizzeria on the globe.

While we might have more options to choose from, the time between order and delivery can become a challenge.

The overall goal with cloud-based workflows is to minimise **data latency** as much as possible. This is why traditional data formats need to be cloud-optimised.

2.0.2 Characteristics of cloud-optimised formats

Cloud-optimised formats are optimised to minimise data latency. By allowing for an efficient retrieval of smaller, specific chunks of information rather than downloading an entire file. Accessing a smaller data subset also reduces the costs associated with data transfer and data processing.

Cloud-optimised geospatial data formats have the following characteristics:

- Data is **accessible over an HTTP protocol**.
- **Read-Oriented**, as it supports partial and parallel reads.
- Data is **organised in internal groupings (such as chunks, tiles, shards)** for efficient subsetting, distributed processing and data access in memory.
- **Metadata** can be accessed in one read.

> **Note**
>
> When accessing data over the internet (e.g., cloud storage), latency is high compared to local storage, so it is preferable to fetch lots of data in fewer reads.

2.0.3 Understanding N-dimensional Arrays

Earth observation data often consists of multiple dimensions - think of a satellite image series that has spatial dimensions (latitude, longitude), spectral bands, and time. This multi-dimensional nature requires specialized data structures and tools.

2.0.3.1 The xarray Data Model

xarray provides a powerful data model for working with labeled multi-dimensional arrays. It introduces:

- **Dimensions**: Named axes (e.g., latitude, longitude, time)
- **Coordinates**: Labels for points along dimensions
- **Variables**: N-dimensional arrays with corresponding dimensions
- **Attributes**: Metadata about the arrays

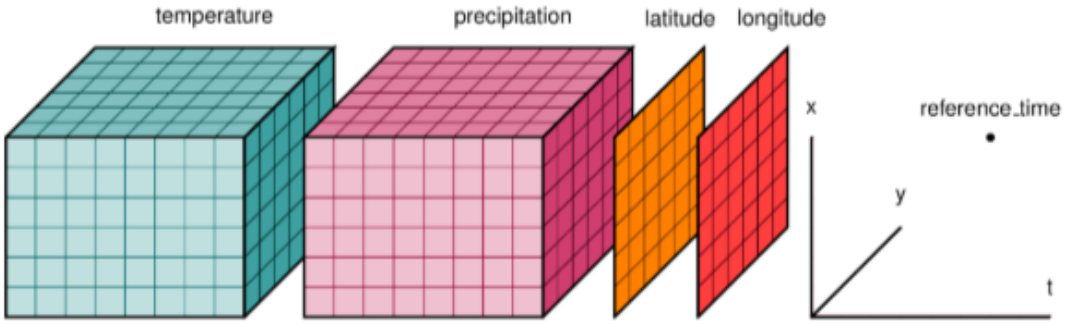

Figure 2.1: xarray's data model showing dimensions, coordinates, and variables

2.0.4 Cloud-Optimised Geospatial Data Formats

For satellite data, there are two main categories of cloud-optimised formats: 1. **Raster Formats**: Optimized for 2D image data (like Cloud-Optimised GeoTIFFs) 2. **Multi-dimensional Array Formats**: Designed for complex, n-dimensional data structures

2.0.4.1 Cloud-optimised GeoTIFF (COG)

COGs have widely been used as cloud-native format for satellite imagery and improve the standard GeoTIFF format by: - Organising data into **tiles**: Dividing the data into smaller, manageable squares (like 512x512 pixels). - Including lower-resolution previews: Having pre-generated, less detailed versions of the data. This allows for fast and efficient data visualisations.

A key feature of COGs is the **Internal File Directory** (IFD), which acts like an internal index. This allows for retrieving only the parts of the data needed using simple web requests. For example, it is possible to access just the tiles covering Paris from a large Sentinel-2 image of Europe.

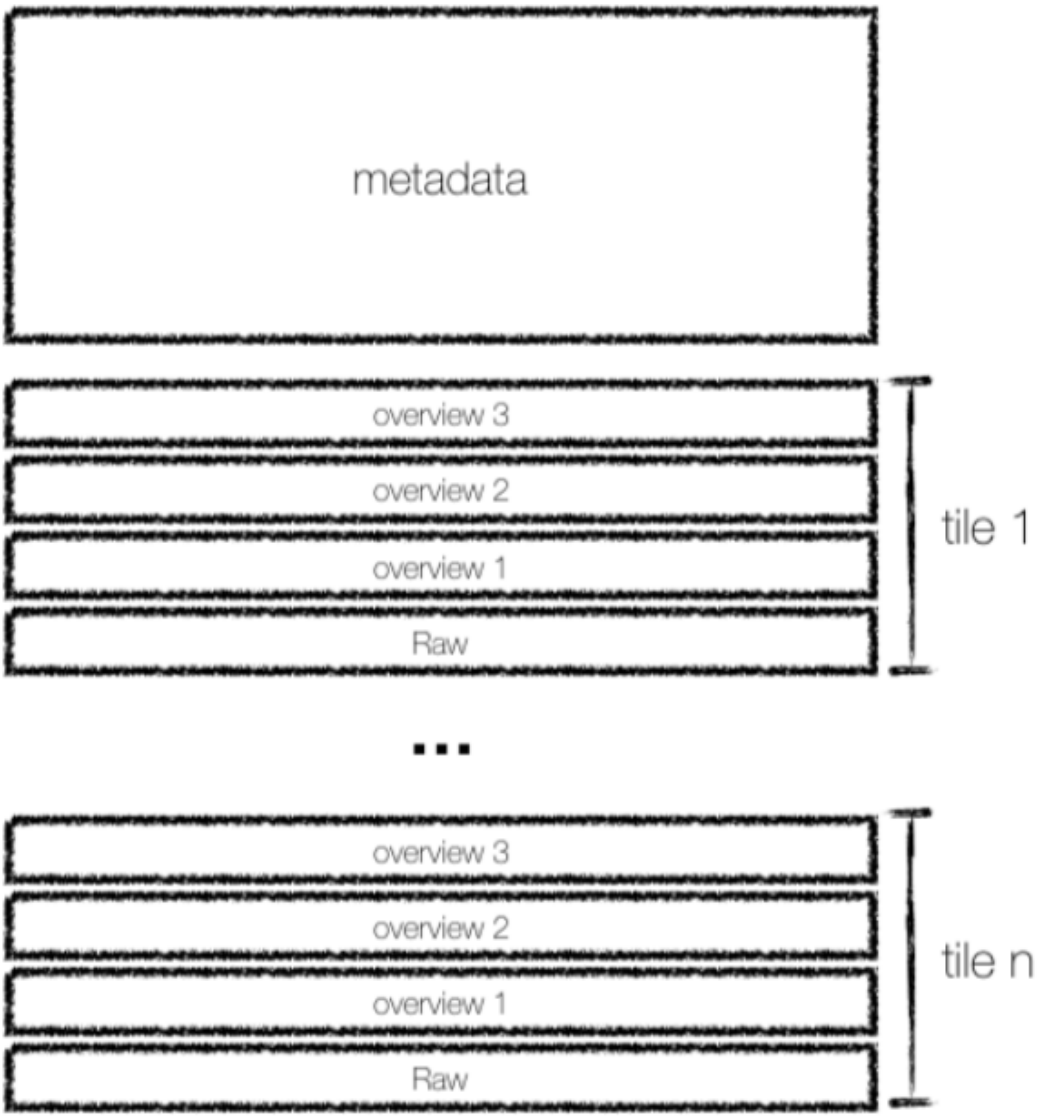

Figure 2.2: COG structure. Retrieved from CNG documentation

2.0.4.2 Multi-dimensional Array Storage with Zarr

While xarray provides the data model for working with n-dimensional arrays, Zarr provides the storage format. Zarr is designed specifically for storing and accessing large n-dimensional arrays in the cloud by:

- **Chunking**: Breaking large arrays into smaller pieces that can be accessed independently
- **Compression**: Each chunk can be compressed individually for efficient storage
- **Hierarchical Organization**: Arrays are organized in groups, similar to folders in a filesystem
- **Cloud-Native Access**: Optimized for reading partial data over HTTP
- **Parallel I/O**: Multiple chunks can be read or written simultaneously
- **Self-Description**: Rich metadata is stored alongside the data using JSON

This makes Zarr particularly well-suited for cloud-based Earth observation data, where datasets often combine multiple dimensions like space, time, and spectral bands.

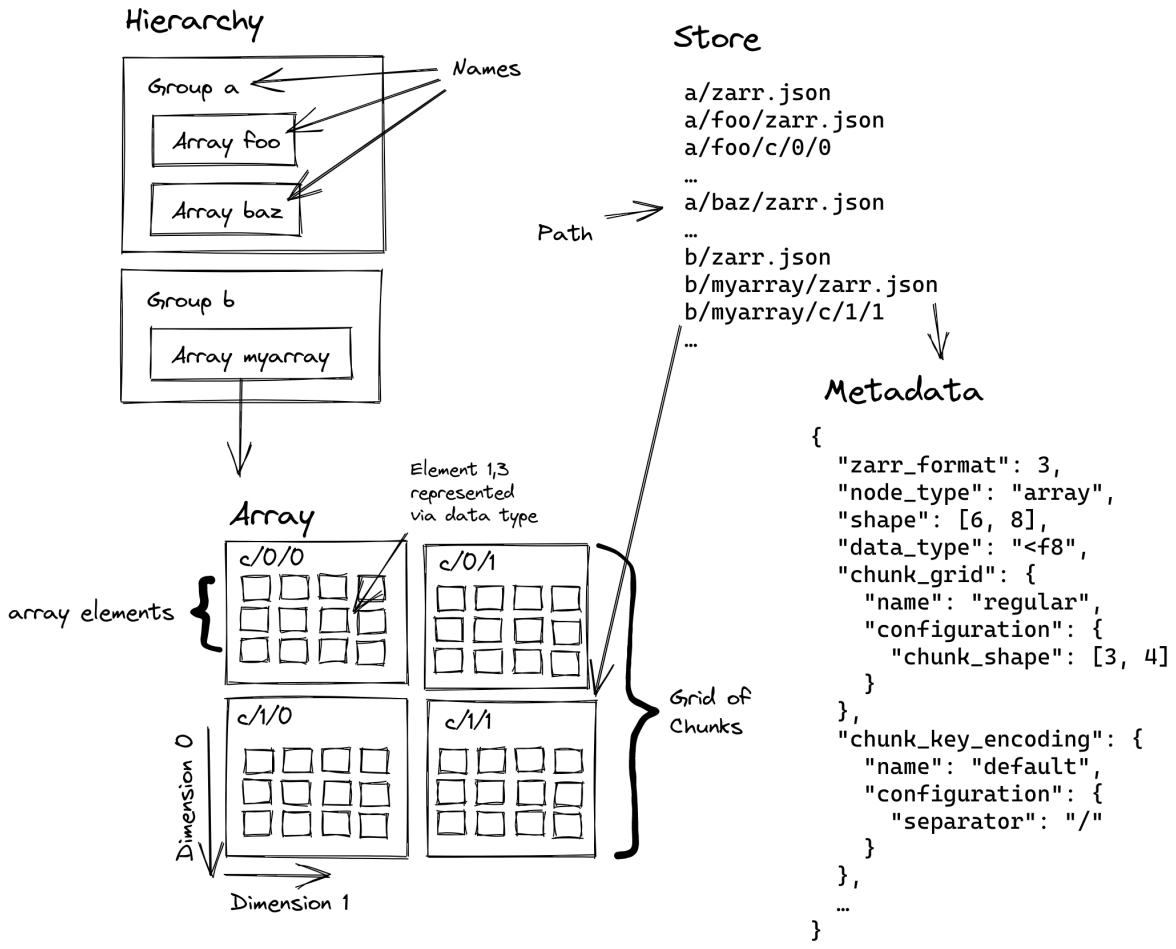

Figure 2.3: Zarr's hierarchical organization showing stores, groups, arrays, and chunks

2.0.5 When to use COGs versus Zarr?

The table below compares some features of COG and Zarr:

Feature	Zarr	COG
Structure	Multi-file chunks	Single file
Access	Parallel	Sequential
Compression	Differently per-chunk	Whole-file
Scales	Multi-scale in single file	Separate, pre-generated lower-resolution files

Based on the structure and capabilities for each format, COGs are used when:

- you work with two-dimensional raster data (like satelliteimages or elevation models)
- you need to easily visualise or access specific geographic areas without loading the entire dataset.
- interoperability with existing GIS software is important, as COG is a widely adopted standard.

On the other hand, Zarr is more often used when:

- you deal with large, multi-dimensional datasets that might be updated or modified.
- you performing complex analyses that involve accessing different parts of the data in parallel.
- an efficient handling of different resolutions or variables within a single dataset is required.

> **Note**
>
> Zarr vs COG: Want to learn more about the differences and similarities of COG and Zarr? Then we recommend the following blogpost by Julia Signell and Jarrett Keifer from Element84 where they discuss "Is Zarr the new COG?"

2.0.6 What's next?

Now that we have an idea of the available cloud-optimised formats for satellite imagery and what cloud-optimised means, we will explore the EOPF data products that will become available as part of the EOPF Zarr Sample Service.

3 The EOPF Available Datasets

Re-engineered datasets as part of ESA's EOPF activity are available for exploration via the EOPF Sentinel Sample Service's STAC Catalog.

At the moment data from Sentinel-1, Sentinel-2 and Sentinel-3 missions are being re-processed and made available.

> **!** Important
>
> The re-processing from the Sentinel Missions is an ongoing activity as part of the EOPF Sentinel Zarr Sample Service. This page and our tutorials will continuously be updated as soon as new data products are available.

An overview of the datasets that are being re-engineered for different processing levels is given below.

3.0.1 Available EOPF products

3.0.1.1 Sentinel-1

Sentinel-1 is a radar imaging mission that is composed of a constellation of two polar-orbiting satellites providing continuous all-weather, day and night imagery.

Product	Instrument	Description	Available at
Level-1 GRD	Ground Range Detected	The Sentinel-1 Level-1 GDR products consist of focused SAR data that has been detected, multi-looked and projected to ground range using the Earth ellipsoid model WGS84.	this link

Product	Instrument	Description	Available at
Level-1 SLC	Single Look Complex (	The Sentinel-1 Level-1 SLC products consist of focused SAR data, geo-referenced using orbit and attitude data from the satellite, and provided in slant-range geometry.	this link
Level-2 OCN	Ocean	The Sentinel-1 Level-2 OCN products for wind, wave and currents applications may contain the following geophysical components derived from the SAR data: Ocean Wind field (OWI), Ocean Swell spectra (OSW), Surface Radial Velocity (RVL).	this link

Sentinel-2

Sentinel-2 acquires optical imagery at high spatial resolution (10m to 60m) over land and coastal waters. The mission supports applications such as agricultural monitoring, emergency management, land cover classifications, and water quality.

Product	Instrument	Description	Available at
Level-1C	Multi-Spectral Instrument	The Sentinel-2 Level-1C product is composed of 110x110 km2 tiles (ortho-images in UTM/WGS84 projection). Earth is subdivided on a predefined set of tiles, defined in UTM/WGS84 projection and using a 100 km step.	this link
Level-2A	Multi-Spectral Instrument	The Sentinel-2 Level-2A Collection 1 product provides orthorectified Surface Reflectance (Bottom-Of-Atmosphere: BOA), with sub-pixel multispectral and multitemporal registration accuracy.	this link

Sentinel-3

Sentinel-3 is a mission that regularly measures our Earth's oceans, land, rivers, lakes, ice on land, sea ice, and the atmosphere. Its goal is to keep track of and help us understand how these large parts of our planet change over long periods.

3.0.1.1.1 Ocean and Land Colour Instrument

Product	Product	Description	Available at
Level-1 EFR	Earth Full Resolution	Provides TOA radiances at full resolution for each pixel in the instrument grid, each view and each OLCI channel, plus annotation data associated to OLCI pixels.	this link
Level-1 ERR	Earth Reduced Resolution	The Sentinel-3 OLCI L1 ERR product provides TOA radiances at reduced resolution for each pixel in the instrument grid, each view and each OLCI channel, plus annotation data associated to OLCI pixels.	this link
Level-2 LFR	Land Full Resolution	The Sentinel-3 OLCI L2 LFR product provides land and atmospheric geophysical parameters computed for full resolution.	this link
Level-2 LRR	Land Reduced Resolution	The Sentinel-3 OLCI L2 LRR product provides land and atmospheric geophysical parameters computed for reduced resolution.	this link

3.0.1.1.2 Sea and Land Surface Temperature Radiometer

Product	Data	Description	Available at
Level-1 RBT	Radiance Brightness Temperature	The Sentinel-3 SLSTR Level-1B RBT product provides radiances and brightness temperatures for each pixel in a regular image grid for each view and SLSTR channel.	this link
Level-2 LST	LST: Land Surface Temperature	The Sentinel-3 SLSTR Level-2 LST product provides land surface temperature.	this link

3.0.2 What's next?

In the following chapter, we dive deeper into the advantages of the `Zarr` data format and we start with practically working with EOPF Zarr datasets.

Part II

About Zarr

4 Overview of the EOPF Zarr format

4.0.1 What Is Zarr?

Zarr is an open-source, cloud-native protocol for storing multi-dimensional arrays. It is specifically desgined to work well with cloud storage and larger-scale computing systems and can be seen as a cloud-native alternative to older formats like HDF5 or NetCDF.

Key advantage to traditional formats is that the Zarr specification stores large multi-dimensional arrays in **chunks**, which are smaller pieces of the larger array. Chunks can be accessed individually or multiple chunks can be read and written in parallel, making data access highly **efficient**.

Zarr works across different storage systems, including local file systems, cloud object storage as well as distributed file systems; offering a greater **flexibility** compared to traditional file formats.

In addition, Zarr embeds **metadata** directly alongside the data. This makes Zarr **self-descriptive**, as each data array contains descriptive information about itself, such as data type, dimensions or additional attributes.

> **Note**
>
> Pro tip: Learn more about Zarr in the official Zarr Documentation and the Zarr V3 storage specification

4.0.2 Components of Zarr

Zarr is organised in a **human-readable**, **hierarchical** structure using simple JSON metadata files and is composed of `groups and stores`, `chunks` and `metadata`:

4.0.2.1 Groups and Stores

Groups and **stores** are concepts that allow Zarr to differentiate between (i) where the data is stored (**stores**) and (ii) how it is organised (**groups**). A **group** is a container for logically organising the data, similar to folders in a file system. A **store** defines where the data is stored; it can be e.g. a bucket in the cloud or a directory on a disk.

4.0.2.2 Chunks

Zarr divides arrays into smaller, independent pieces (**chunks**). Through chunking it is possible to retrieve and process specific areas without loading the complete dataset. Its organisation into chunks is the main reason Zarr's high performance. Chunks are saved as binary files inside a `/c` directory and are further organised through nested folder paths based on their index, e.g. `c/0/0/0` for the chunk position `[0,0.0]`.

4.0.2.3 Metadata

Zarr uses descriptive **metadata** to describe the individual arrays but also the full hierarchy of the dataset. Metadata are stored in `zarr.json` files and are available on the array, group and store level. This structured metadata approach makes Zarr datasets **self-descriptive** and easy to navigate.

The graphic below shows an overview of all relevant Zarr components.

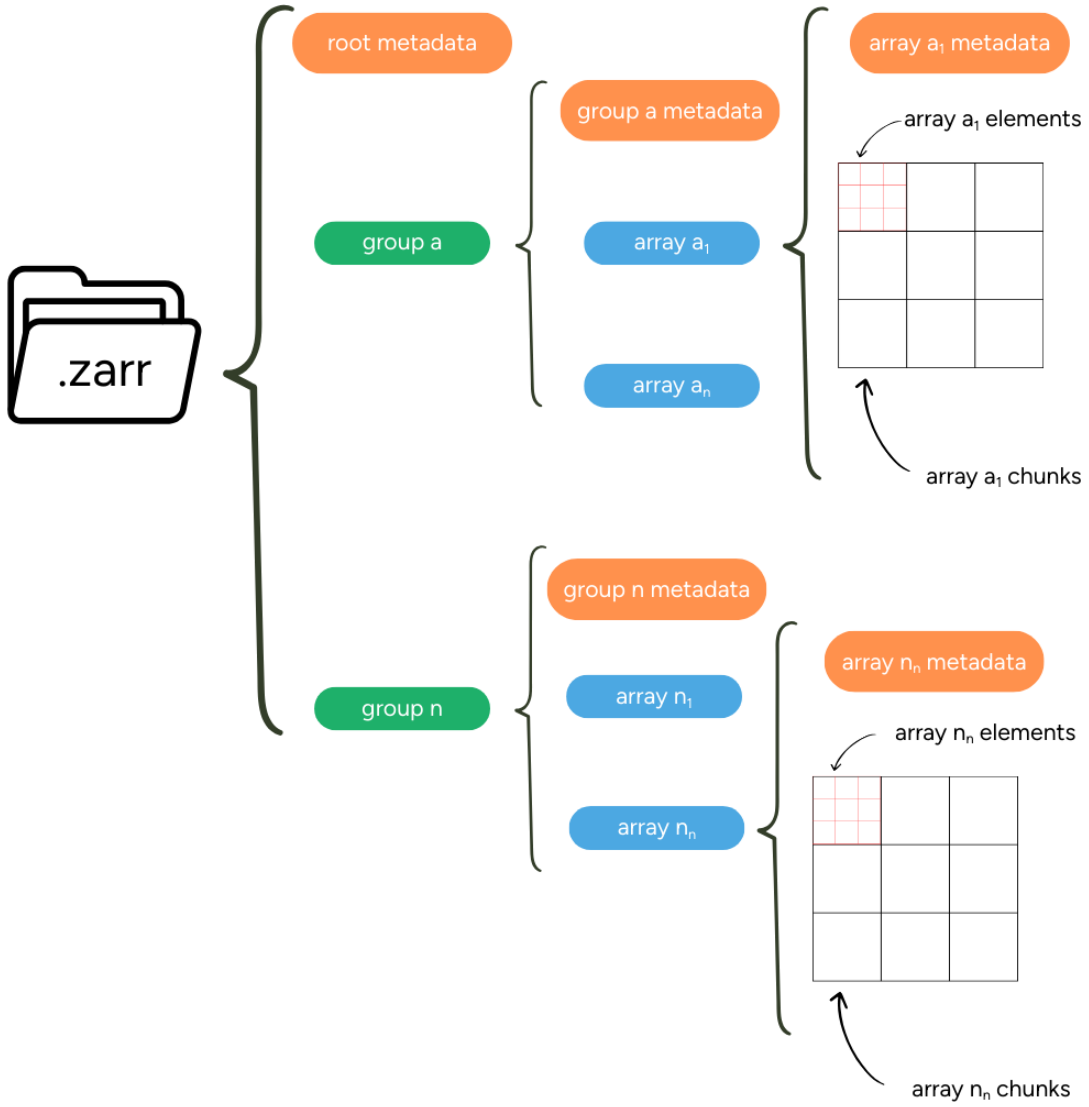

Figure 4.1: Zarr conceptual structure

4.0.3 Zarr EOPF Format Structure

The ESA Copernicus Earth Observation Processor Framework defines `.zarr` as the encoding format for the EOPF catalogue. The Zarr encoding is well aligned with ESA's objective of enhancing the accessibility of Sentinel Data by modernising the previous `.SAFE` encoding into a flexible, cloud-native structure. The cloud-native nature of `zarr` is expected to broaden

the applications of the Sentinel data within the geospatial community while maintaining data quality and established algorithms.

EOPF Zarr products contain of four main groups:

Group	Contents
Attributes	STAC format metadata for the `.zarr` element
Measurements	Main retrieved variables
Conditions	Measurement context (geometric angles, meteorological/instrumental data)
Quality	Flags and quality information for measurement filtering

Let us imagine a Sentinel-2 L2A tile. The tile has dimensions of approximately 10,980 by 10,980 pixels, and include 12 spectral bands (B01 to B12, excluding B10) at different resolutions, plus additional data arrays like a Scene Classification Map (SCL) and an Atmospheric Optical Thickness (AOT) array.

For efficient handling, the data is divided into 1024 by 1024-pixel chunks. This chunking strategy allows for optimal performance when reading specific spatial regions of interest.

Following the defined EOPF Zarr product structure, a Sentinel-2 L2A `.zarr` file is organized as follows:

- Under `attributes`, you'll find:
  - Processing history metadata
  - Chunking configuration
  - Global metadata (acquisition time, sensing time, etc.)
  - Product-specific metadata
- Under `measurements`, the spectral bands are stored at their native resolutions:
  - 10m resolution (r10):
    - B02 (Blue, 490nm)
    - B03 (Green, 560nm)
    - B04 (Red, 665nm)
    - B08 (NIR, 842nm)
  - 20m resolution (r20):
    - B05 (Red Edge 1, 705nm)
    - B06 (Red Edge 2, 740nm)
    - B07 (Red Edge 3, 783nm)
    - B8A (Narrow NIR, 865nm)
    - B11 (SWIR 1, 1610nm)

        * B12 (SWIR 2, 2190nm)
    - 60m resolution (r60):
        * B01 (Coastal aerosol, 443nm)
        * B09 (Water vapour, 945nm)

- Under `quality`, you'll find quality indicators including:
    - Scene Classification Layer (SCL)
    - Quality flags for each band
    - Detector footprint
    - Defective pixels masks

- Under `conditions`, there's measurement context information:
    - Sun angles (zenith, azimuth)
    - Viewing angles
    - Mean solar irradiance
    - Atmospheric parameters like:
        * Aerosol Optical Thickness (AOT)
        * Water Vapor (WV)
        * Cloud and snow probability

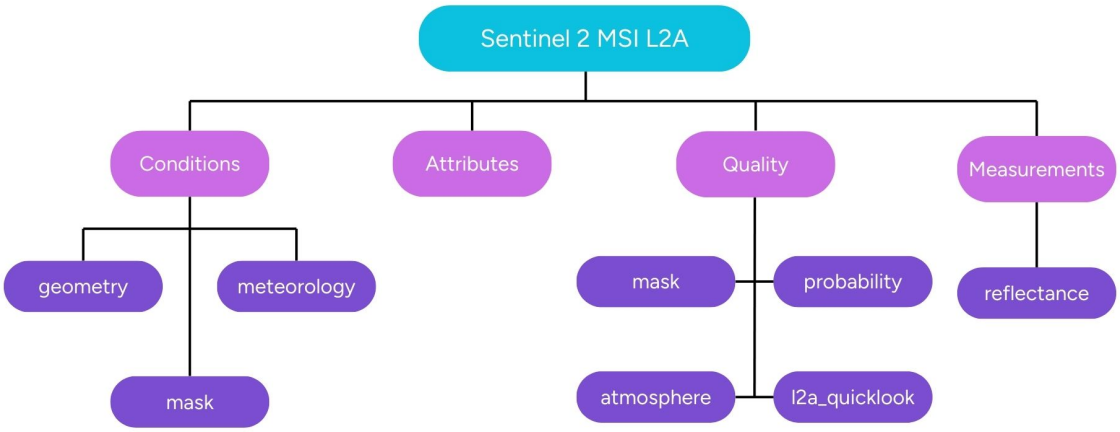

Figure 4.2: Overview of EOPF Zarr product structure on the example of Sentinel-2 L2A

This EOPF Zarr organisation for Sentinel-2 L2A data allows for efficient access to individual

bands or specific spatial regions without loading the entire dataset, making it ideal for large-scale geospatial analysis. It also ensures all relevant metadata is co-located with the data it describes, enhancing data discoverability and usability.

> **Note**
>
> Zarr Deep Dive: Dive deeper into the benefits of Zarr in a blogpost by Lindsey Nield from the Earthmover team: Fundamentals: What is Zarr? A Cloud-Native Format for Tensor Data.

4.0.4 What's next?

Part III

[COMING] EOPF and STAC

Part IV

[COMING] Tools to work with Zarr

Part V

[COMING] EOPF in Action

5 Glossary

Here we introduce some important terms that are mentioned throughout the present book.

EOPF: Earth Observation Processing Framework

CDSE: Copernicus Data Space Ecosystem

CMP: Core Python Modules

HEALPix: Hierarchical Equal Area isoLatitude Pixelation.

GDR: Ground Range Detected

SLC: Singlme Look Complex

NRB: Normalized Radar Backscatter

SAFE: Standard Archive Format for Europe

STAC: Spatio Temporal Asset Catalog

References