Committee on Earth Observation Satellites

Jupyter Notebooks Best Practice

Version 1.1

November 2024

Authored by the CEOS Working Group on Information Systems and Services

Document History

Issue	Date	Comment	Editor
0.5	October 2023	Circulated draft to CEOS WGISS Working Group for comment	Esther Conway
0.6	October 2023	Draft reviewed by participants of the WGISS-56 "Jupyter Notebooks as a common resource workshop" input and review comments provided by United Kingdom Space Agency, National Centre for Earth Observation, United Nations Office for Outer Space affairs, GEO- Secretariat, ESA, NASA, Plymouth Marine/Laboratory/NEODAAS, EUMETSAT, CNES and CSIRO	Esther Conway
0.7	November 2023	Addition of EUMETSAT metadata exemplar to Annex C	Esther Conway
0.8	November 2023	Addressing comments by ESA/Spacebel on access and discovery issues	Esther Conway
0.9	October 2024	Additional Notebook Exemplars added to Annex B First Version Circulated at WGISS-58	Esther Conway
1.0	November 2024	Circulated to WGISS Exec for comments	Esther Conway
1.1	January 2025	Addressing comments from Auspatious/Geosciences Australia and CEOS Executive Officer	Esther Conway

Table of Contents

1. Introduction

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualisations and narrative text. Uses include data cleaning and transformation, numerical simulation, statistical modelling, data visualisation, machine learning, and much more. A Jupyter Notebook allows you to combine rich documentation, live and adaptable code, and data visualisations. It can also be used as a tool to share your data analysis with others, collaborate, teach, and promote reproducible science.

The Jupyter Notebook began as an IPython Notebook; this use of Python makes it an ideal starting point for many people using EO data for the first time. In addition, it currently supports around 40 programming languages, including Python, R and Julia (Ju-pyt-R). We believe it may be a useful tool for a broad range of CEOS agencies and EO data users for the following reasons:

- This technology lowers the barrier to use, Jupyter notebooks can be provided alongside data in an executable form (Click and Run)
- It provides a good starting point for users wishing to improve their Python skills
- Users are able to obtain meaningful results from EO data quickly
- There are a lot of freely available training materials to support users and help improve skills
- It can be integrated with data cubes and CEOS Analysis Ready Data
- JupyterLab and JupyterHub technologies mean that notebooks can be deployed as a web service or multi-user environment
- It can be used to train large classes, simplifying access to a computation environment
- Notebooks can be run on standalone computers/laptops and can be supplied alongside the relevant data; which is helpful in regions where there are bandwidth/data download issues.

From recent CEOS WGCapD and WGISS meetings, we have seen how a number of different CEOS agencies are employing Jupyter notebooks in several different ways. In a 2021 survey conducted of CEOS agencies, the community strongly indicated there was a need to develop a best practice guidance to ensure the optimal reuse of these valuable but technically fragile research assets.

For more information on this technology, please go to the Jupyter Notebook Website.[1]

[1] Jupyter Notebooks Website, https://jupyter.org/, [accessed 5th November 2024]

Purpose of the document

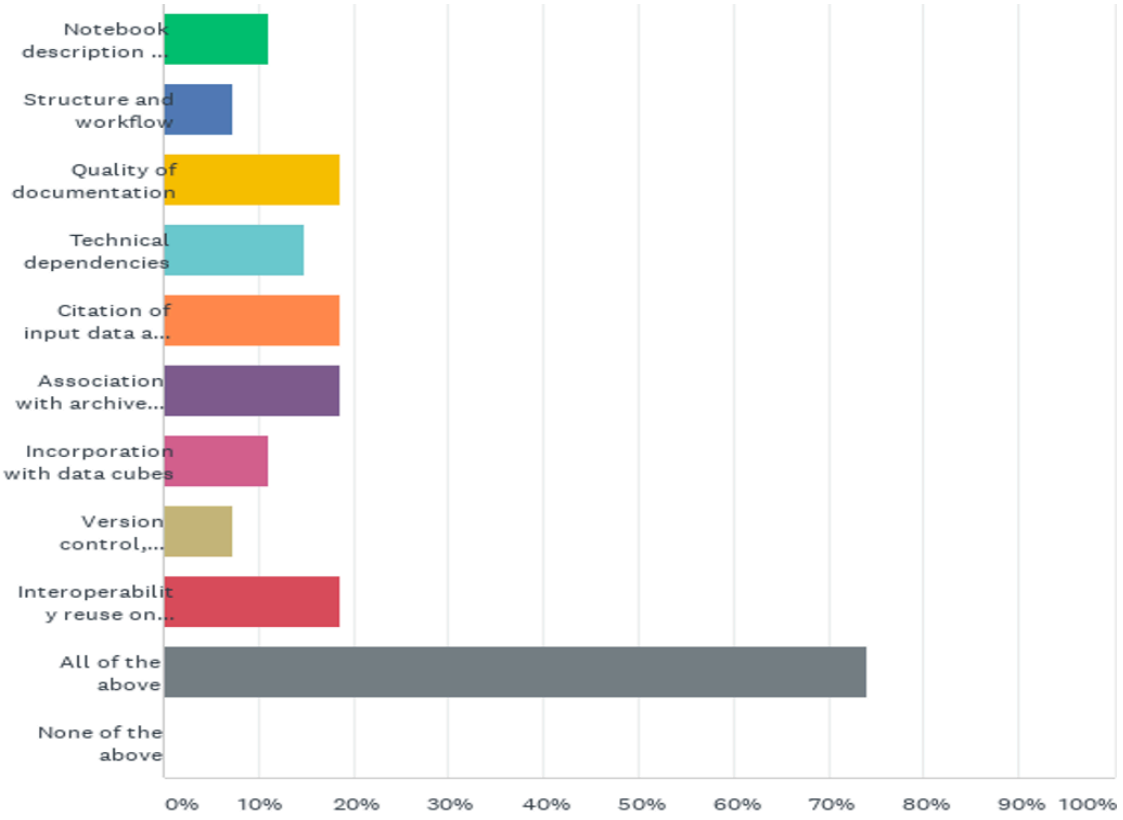
While Jupyter Notebooks can be a valuable resource, there are issues surrounding input data, processing, technical dependencies and quality. Poor quality notebooks with hidden dependencies may cause new users a lot of problems. The purpose of this document is to encourage the development of high quality Jupyter notebooks, which are of genuine benefit to the broader community, in order to ensure their discoverability, accessibility and reusability.

2. Background

In 2019 discussions emerged between the Working Group on Information Systems and Services (WGISS) and the Working Group on Capacity Building and Data Democracy (WGCapD) of CEOS. The question was asked if there was an emerging technology that could facilitate broad exploitation of EO data across many domains and countries. It became evident that Jupyter Notebooks were employed across many of our systems and could provide an easy entry point for many new users, in addition to providing a record of scientific research. Jupyter Notebooks were also beginning to emerge as tools and training materials in their own right. In order to understand the challenges faced by the EO community and envisage how they could be supported by CEOS, members of WGISS conducted a survey of agencies involved in the two working groups. We received responses from 52 individuals, and it emerged that the two primary community needs were for:

1. An awareness raising webinar for the broader EO community.
2. A best practice that covered Jupyter Notebooks for tools/training materials, as well as a record of research.

The survey showed the need for a broad range of topics to be covered by such a best practice. In Q11 we posed the following: Do you think the CEOS community would benefit from best practice advice on a set of defined topics? The results indicated a strong community need for such guidance and the clear identification of gaps which need to be addressed.



Responses to Question 11 from the WGISS Jupyter Notebooks Survey in 2019

The CEOS WGCapD[2] and WGISS[3] ran a joint webinar on Jupyter Notebooks for Capacity Development[4] on the 21st July 2021. The aim of this webinar was to introduce space agencies and environmental organisations worldwide to Jupyter Notebooks and take a tour of emerging services from CEOS Agencies and their applications. We illustrated how Jupyter Notebooks can be used to support capacity development and the exploitation of Earth observation data by a broad range of users. There were two sessions via Zoom to allow for global attendance. We also presented the preliminary topic area to 536 registrants of the webinar and conducted an in depth follow up survey. Over 94% of respondents endorsed the need for a best practice covering the identified topic areas. We then held 2 dedicated workshops on the 21st October 2022 and 26th October 2023 which led to the outline of the technical content presented in section 4 of this document.

[2] The CEOS Working Group on Capacity Building and Data Democracy website, https://ceos.org/ourwork/workinggroups/wgcapd/ [accessed 5th November 2024]
[3] The CEOS Working Group on Information Systems and Services website, https://ceos.org/ourwork/workinggroups/wgiss/ [accessed 5th November 2024]
[4] Jupyter Notebooks for Capacity Development website, , https://ceos.org/meetings/jupyter-notebooks-for-capacity-development-webinar/ [accessed 5th November 2024]

3. Objectives and Needs

The target audience for this Best Practice would be the following groups:

1. Data Custodians
2. Authors of EO Jupyter Notebooks intended for reuse
3. Providers of EO data training
4. Users of EO data
5. Providers of Data Analysis Infrastructure

Issues to be addressed by the document are as follows:

1. Clear understanding of the purpose of a notebook and deciding if it is a reusable asset. Successfully conveying how and why a notebook should be used by its intended community.
2. Encourage the development of suitable workflow and structure within notebooks along with quality documentation. Support their reuse and adaptation by new users.
3. Support discovery of relevant notebooks in terms of dataset, domain, application/function and skill level.
4. Delaying technical obsolescence and ensuring longevity of relevant notebooks.
5. Maintaining the quality of archive by timely retirement of redundant notebooks.
6. Lowering barriers for EO data exploitation and raising technical skill level.

4. Jupyter Notebook Best Practice Content

4.1. Notebook description, purpose and discoverability

The treatment of Notebooks should be highly dependent on their purpose and value. GitHub reached the milestone of 1 million Jupyter Notebooks in 2020. This proliferation of notebooks within EO raises a challenge for the community. The sheer number of available notebooks necessitates a highly selective approach which clearly identifies notebooks of value to be brought under active management. To achieve this the purpose of notebooks must be assessed. They tend fall into two categories:

1. Notebooks produced as part of the scientific process and record, which are by their very nature static.
2. Notebooks that were created as tools to support the exploitation of EO data. These have the potential to evolve and be ported across systems.

Every notebook should have a clear title and abstract which describes who generated it, what it does and its intended purpose.

Notebooks that were created as part of the scientific process and record, need to clearly and unambiguously describe published data they generated and other research outputs, so they can be associated with them. If a notebook is expected to generate multiple datasets or research outputs by multiple users, it should be considered a tool.

Notebooks that are considered to be tools require additional descriptive information to make them discoverable for new users. Additional information on input datasets, domain/thematic area, application, function and skill level are desirable to ensure that notebooks are targeted at the correct communities.

To achieve a higher level of discoverability, it is recommended that CEOS agencies compile metadata in compliance with the CEOS Service Discovery Best Practice[5]. It is recommended that it is embedded as machine-readable metadata in the "metadata" section of the top-level structure of the Notebook. It is recommended to use the schema.org[6] (CreativeWork) encoding for the embedded metadata. See Annex C for more information and an embedded metadata example. Adherence to the subsequent sections of this best practice will generate sufficient metadata to ensure minimal compliance.

[5] CEOS Service Discovery Best Practice, https://ceos.org/document_management/Working_Groups/WGISS/Documents/WGISS%20Best%20Practices/CEOS-Service-Discovery-Best-Practices.pdf , [accessed 5th November 2024]
[6] Schema.org website, Https://schema.org , [accessed 5th November 2024]

4.2. Structure, workflow, code and documentation

When designing a Jupyter Notebook suitable for publication it is important to fully consider the structure and workflow. The notebook should present a cohesive and clear narrative to the user, breaking the process down in logical manageable steps. Use descriptive markdown headers to organise your notebook into sections that can be used to easily navigate the notebook, and if the notebook is long, consider adding a table of contents. Descriptive text can be split between markdown and code comments as appropriate.

The Jupyter Notebook can be considered to be a hybrid of documentation and software. As such su software engineering principles should be used when writing and maintaining your Jupyter notebooks. Writers should try to achieve maintainability, dependability, efficiency and usability of code. While not all principles can be followed due to the ambiguous execution order of cells, the following should be considered

- Naming of notebooks: Remember to give your notebook and associated files logical names which indicate what they are.
- Functions and Modules: Modularise code and use markdown above cells.
- Put low-level documentation in code comments.
- Avoid cells in excess of 100 lines.
- Use unit testing[7]
- Use standard libraries that can readily be deployed in new environments, where possible.
- Use a linting tool to check and format your code consistently, for example, Python Black supports Jupyter[8]

The level of documentation required is determined by the purpose of the notebook. For notebooks that are scientific records that also ensure the reproducibility of the science, the associated documentation should provide a narrative that guides users through the notebooks and provides a clear explanation of the function of code contained in cells.

[7] Online Jupyter Notebooks tutorial on unit testing, https://jupyter-tutorial.readthedocs.io/en/24.1.0/notebook/testing/unittest.html ,[accessed 5th November 2024]
[8] Python Black, https://github.com/psf/black, [accessed 20 November 2024]

However when writing notebooks for use in Data Science Education or Tool for Data Exploitation, authors should adhere to additional advice given in Five Guiding Principles to Make Jupyter Notebooks Fit for Earth Observation Data Education, Wagemann et al 2022[9]. This guidance advises to follow a literate programming paradigm by a text/code ratio of 3 and then use instructional design elements to improve navigation and user experience.

4.3. Technical dependencies and Virtual Environments

All notebooks have the common dependencies that should be explicitly stated within the header of the main .iypnb file. This is to ensure easy extension, portability and maintenance of the notebooks.

4.3.1. Notebook Specific Dependencies

Language versions: Jupyter notebooks support Python, Java, R, Julia, Matlab, Octave, Scheme, Processing, Scala, and many other languages. It is essential to note not only the language but also the version employed in the notebooks. This info is typically already included automatically by the authoring tools in the top level (machine readable) metadata, but may need to be added in the human-readable content of the notebook.

Libraries: Libraries used by the notebook should be clearly identified at the beginning of a notebook. Notebooks are being run on standard machines on data analysis platforms, meaning sometimes libraries need to be installed on the fly: for example, using conda. When this happens, the first cell should be used for that purpose. The installation mechanism should be clearly noted and explained.

Additional scripts and files: Frequently notebooks utilise external scripts, shapefiles and other types of file. The header should contain clear information on their name, location and purpose. Ideally these should be kept in the same repository directory as the notebook rather than being linked to it.

4.3.2. Platform and Environment Specific Dependencies:

A number of CEOS agencies are running systems and services that support Jupyter Notebooks. Many notebooks for training and tools for exploitation are now created on these systems. This places a responsibility on systems developers to publish information that will allow identification of a technical dependencies environment or service which notebooks can cite in their header. It is a recommendation that such information is made available. While the examples below are not an exhaustive list of service and platforms that can support Jupyter

Stand alone computers: JupyterLab can be installed and run on personal laptops and computers. These notebooks are commonly generated off-platform on local computers,

[9] Five Guiding Principles to Make Jupyter Notebooks Fit for Earth Observation Data Education Wagemann et al 2022, https://www.mdpi.com/2072-4292/14/14/3359, [accessed 5th November 2024]

where the input data is easily downloadable and no large scale processing is required. While you could describe the operating system (OS) and environment in the header, a more elegant solution would be to use a .yml file to recreate the environment. This should be kept in the same repository directory as the notebook. Alternatively you could include the Ipykernel library in the dependencies. You can then use the environment in the Jupyter notebook if suited to the needs of your notebook.

JupyterHub deployed on Data Analysis Platform: Many CEOS Agencies have deployed JupyterHub on their data analysis platforms. Very often this is deployed on specific analysis machines whose environments can be published and version information added to the notebook header. For example, the following JASPY environment information[10] is published for the JASMIN platform.

Zero Install Environments: Many notebooks can readily be ported to zero install environments such as Binder[11] or Google Colab[12]. If this is the case, use an environment .yml file which should be archived in the repository directory with your notebook.

[10] Jaspy environment information website, https://help.jasmin.ac.uk/docs/software-on-jasmin/jaspy-envs/ , [accessed 5th November]
[11] Binder website, https://mybinder.org/, [accessed 5th november 2024]
[12] Google Collabs website, https://colab.google/, [accessed 5th November, 2024]

4.3.3. Data Access and Service Dependencies

OpenDap Services: In addition to running on a platform, notebooks may also rely upon data transfer services run by an archive or platform. Frequently these services require users to hold accounts or use specific code within cells to enable access. Platforms/archives should publish up to date information on their services and access requirements. The notebook should explain the access pathway and link to such information. For example, the following give details of the CEDA OPeNDAP services[13].

Data Cube Services: The OpenDataCube[14] is a powerful platform for managing and analysing Earth observation (EO) data at scale. It offers efficient cataloguing and organisation of vast EO datasets, along with robust metadata management and data provenance tracking. Its Python-based application programming interface (API) enables high-performance querying and custom analysis development. The platform supports multi-sensor data integration, flexible data access, and scalable processing capabilities. Processing workflows, analysis and visualisation of output are frequently handled by Jupyter Notebooks; the relevance of these notebooks will be dependent on this type of service. OpenDataCube environments need to be managed to provide an index to available data, and a direct Postgres connection available to the environment that the notebook is running in, which limits this to organisational infrastructure.

Cloud Native Geospatial: A range of modern tools and APIs have been developed that support working on the public cloud, and this new paradigm has been termed Cloud Native Geospatial. CEOS agencies such as USGS and NASA now publish an index to their data as a Spatio Temporal Asset Catalogue (STAC[15]) API (which is a standard OGC API for Features REST API). New tools such as odc-stac can be used to directly load data from STAC documents, which means that it's much easier to do work using a datacube without needing privileged access to a database, which the OpenDataCube required. Cloud Native Geospatial data formats, such as Cloud Optimised GeoTIFF (COG), Zarr and Geoparquet facilitate this new way of working, simplifying access to very large data repositories.

Python Communities of Practice: Given the size and complexity of EO, communities such as Pangeo[16] support a variety of software tools that can be used in conjunction with Jupyter notebooks. Pangeo, the OpenDataCube and Microsoft Planetary Computer, for example, use a common set of tools including:

- Xarray[17]: N-D labelled arrays and datasets in Python.
- dask[18]: Distributed arrays and advanced parallelism for analytics, enabling performance at scale.

[13] CEDA OpenDap services website, https://help.ceda.ac.uk/article/4442-ceda-opendap-scripted-interactions ,[accessed 5th November 2024]
[14] Open Data Cube website, https://www.opendatacube.org/ , [accessed 5th November 2024]
[15] Spatio Temporal Asset Catalogue, https://stacspec.org/en , [accessed 5th November 2024]
[16] Pangeo website, https://www.pangeo.io/ , [accessed 5th November 2024]
[17] Xarray Documentation website, https://docs.xarray.dev/en/stable/, [accessed 5th November 2023]
[18] dask website, https://www.dask.org/, [accessed 5th November]

- Zarr[19]: An implementation of chunked, compressed, N-dimensional arrays for Python.
- High-level visualisation libraries for the PyData[20] ecosystem, including HoloViews[21], Plotly, Folium, Matplotlib and CartoPy.

If possible, a notebook should be mapped to the service it is dependent upon. However with highly complex or rapidly evolving service, the maintenance of notebooks by active development teams may be the only feasible approach.

4.3.4. Complex Systems and Dependencies on Active Teams

Many notebooks can also be integrated into complex systems and are dependent upon several technologies. These layers may also be updated frequently for purposes of security and optimization. In such cases it becomes unrealistic to capture and update dependency information frequently enough to ensure a stable and usable notebook. In such cases the onus must lie with the development team to monitor and update with a new version of the notebook.

[19] Zarr website, https://zarr.dev/, [accessed 5th November]
[20] Pydata website, https://pydata.org/, [accessed 5th November 2024]
[21] HoloViews website, https://holoviews.org/, [accessed 5th November]

4.4. Citation, access to and navigation of input data

For the EO community, the most important input will naturally be EO data. It is therefore essential that the correct data can be identified, located, accessed and navigated successfully.

In the first instance it is important to consider the following:

- In a temporary or permanent location (data at the end of a project is often unpublished and in a temporary location).
- Has it been published or is a formal citation available? Has it had a digital object identifier (DOI) issued in line with CEOS Persistent Identifier Best Practice[22]?
- It is important to note that the location of data used may be different from that indicated by a DOI landing page.
- Is it in the primary archive or being held in a secondary archive or platform?
- What is the data structure and is it different to that of the primary archive?
- Are there any tools to support navigation or data cube services to support extraction of specific files?

Once the appraisal is complete the following information should be included in notebook headers:

- Physical location of data
- Citation or DOI where available - indicating if this is a third party of data held on a different analysis platform to that of the primary archive
- Subsection or file types with the dataset that can be used with the notebook
- Structure of data
- Links to navigation tools and info on data retrieval services (i.e. data cube)
- Information on access restriction and where to apply for access (normally dataset landing page or catalogue record)

[22] CEOS Persistant Identifier Best Practice, https://ceos.org/document_management/Working_Groups/WGISS/Documents/WGISS%20Best%20Practices/CEOS%20Persistent%20Identifier%20Best%20Practice.pdf , [acesssed 5th November 2024]

4.5. Association with archived data

A notebook has the potential to act as documentation or representation information (OAIS[23]) for a dataset in addition to being a tool. However due to the fragile technical nature of the software component and therefore assess the role a notebook will perform for a dataset .

Tool: As a tool, a notebook can be directly associated with a dataset provided it passes a minimum quality threshold. It can then be stored in an institution or community repository and linked to the dataset.

Documentation: A project notebook can frequently act as good quality documentation. If it is solely acting in this role, it should immediately be saved with outputs and converted to a PDF for longer term purposes before association with a dataset.

Representation Information: A good quality notebook can also be used as representation information and left in an active state. However, critical representation should also be explicitly stated. For example, the format of a file should never be inferred from import statements, they should instead be part of the formal metadata.

[23] Reference Model for Open Archival Information System, https://public.ccsds.org/Pubs/650x0m2.pdf, [accessed November 2024]

4.6. Archival, Preservation and Retirement

It is anticipated that notebooks will both evolve and become technically obsolete over time. It is therefore important to manage the natural lifespan of a notebook. In addition to the notebook being physically stored in a trusted repository and made discoverable, the threat of technical obsolescence must be managed. A notebook should only be archived and managed in its active state if there is a reasonable expectation that it will remain usable for 5 years. By recording dependencies the chance of this being the case is greatly increased. Notebooks, unlike data, should be considered for medium term preservation only. You will also have removed the most common cause of notebook technical obsolescence.

Review of notebooks

Notebooks under active management should be subject to review. Some archives will have the capability to do this in automated way via unit testing, while others will need to take a more manual approach. This reinforces the need to be highly selective when taking notebooks under active management as opposed to archiving a dormant PDF copy as documentation. The following should be considered:

- Is the notebook still useful and being used?
- Is the platform/computer environment still available or easily recreated?
- Are all services for data acquisition still available and interfaces the same?
- Is the data still available (particularly relevant for commercial data)?
- Is there a better notebook available?
- Has the data been physically moved or restructured in any way?
- Are you releasing a new data analysis platform or Jupyter hub service if this is being used?
- If using binder or other on-demand environments, are these services likely to go away?

Preservation Strategies

Having identified risks and problems, the archivist or custodian of the notebooks should consider the following preservation strategies and perform a cost/benefit analysis of the different approaches. It is also important to make sure that the notebook header is updated with the relevant technical/user information information in the header.

- Deprecate the old version and replace with a new version where appropriate.
- Migrate the code to a new version so that it can be run in a new environment/OS/Platform.
- Recreate the environment; many data analysis platforms will have the ability to spin up new versions of old machines. Emulation and Virtualization strategies could also be considered, but they would tend to be prohibitively expensive for the sake of an individual notebook.

Graceful Retirement

There will naturally come a point when notebooks should be retired (hibernation strategy). The notebook should be saved with outputs, a PDF copy taken and then marked as dormant. This leaves the possibility of the notebook being revived, which should only occur in exceptional cases. Critically, this prevents new users becoming discouraged by malfunctioning notebooks but allows valuable legacy information to be retained. Automatic retirement after a specified time period should be considered by repositories. Retirement of notebooks in a timely fashion is essential to avoid a repository of broken things.

4.7. Open source software licensing

The Jupyter notebook is a challenging digital object in that it is a combination of documentation (a creative digital object) and code (a software object). As such, we advocate the Software Carpentry approach[24] to the assessment of IPR and rights to be asserted using a two level assessment for the documentation and software. For example software carpentry uses the Creative Commons [25] and GNU Software Licences [26]. Unlike Software Carpentry, individuals may wish to restrict access or assert institutional rights, requiring different licences in line with their organisational requirements. However we world encourage the use of CC-BY and Apache 2.0[27] for simplicity.

4.7.1. For Notebooks as Documentation

For the creative and documentation aspect of the work, we recommend you carry out the following Creative Commons assessment. This covers the most common concerns of depositors.

Attribution: Do you want attribution for your work?

Yes. Anyone using my work must include proper attribution.

No. Anyone can use my work, even without giving me attribution.

Commercial Use: Do you want to allow others to use your work commercially?

Yes. Others can use my work, even for commercial purposes.

No. Others can not use my work for commercial purposes.

Derivative Works: Do you want to allow others to remix, adapt, or build upon your work?

Yes. Others can remix, adapt, or build upon my work.

No. Others may only use my work in an unadapted form.

[24] Software Carpentry approach to notebook licensing website, https://reproducible-science-curriculum.github.io/sharing-RR-Jupyter/LICENSE.html , [accessed 5th November 2024]
[25] Creative Commons Website, https://creativecommons.org/share-your-work/ , [accessed 5th November 2024]
[26] GNU Software Licensing Website,https://www.gnu.org/licenses/licenses.en.html , [accessed 5th November2025]
[27] Apache 2.0 license, https://www.apache.org/licenses/LICENSE-2.0 , [accessed 10th January 2025]

Sharing Requirements: Do you want to allow others to share adaptations of your work under any terms?

Yes. Others can share adaptations of my work under any terms.

No. Others must use the same CC licence if they adapt my work.

After this assessment the appropriate Creative Commons licence may be used. Alternatively, an institutionally approved licence may be used which covers these aspects as a minimum. These permissions are key to allowing reuse and extension of a notebook by new users in the community.

4.7.2. Software Considerations

In many cases it will be sufficient to use a creative licence by itself. However for notebooks where there is a substantial presence of novel code over which you wish to exert copyright, additional code specific licences should be considered.

Where notebooks are published and open for reuse, Apache 2.0 software licences should be considered. If desired a disclaimer may also be included; below is an example from Software Carpentry.

"THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

4.8. Publishing, Versioning and DOI

Publishing: To ensure the broadest possible exploitation of notebooks and associated data, notebooks should be deposited in an institutional or trusted public access repository. Where possible, key metadata should be federated to CEOS or other community discovery services. For institutions that prefer not to run their own repository, GitHub[28] and Zenodo[29] offer a potential solution.

Versioning and DOIs: Where possible a DOI should be minted for archived notebook and versioning employed. Versioning has an important role for notebooks and should follow standard software versioning practices. Notebooks that run on different platforms or have been extended by different groups should be considered distinct from each other rather than versions. For notebooks that evolve over time or require technical adaptation, a standard major/minor versioning convention should be used. It should noted that Zenodo now supports versioning [30].

[28] Github website, https://github.com/, [accessed 5th November]
[29] Zenodi website, https://zenodo.org/, [accessed 5th November]
[30] Zenodo versioning blog information,https://blog.zenodo.org/2017/05/30/doi-versioning-launched/ [accessed 5th November]

Annex A: More information on Jupyter Notebooks Platforms and Services

JupyterHub

JupyterHub brings the power of notebooks to groups of users. It gives users access to computational environments and resources without burdening the users with installation and maintenance tasks. Users - including students, researchers, and data scientists - can get their work done in their own workspaces on shared resources which can be managed efficiently by system administrators.

JupyterHub runs in the cloud or on your own hardware, and makes it possible to serve a pre-configured data science environment to any user in the world. It is customizable and scalable, and is suitable for small and large teams, academic courses, and large-scale infrastructure. For more information visit the JupyterHub Website.

Many CEOS agencies have deployed JupyterHub on their data analysis platforms colocated with their EO data holding such as the Jasmin Jupyter Notebooks Service and MAGEO.

Jupyter based training portals

The EUMETSAT TrainHub provides you access to Jupyter notebook training resources offered by EUMETSAT. You can browse and discover notebooks for different thematic application domains, such as Atmosphere, Climate, Land Surface, Marine or Weather.

Binder

Many EO repositories or institutions providing EO training use Binder, which can use a GitHub repository that contains Jupyter notebooks as well as a branch, tag, or commit hash. Launch will build your Binder repository. If you specify a path to a notebook file, the notebook will be opened in your browser after building. Binder will search for a dependency file, such as requirements.txt or environment.yml, in the repository's root directory: see documentation for further details. The dependency files will be used to build a Docker image. If an image has already been built for the given repository, it will not be rebuilt. If a new commit has been made, the image will automatically be rebuilt. For more information visit the Binder website.

Google Colabs

An alternative to the Jupyter notebook are the iPython notebooks used by Google Colabs. These can be readily transferred to run on JupyterHub. Google Colabs allow access to artificial intelligence (AI) tools such as TensorFlow and Gemini models created by Google DeepMind. Gemini models are built from the ground up to be multimodal, so you can reason seamlessly across text, images, code, and audio. Google Earth Engine can also be accessed via Google Colabs supporting the use of a broad range of EO data. For more information visit the website.

Annex B: Jupyter Notebook Exemplars

CEOS WGISS have run a series of webinars and workshops. In the first webinars, we examined a range of examples to confirm our selection of best practice topics and provide a basis for the development of these topics. In the final workshop, we examined a different range of notebooks and EO training scenarios to test our proposed best practice content.

First Webinar Exemplars

Jupyter Hub and Notebooks on Data Analysis Platforms

We looked at two examples from the UK's JASMIN Jupyter Notebook service, which can access over 20 petabytes of data on the CEDA archive. We then explored the Sentinel-5P global archive of data and demonstrated how to use a very basic notebook to use the data and answer valuable questions, e.g. how did pollution levels change in large cities during the Covid-19 pandemic? We also looked at a smaller scale specialist example: regional NCEO biomass maps. This session helped to demonstrate how, in addition to helping users use Jupyter Notebooks to obtain domain-specific information from data, we can also help them learn technical knowledge and skills related to libraries, modules, and shape files.

Presentation: Jupyter Notebooks on Data Analysis Platforms by Esther Conway NCEO-UKSA

Supporting Notebooks: JASMIN Jupyter Notebooks

Open Data Cube and Google Earth Engine – A Jupyter Notebook Sandbox Demonstration

The Open Data Cube (ODC) Google Sandbox is a free and open programming interface that connects users to Google Earth Engine datasets. This open-source tool allows users to run Python application algorithms using Google's Colab Notebook environment. This demonstration showed two examples of Landsat applications focused on scene-based cloud statistics and historic water extent. Basic operation of the tool will support unlimited users for small-scale analyses and training but can also be scaled in size and scope with Google Cloud resources to support enhanced user needs.

Presentation: Open Data Cube Sandbox by Brian Killough CEOS SEO

Digital Earth Australia Sandbox Notebooks:

- Notebooks on GitHub: https://github.com/GeoscienceAustralia/dea-notebooks
- Broader DEA Knowledge Hub: https://knowledge.dea.ga.gov.au/

Digital Earth Africa Sandbox Notebooks

- https://github.com/digitalearthafrica/deafrica-sandbox-notebooks

ESA PGDS Data Cube and Time Series Data

The ESA PDGS Data Cube is a pixel-based access service that enables human and machine-to-machine interfaces for Heritage Missions (HM), Third-Party Missions (TPM) and Earth Explorer (EE) datasets. The pixel-based access service provides the users with advanced retrieval capabilities, such as time series extraction, data subsetting, mosaicking, band combinations, and index generation (e.g. NDVI, anomalies, and more) directly from the EO-SIP packages with no need for data duplication or data preparation.

In addition to the web-based Explorer graphic user interface, the ESA PDGS Data Cube service also provides a Jupyter processing environment to allow users to import, write, and execute code that runs close to the data. This demonstration showcased how to retrieve Soil Moisture time-series using the Jupyter environment in order to generate thematic maps (monthly anomalies map) over an area of interest. The benefits of using the pixel-based service with respect to traditional access services in terms of resource usage were also highlighted.

Presentation: ESA PGDS Data Cube and Time Series Data by Simone Mantovani MEEO

Earth Analytics and Interoperability Lab – Big Data Processing

The CEOS Analytics Lab[31] (CAL) is a platform for CEOS projects to test interoperability in a live Earth Observation (EO) ecosystem. CAL is hosted on Amazon Web Services and includes facilities for Jupyter Notebooks, scalable compute infrastructure for integrated analysis, and data pipelines that can connect to new and existing CEOS data discovery and access services. This demonstration showed how we use Jupyter Notebooks with the Python Dask Library to efficiently compute and perform large-scale analyses (10s GB) with interactive plotting and scalable compute resources in CAL.

Presentation: Earth Analytics Interoperability Lab, Big Data Processing by Matt Paget CSIRO

[31] https://ceos.org/cal/

Final Workshop Exemplars and training scenarios

The final workshop at MISSING LOCATION examined Jupyter Notebooks where we examined active Group on Earth Observation (GEO) and CEOS projects and the training needs of CEOS Agencies.

Paula De Salvo presented the GEO Knowledge Hub archiving Jupyter notebooks for GEO and CEOS projects. Yves Coen and Damiano Guerci presented their vision for FEDEO, ESA registry of tools and the service metadata and discovery best practice, thereby providing valuable insight into how these standards could effectively be used to support the longer term use of Jupyter notebooks. Esther Conway then went on to present the challenges faced by a data archive attempting to archive notebooks born on different platforms within a centralised data archive. David Borges then described the use and challenges of using Jupyter Notebooks within the CEOS Analytics Lab.

Jupyter Notebooks were explored from a training needs perspective, as supporting training by using notebooks as a centralised training resource was thought to be an area of great potential. Dan Clewley presented the use of AI training notebooks within the MAGEO Jupyter hub based platform. This was followed by Ben Loveday, who gave the perspective of EUMETSAT's training hub which is one of the most developed Jupyter notebooks-based training portals. Lauren Childs-Gleason and Kenton Ross provided valuable insight into the potential of Jupyter notebooks from the capacity development perspective, detailing their current situation and training needs. The presentation concluded with Uzma Saeed describing the training needs of NCEO and organisations trying to support the use of EO data on a national level. The full set of presentations from the workshop can be found here.

Annex C: Notebook metadata

It is recommended to embed metadata information in each notebook. A Jupyter Notebook is a JSON file, which can be modified via a text editor. Right at the end of the JSON file is a "metadata section", which can be enriched with a set of metadata keys (see table below).

The following table provides an overview of mandatory and optional metadata keys and a description thereof. Please note that the specified keys are case-sensitive.

OPTION 1: Original example from EUMETSAT

Metadata key	**Description**	**Requirement level**
author	Author of the notebook	mandatory
title	Title of the notebook	mandatory
description	A short description of what the notebook is about (1-2 sentences max)	mandatory
image	link to the thumbnail image (to be provided in the Gitlab / Github repository) - Dimensions: 400px x 250 px	optional
services	Provide information about where the notebooks are published or can be executed	mandatory
tags	A dictionary with a set of additional metadata keys that help to construct the search interface: • domain, e.g. Atmosphere, Ocean, Climate • subtheme, e.g. Atmospheric Composition, Dust, Fire, ... • platform • sensor • service • tags (can be a list of variables e.g. Nitrogen dioxide concentration (tropospheric column) Please note: the keys listed here are not mandatory and have to be defined if available.	optional

The example below illustrates how the above properties (up to the tags section) can be added to the “metadata” property of the notebook file.


```
"metadata": {
   "author": "Julia Wagemann",
   "title": "Explore AC SAF Metop-A/B GOME-2 - Tropospheric Nitrogen Dioxide L2 - Part 1",
   "description": "This notebook is the first of two 'data discovery' modules on AC SAF Metop-A GOME-2 data. It shows you how AC SAF Metop-A GOME-2 Level 2 data are structured and how the variable 'Tropospheric Nitrogen Dioxide (NO2)' can be visualised.",
   "image": "../img/211_img.png",
   "services": {
      "eumetsat": {
         "jupyter": {
            "link": "https://ltpy.adamplatform.eu/hub/user-redirect/lab/tree/20_data_exploration/211_Metop-A_GOME-2_NO2Tropo_L2_load_browse.ipynb",
            "service_contact": "ltpy@meeo.it",
            "service_provider": "MEEO s.r.l."
         },
         "git": {
            "link":"https://gitlab.eumetsat.int/eumetlab/atmosphere/atmosphere/-/blob/master/20_data_exploration/211_Metop-A_GOME-2_NO2Tropo_L2_load_browse.ipynb",
            "service_contact": "training@eumetsat.int",
            "service_provider": "EUMETSAT"
         },
         "colab": {
            "link":"$$$",
            "service_contact": "training@eumetsat.int",
            "service_provider": "EUMETSAT"
         },
```

```
        "binder": {
            "link":"$$$",
            "service_contact": "training@eumetsat.int",
            "service_provider": "EUMETSAT"
        }
    }
},
"tags": {
    "domain": "Atmosphere",
    "subtheme": "Air quality",
    "service": "AC SAF",
    "platform": "Metop-A",
    "sensor": "GOME-2",
    "tags": "Nitrogen dioxide (tropospheric column)"
},
```

OPTION 2: schema.org encoding

The following table provides an overview of mandatory and optional metadata keys and a description thereof. The specified keys are case-sensitive. The keys correspond to properties of a https://schema.org/CreativeWork object.

Metadata key	**Type**	**Description**	**Requirement level**
identifier	Text	Identifier of the notebook.	optional
author	Person | [Person]	Author(s) of the notebook. For example: { "familyName": "Wagemann", "givenName": "Julia" },	mandatory
name	Text	Title of the notebook. For example: "Explore AC SAF Metop-A/B GOME-2 - Tropospheric Nitrogen L2 - Part 1"	mandatory
description	Text	A short description of what the notebook is about (1-2 sentences max). Example: “This notebook is the first of two 'data discovery' modules on AC SAF Metop-A GOME-2 data. It shows you how AC SAF Metop-A GOME-2 Level 2 data are structured and how the variable 'Tropospheric Nitrogen Dioxide (NO2)' can be visualised.”	mandatory
image	URL	Link to the thumbnail image (to be provided in the Gitlab / Github repository) - Dimensions: 400px x 250 px.	optional

potentialAction	[Action]	Provide information about where the notebooks are published or can be executed.
		
		Example 1 (GitLab):
		
		{
		"name": "GitLab",
		"target": "https://gitlab.eumetsat.int/eumetlab/atmosphere/atmosphere/-/blob/master/20_data_exploration/211_Metop-A_GOME-2_NO2Tropo_L2_load_browse.ipynb",
		"provider": {
		"name": "EUMETSAT",
		"email": "training@eumetsat.int"
		}
		}
		
		Example 2 (Binder):
		{
		"name": "Launch Binder",
		"target": "https://mybinder.org/v2/gh/ceos-seo/data_cube_notebooks/master?labpath=%2Fnotebooks%2Fwater%2Fcoastline%2FCoastline_Classifier.ipynb"
		}
		
		Example 3 (Colab):
		{
		"name": "Open in Google Colab",
		"target": "https://colab.research.google.com/github/ceos-seo/data_cube_notebooks/
			optional

		blob/master/notebooks/water/coastline/Coastline_Classifier.ipynb" }	
keywords	[Text | DefinedTerm]	Set of additional keywords (DefinedTerm) from GCMD controlled vocabularies to identify: • Domain and subtheme as science keywords. • Platform (e.g. satellite) • Instrument And additional free text keywords. Each DefinedTerm may include a "name" (mandatory); "@id" (optional) and "inDefinedTermSet" (mandatory) property.	mandatory

domain	DefinedTerm	Example:
		
		{
		"name": "ATMOSPHERE",
		"@id": "https://gcmd.earthdata.nasa.gov/kms/concept/c47f6052-634e-40ef-a5ac-13f69f6f4c2a",
		
		"inDefinedTermSet": "https://gcmd.earthdata.nasa.gov/kms/concepts/concept_scheme/sciencekeywords"
		}	

platform	DefinedTerm	Example: {	

		"name": "METOP-A", "@id": "https://gcmd.earthdata.nasa.gov/kms/concept/8143808e-1005-4fed-a469-c2bd5f1521bf", "inDefinedTermSet": "https://gcmd.earthdata.nasa.gov/kms/concepts/concept_scheme/platforms" }	
instrument	DefinedTerm	Example: { "name": "GOME-2", "@id": "https://gcmd.earthdata.nasa.gov/kms/concept/5eaf2209-904b-49c8-b99f-1e8550cf95d0", "inDefinedTermSet": "https://gcmd.earthdata.nasa.gov/kms/concepts/concept_scheme/instruments" },	
tags	Text	Example: "AC SAF", "Nitrogen dioxide (tropospheric column)"	
license	URL | Text	URL or SPDX identifier of the license, e.g. "https://spdx.org/licenses/MIT" or “MIT”	optional

The example below illustrates how the above properties (up to the keywords section) can be added to the "metadata" property of the notebook file. The "kernelspec" and "language_info" information is typically already present in the metadata object and the additional notebook metadata data can be added in front:


```
{
        "metadata": {
                "identifier": "eum_211_metop-a_gome-2_no",
                "author": {
                        "familyName": "Wagemann",
                        "givenName": "Julia"
                },
                "name": "Explore AC SAF Metop-A/B GOME-2 - Tropospheric Nitrogen L2 - Part 1",
                "description": "This notebook is rhe first of two 'data discovery' modules on AC SAF Metop-A GOME-2 data.",
                "image": "../img/211_img.png",
                "license": "https://spdx.org/licenses/MIT",
                "potentialAction": [
                        {
                                "name": "GitLab",
                                "target": "https://gitlab.eumetsat.int/eumetlab/atmosphere/atmosphere/-/blob/master/20_data_exploration/211_Metop-A_GOME-2_NO2Tropo_L2_load_browse.ipynb",
                                "provider": {
                                        "name": "EUMETSAT",
                                        "email": "training@eumetsat.int"
                                }
                        },
                        {
                                "name": "Git",
                                "target": "https://gitlab.eumetsat.int/eumetlab/atmosphere/-/blob-master/20_data_exploration/211",
```

```
                        "provider": {
                                "name": "EUMETSAT",
                                "email": "training@eumetsat.int"
                        }
                },
                {
                        "target": "$$$",
                        "name": "Google Colab",
                        "provider": {
                                "name": "EUMETSAT",
                                "email": "training@eumetsat.int"
                        }
                },
                {
                        "target": "$$$",
                        "name": "Binder",
                        "provider": {
                                "name": "EUMETSAT",
                                "email": "training@eumetsat.int"
                        }
                }
        ],
        "keywords": [
                {
                        "name": "ATMOSPHERE",
                        "inDefinedTermSet":
"https://gcmd.earthdata.nasa.gov/kms/concepts/concept_scheme/sciencekeywords"
                },
                {
                        "name": "AIR QUALITY",
```

```
                        "inDefinedTermSet":
"https://gcmd.earthdata.nasa.gov/kms/concepts/concept_scheme/sciencekeywords"
                    },
                    {
                        "name": "METOP-A",
                        "inDefinedTermSet":
"https://gcmd.earthdata.nasa.gov/kms/concepts/concept_scheme/platforms"
                    },
                    {
                        "name": "GOME-2",
                        "inDefinedTermSet":
"https://gcmd.earthdata.nasa.gov/kms/concepts/concept_scheme/instruments"
                    },
                    "AC SAF",
                    "Nitrogen dioxide (tropospheric column)"
                ],
                "kernelspec": {
                    "display_name": "Python 3 (ipykernel)",
                    "language": "python",
                    "name": "python3"
                },
                "language_info": {
                    "codemirror_mode": {
                        "name": "ipython",
                        "version": 3
                    },
                    "file_extension": ".py",
                    "mimetype": "text/x-python",
                    "name": "python",
                    "nbconvert_exporter": "python",
                    "pygments_lexer": "ipython3",
                    "version": "3.10.4"
```

```
                }
        }
}
```