**CEOS Jupyter Notebooks Best Practice**

CEOS

Tech Expo Interest Group

Doc. Ref.: CEOS/

Date: XX 20XX

Issue: Version X.X

**Document Status Sheet**

**{to be filled in passed on feedback from workshops and surveys}**

| **Issue** | **Date** | **Comment** | **Editor** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[**1 Introduction 4**](#_rfk7gqopp4rk)

[1.1 Purpose of the document 5](#_7yqbomebi3zx)

[1.2 Document Overview 5](#_nml2e4msjy1w)

[1.3 Acronyms 5](#_nml2e4msjy1w)

[1.4 Definitions 5](#_nml2e4msjy1w)

[1.5 Related Documents 5](#_nml2e4msjy1w)

[1.5.1 Applicable Documents 5](#_nml2e4msjy1w)

[1.5.2 Reference Documents 5](#_nml2e4msjy1w)

[**2 BACKGROUND 6**](#_d2wvr1e3a65t)

[**3 OBJECTIVES AND NEEDS 8**](#_sbbpys804brb)

[**4 JUPYTER NOTEBOOK BEST PRACTICE CONTENT 9**](#_48p20v2b34ju)

[4.1 Notebook description, purpose and discoverability 9](#_waiyyl49e8ak)

[4.2 Structure, workflow, code and documentation 10](#_d1gesxgb0qzl)

[4.3 Technical dependencies and Virtual Environments 11](#_3v0aumf9262c)

[Platform/Service Specific Dependencies 11](#_byw0tkcsn3w2)

[4.4 Citation, access to and navigation of input data 13](#_hqmolkzh4ask)

[4.5 Association with archived data 14](#_phz8vhue0x61)

[4.6 Archival, Preservation and Retirement 15](#_xzjpmbtpmwwq)

[Review of notebooks 15](#_tkcb4v9dzq6o)

[4.7 Open source software licensing 17](#_mj3o021jtu13)

[4.8 Publishing, Versioning and DOI 18](#_utn24zu76a07)

[**Annex A: More information on Jupyter Notebooks Platforms and Services 19**](#_qi401xfcgs50)

[**Annex B: Use case scenarios 20**](#_w6b855vzvvd1)

# 1 Introduction

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more. A Jupyter Notebook allows you to combine rich documentation, live and adaptable code, and data visualizations. 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 user as:

* 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 user and help improve skills
* It can be integrated with data cubes and CEOS Analysis Ready Data
* Jupyter Lab technologies mean that notebooks be deployed as a web service
* It can be used to train large classes (for more advanced countries)
* Notebooks can be run on standalone computer/laptops and can be supplied alongside the relevant data; in regions where there are bandwidth/data download issues.

From recent CEOS CapD 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 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: <https://jupyter.org/>

## 1.1 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. This ensure their discoverability, accessibility and

## 1.2 Document Overview

{To be completed}

{To be completed}

{To be completed}

## 1.3 Acronyms

## 1.4 Definitions

## 1.5 Related Documents

CEOS Service Discovery Best Practice

<https://ceos.org/document_management/Working_Groups/WGISS/Documents/WGISS%20Best%20Practices/CEOS-Service-Discovery-Best-Practices.pdf>

## 1.5.1 Applicable Documents

## 1.5.2 Reference Documents

# 

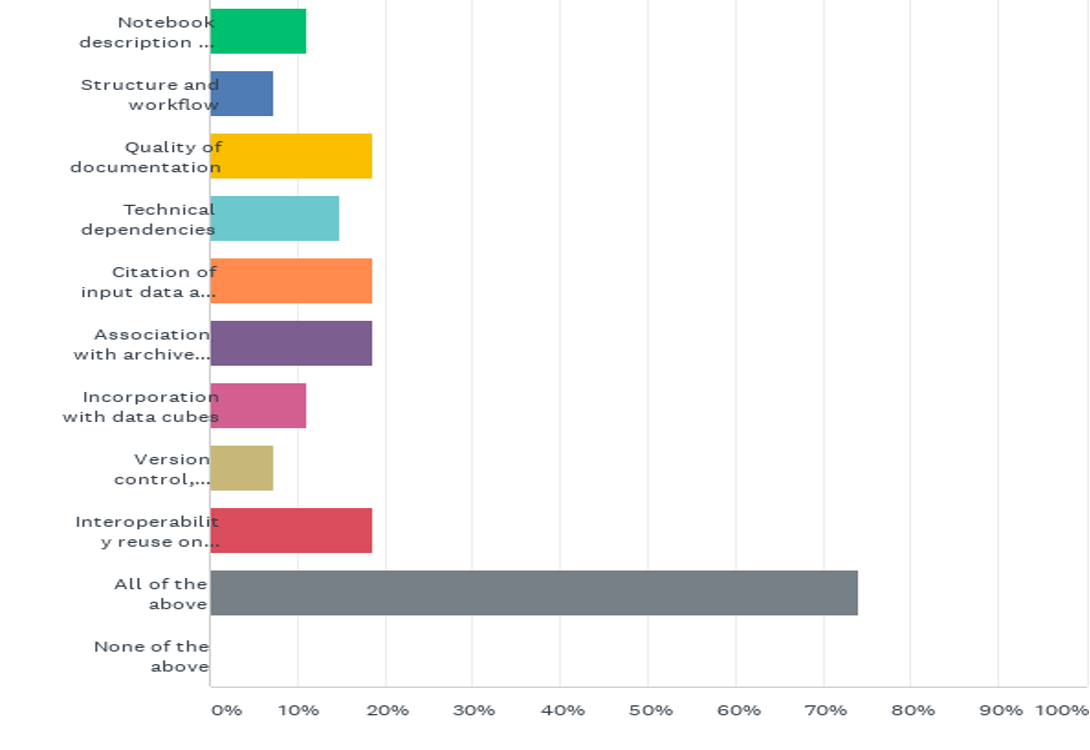
# 2 BACKGROUND

In 2019 discussions emerged between the Working Group on Information Systems and Services and Capacity Development Working Group of CEOS. The question was asked if there was and 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. They 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 we conducted a survey of agencies involved in the two working groups. We receive responses from 52 individuals and it emerged that the two primary community needs where for

1. An awareness raising webinar for the broad EO community
2. A best practice that covered Jupyter Notebooks as 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

Q11: Do you you think the CEOS community would benefit from best practice advice on any of the following



The CEOS [Working Group on Capacity Building and Data Democracy](https://ceos.org/ourwork/workinggroups/wgcapd/) and the [Working Group on Information Systems and Services](https://ceos.org/ourwork/workinggroups/wgiss/) ran a joint webinar on Jupyter Notebooks for Capacity Development on the 21st July 2021. The aim of this webinar was to introduce space agencies and environmental organizations 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 dedicates workshops on the 21st October 2022 and 26th October 2023 to develop the technical content in section 4 [add additional info after final workshop]

# 

# 3 OBJECTIVES AND NEEDS

Who will benefit from a Best Practice

1. Data Producers
2. Authors of Jupyter Notebooks
3. Providers of EO data training
4. Users of EO data
5. EO data archivers
6. Providers of Data Analysis Infrastructure

Issues to be addressed:

1. Clear understanding 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 good 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 notebook
5. Maintaining the quality of archive by timely retirement of redundant notebooks
6. Lowering barrier 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 the 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, application, function and skill level are desirable to ensure that notebooks are targeted at the correct communities. [ask Ben Loveday for EUMETSAT schema amd reference]

To achieve a higher level of discoverabilityIt is recommended that CEOS agencies compile metadata in compliance with the CEOS Service Discovery Best Practice.

<https://ceos.org/document_management/Working_Groups/WGISS/Documents/WGISS%20Best%20Practices/CEOS-Service-Discovery-Best-Practices.pdf>

Adherence to the subsequent sections of this Best practice will generate sufficient metadata to ensure minimal compliance [Check Yves Coen is in agreement with this]

## 4.2 Structure, workflow, code and documentation

When embarking on writing a jupyter notebook it is important fully consider the structure and workflow. The note notebook should present a cohesive and clear narrative to the user breaking the process down in logical manageable steps. Use descriptive markdown headers to organize your notebook into sections that can be used to easily navigate the notebook and add a table of contents.

The Jupyter Notebook can be considered to be a hybrid of documentation and software. As such good 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 sensible 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
* Unit tests: Use unit testing for code in line with <https://jupyter-tutorial.readthedocs.io/en/stable/notebook/testing/unittest.html>
* Use standard libraries that can readily be deployed in new environments where possible

The level of documentation required is determined by the purpose of the notebook. For notebooks that are scientific record that also ensure the reproducibility of science. Documentation should provide an understandable narrative that guide users through the notebooks and provide a clear explanation of the function of code contained in cells.

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 <https://www.mdpi.com/2072-4292/14/14/3359> . Where they advise following a literate programming paradigm by a text/code ratio of 3 and the use instructional design elements to improve navigation and user experience

## 4.3 Technical dependencies and Virtual Environments

**Universal and Notebook Specific Dependencies**

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.

**Platform and Service Dependencies**: All notebooks will run on something whether it is a personal laptop or integrated into a service

**Language versions** :Jupyter notebooks support Python, Java, R, Julia, Matlab, Octave, Scheme, Processing, Scala, and many other languages. It essential to note not only the language but also the version employed in the notebooks

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

**Additional scripts and files**: Frequently notebooks utilize 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 linked to.

## **Platform/Service Specific Dependencies**

Many CEOS agencies are running systems and services that support Jupyter Notebooks. Many notebooks for training and tools for exploitation are now born on this 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 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 computer:** Jupyterlab can be installed and run on personal laptops and computers. These notebooks are commonly generated off-platform on local computers where the input data is easily downloadable and no large scale processing is required. While you describe the OS and environment in the header a more elegant solution would be to use .yml file to recreate the environment. This should be kept in the same repository directory as the notebook. Alternatively you could include Ipykernel library in the dependencies and then adding the kernel to jupyter notebook, 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 machine whose environments can be published and version information added to the notebook header for example: <https://help.jasmin.ac.uk/article/4729-jaspy-envs>

**OpenDap and http 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 <https://help.ceda.ac.uk/article/4442-ceda-opendap-scripted-interactions>

**Zero Install Environments:** Many notebooks are born on or can readily be ported to zero install environments such as Binder Google Collabs or PANGEO. If this is the case use an environment .yml file which should be archived in the repository directory with your notebook.

**Google Colabs:**

ESA PGDS

Get example from Simone Mantovani at Workshop

Open Data Cube Google Sandbox

Get example from Brian Killough/Dave Borge

Earth Analytics Interoperability Lab

Get Exmaple from CSIRO

EUMETSAT TrainHub

MAGEO

## 

## 

## 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 been published or is a formal citation available. Has had a DOI issue in line with CEOS persistent identifier Best Practice <https://ceos.org/document_management/Working_Groups/WGISS/Documents/WGISS%20Best%20Practices/CEOS%20Persistent%20Identifier%20Best%20Practice.pdf>

It is important note that the location of data used may be different from that indicated by a DOI landing page.

* Is 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 a 3rd Party of data held on a different analysis platform to that of the primary archive.
* Subsection or filetypes with the dataset that the Notebook can be used with
* 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)

## 

## 4.5 Association with archived data

A notebook has the potential to act as documentation or representation information (OAIS) for a dataset in addition to being a tool. However due to the fragile technical nature of 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 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 file should never be inferred from import statements, they should be part of the formal metadata.

## 

## 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 the capability to do this in automated way via unit testing other will ned 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 relevance for commercial data)
* Is there a better notebook available
* Has the data been physically moved or restructured in any way
* Are you releasing as new data analysis platform or Jupyter hub service if this being used
* If using binder collabs etc are these services is likely to go away

**Preservation Strategies**

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

* Deprecate the old version and replace with new 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 save with outputs, a PDF copy taken and marked as dormant. This leaves the possibility of the notebook being revived, which should only occur in exceptional case. Critically this prevents new users becoming discouraged by malfunctioning notebook but allows valuable legacy information to be retained. Automatic retirement after a specified time period should be considered by repositories. Retirement of notebooks 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 to the assessment of IPR and rights to be asserted. <https://reproducible-science-curriculum.github.io/sharing-RR-Jupyter/LICENSE.html> using a two level assessment for the documentation and software.

Unlike software carpentry individuals may wish to restrict access or assert institutional rights. As a default the Creative Commons <https://creativecommons.org/share-your-work/> and GNU Software Licences <https://www.gnu.org/licenses/licenses.en.html> may be used.

**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 unadapted form.

**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 use which covers these aspects as minimum. These permissions are key to allowing reuse and extension of a notebook by new users in the community.

**Software**

In many case 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 GNU 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 and other community discovery services [check this with Yves and Damiano]. For institution that prefer not running their own repository GitHub <https://github.com/> and Zenodo <https://zenodo.org/> offer a potential solution

**Versioning and DOI :** 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 earth other rather than versions. For notebooks that evolve over time or require technical adaption a standard major/minor versioning convention should be used. It should noted that Zenodo now supports versioning https://blog.zenodo.org/2017/05/30/doi-versioning-launched/

# Annex A: More information on Jupyter Notebooks Platforms and Services

[Discuss Examples at Workshop]

JupyterHub

EUMETSAT Training Hub

Binder

Google Colabs

Google Earth Engine

ESA PGDS

# Annex B: Use case scenarios

[Discuss Examples at Workshop but they will be used to illustrate best practice content ]