CEEDS

CENTRE OF EXCELLENCE IN ENVIRONMENTAL DATA SCIENCE

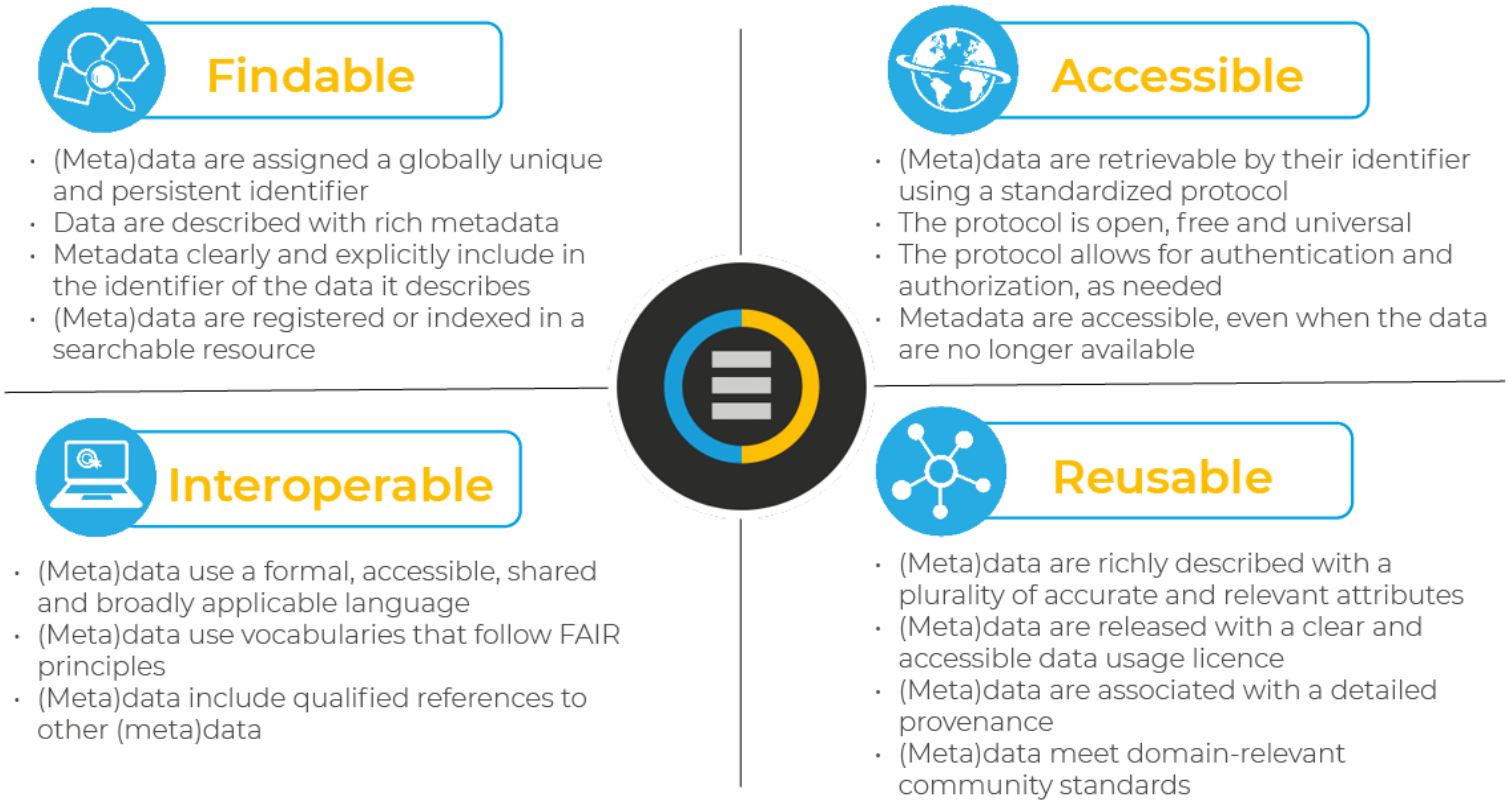
Data access and data APIs

Michael Tso and Matt Nichols (UKCEH)
DSNE Spring school 2024

What are the ways you can access environmental data?

- Download from a data website or repository
    - Data.gov.uk
    - Data Centres (e.g. EIDC, NASA, EU)
    - Repositories (e.g. Figshare, Zenodo)
- Download from a data app
- Direct access from a cloud compute platform (e.g. Google Earth Engine, DataLabs)
- Download using a data API

- FAIR environmental data



https://www.ccdc.cam.ac.uk/solutions/about-the-csd/fair-data-principles

NetCDF(.nc) files advantages:

https://nsidc.org/data/user-resources/help-center/what-netcdf

- **Self-Describing**. A netCDF file includes information about the data it contains.
- **Portable**. A netCDF file can be accessed by computers with different ways of storing integers, characters, and floating-point numbers.
- **Scalable**. A small subset of a large dataset may be accessed efficiently.
- **Appendable**. Data may be appended to a properly structured netCDF file without copying the dataset or redefining its structure.
- **Shareable**. One writer and multiple readers may simultaneously access the same netCDF file.
- **Archivable**. Access to all earlier forms of netCDF data will be supported by current and future versions of the software.

- Even better if they are CF-compliant

https://mapscaping.com/what-is-netcdf/

NetCDF files

```
In [46]: ds.to_netcdf("example.nc")

In [47]: reopened = xr.open_dataset("example.nc")
In [48]: reopened
Out[48]: <xarray.Dataset>
Dimensions: (x: 2, y: 3)
Coordinates:
                * x (x) int64 10 20
Dimensions without coordinates: y
Data variables:
foo (x, y) float64 ...
bar (x) int64 ...
baz float64 ...
```

- R packages: ncdf4, tidync
- Python packages: xarray, rioxarray, also netCDF4, nctoolkit
- ArcGIS and QGIS

WMS tiles

Steps:
- Open the web service URL
- Look for the layer map to use

- Easy way to overlay map layers without downloading

- - R package: leaflet (more help), terra
- - python package: ipyleaflet
- - GIS software

Data Application programming interfaces (API)

- API is a software intermediary that allows two applications to talk to each other.
- A data API allows you to programmatically query and access data on a web server.
- Usually, it powers data apps
- You can specify in great detail what to download >> save time and disk space

Examples:

- ERA5 weather reanalysis
- National River Flow Archive (NRFA)
- COSMOS-UK soil moisture
- OpenAIR air quality
- Environmental Agency (WQ and more)

When to use a data API?

You know your query quite well (e.g. time, location, variables)

Vs.

You need to do complex operations on a majority of the data

- Benefits of using API: when you re-run the API request, you get the most updated data!

Understanding data API calls

- A web request done in a specific way
- Example: https://cosmos-api.ceh.ac.uk/collections/1D/locations/CHIMN
  - Try running it on your browser
  - (typically) returns JSON format, easier to use Python or R to convert to data frame
  - Base_url: https://cosmos-api.ceh.ac.uk
  - "daily collection": collections/1D
  - "at CHIMN site": locations/CHIMN

Tip: read documentation carefully!

Documentation sites

- COSMOS-UK
- National River Flow Archive
- Environmental Agency (e.g. water quality) – and others in api.gov.uk
- ERA5
- openair

Apps that uses data API

- UKSCAPE hydrological sensor data integration tool
- COSMOS-UK data request form

- EA Ecology and Fish Explorer – a Shiny app!

Notebooks (in DataLabs)

- COSMOS-UK API
- EA Water Quality API
- ERA5 API
- Using WMS tiles