InterPlanetary File System plus FAIR Digital Objects: Making Campaign Data Findable and Verifiable

A user-centric software infrastructure for geoscience data using the Interplanetary File System (IPFS) and FAIR Digital Objects (FDOs)

Marco Kulüke[1], Ivonne Anders[1], Karsten Peters-von Gehlen[1], Carsten Ehbrecht[1], Kameswar Rao Modali[1], Beate Kruess[1], Hannes Thiemann[1]

The ORCESTRA campaign

Organized Convection and EarthCARE Studies over the Tropical Atlantic ran from August to September 2024. Satellites, research aircraft, the RV Meteor and ground stations on Cape Verde and Barbados produced large, heterogeneous datasets that had to be shared while the campaign was still running.

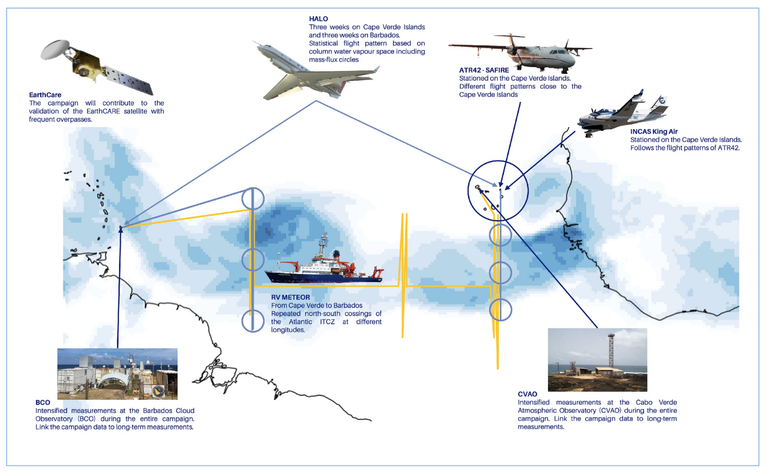


Fig. 1: Platforms and instruments contributing to ORCESTRA.
Source: orcestra-campaign.org

Data challenges in the field

- Many sources, and so many initial storage locations.
- Unreliable internet: minutes in flight, weeks at sea, patchy on the ground.
- Limited local storage on site.
- Data loss from technical defects in a harsh environment.

Objective

Keep campaign data available, verifiable and machine-actionable from the first day in the field through to long-term archival, without a central server the field teams have to reach.

Contribution of DKRZ

If every node holding a dataset goes offline, the dataset is gone. DKRZ runs a permanently available IPFS node that pins the campaign data, guarantees persistence beyond the campaign, and manages storage: monitoring use and allocating capacity to projects.

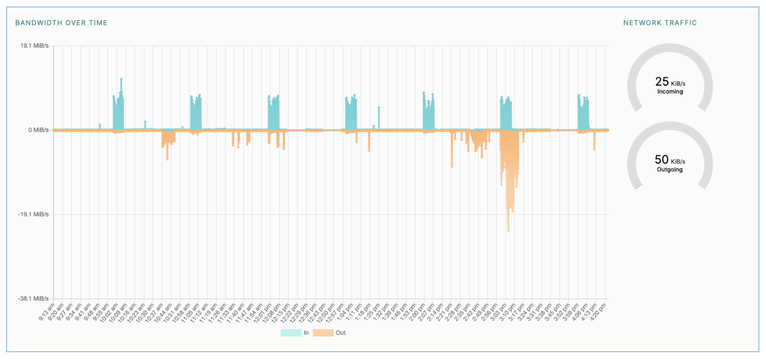


Fig. 2: Incoming and outgoing network traffic over time for the DKRZ IPFS node.

The InterPlanetary File System

A peer-to-peer network protocol rather than a storage system. Every node contributes its own storage and has equal standing, so data can be shared without a central server. Content is addressed by what it is, not where it sits, and version control comes with the protocol.

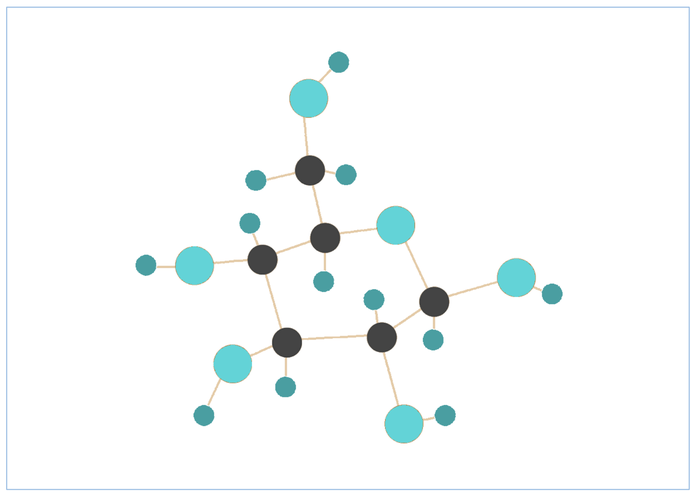


Fig. 3: A decentralised peer-to-peer network: no node is privileged, and any node can serve the content it holds.

Content identifiers (CIDs)

- A CID is a cryptographic hash of the content itself, a fingerprint, not a URL.
- Self-verifying: a node hashes what it received and checks it against the CID it asked for, so tampering is immediately detectable.
- Immutable by design: change one byte and the CID changes completely.
- Deduplication is automatic, identical content always yields the same CID.

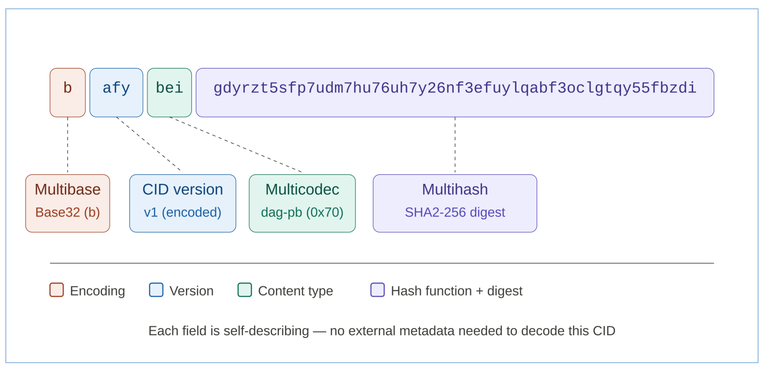


Fig. 4: Anatomy of a CID. Every field is self-describing, so no external metadata is needed to decode it.

In practice

A user picks a dataset in the catalogue and obtains its content identifier. Analysis scripts then open the CID instead of a path, so the same workflow runs unchanged during the campaign and years after it.

```
import xarray as xr

ds = xr.open_dataset("ipfs://bafybeibei32rn7fcyrrhhx3gfsdz2yonze6rpqvvyih2avpxb74qqlqss4", engine="zarr")

ds.sea_surface_temperature.plot()

[<matplotlib.lines.Line2D at 0x147abead0>]
```

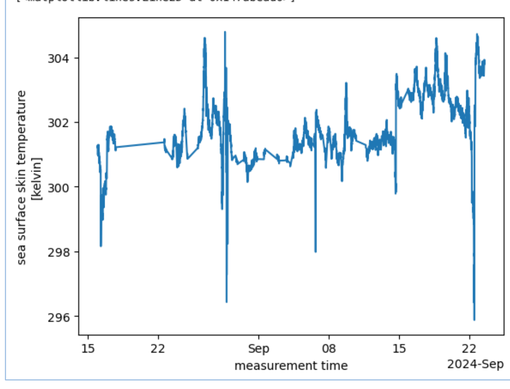


Fig. 5: Opening a dataset by CID in xarray, and the resulting sea-surface temperature series. The script does not know where the file lives.

The SpatioTemporal Asset Catalog (STAC)

The STAC specification organises the data: an Item is a single unit with geometry, datetime and links to files, a Collection groups related Items, and a REST API makes a large catalogue searchable.



Fig. 6: The ORCESTRA campaign data catalogue, served from a STAC Browser, FastAPI and Elasticsearch stack at DKRZ.

FAIR Digital Objects complement IPFS

An FDO bundles data, metadata, documentation and software into one unit a machine can act on. Its persistent identifier (PID) resolves to a record saying what the object is, what you may do with it, which version it is and under what license. The FDO profile fixes which of those fields the type must carry. The IPFS CID adds what that FDO record cannot: proof that the bytes are the ones described, and a name that resolves from any node holding them.

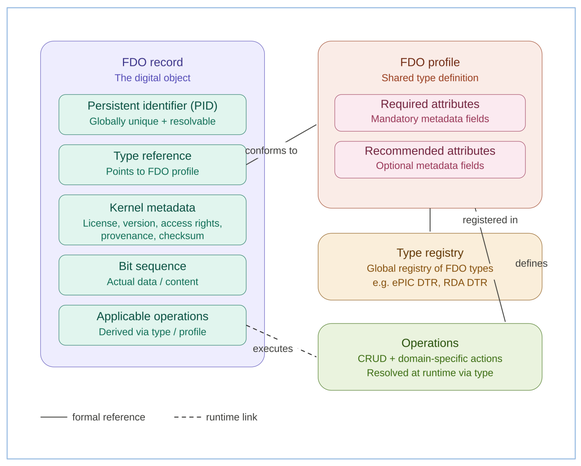


Fig. 7: FDO record and FDO profile: the PID resolves to kernel metadata, a type reference and the operations resolved at runtime.

The ORCESTRA data catalog as FDO

The catalogue is registered as an FDO: a Handle PID resolves to a kernel record naming the type, the profile and the rights, and pointing at the API.

HANDLE RECORD

21.T14995/1554ed15-276a-4ddf-b35d-bd3625c6bb86

URL	https://orcestra.cloud.dkrz.de/api/
name	STAC catalog
FDO_Profile_Ref	21.T11969/2b3dbd8491ca0edc4dc1
FDO_Type_Ref	21.T11969/68398d5511a138742b47
FDO_Genre_Ref	21.1/climate_model_data
FDO_Rights_Ref	2.1/CC0
FDO_Data_MD_Locs	https://orcestra.cloud.dkrz.de/api/

Fig. 8: Resolved Handle Record

CONCLUSION

IPFS and FDOs connect data with their metadata, making a field catalog persistently available and machine-actionable.

[1]Deutsches Klimarechenzentrum (DKRZ), Hamburg, Germany

Contact: Marco Kulüke, DKRZ

kulueke@dkrz.de

Funded by the European Union. This work has received funding from the European High Performance Computing Joint Undertaking (JU) under grant agreement No 101093054.
Also funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), project number 460036893.

References

Benet, J. (2014). IPFS — content addressed, versioned, P2P file system. arXiv:1407.3561

Wittenburg, P., Strawn, G., Mons, B., Bonino, L., Schultes, E. (2019). Digital Objects as Drivers towards Convergence in Data Infrastructures. EUDAT. doi:10.23728/b2share.b605d85809ca45679b110719b6c6cb11

De Smedt, K., Koureas, D., Wittenburg, P. (2020). FAIR Digital Objects for Science: From Data Pieces to Actionable Knowledge Units. Publications, 8. doi:10.3390/publications8020021

Kulüke, M., Peters-von Gehlen, K., Krüss, B., Anders, I. (2026). Report on application of the FDO standard. Zenodo. doi:10.5281/zenodo.18258999

Scan for the ORCESTRA STAC catalogue