

Inspect your own model data directly in the browser.

From cloud storage to interactive visualization: just open a browser and a URL.

Introduction

Earth System Models produce output on a wide range of structured and unstructured grids. In practice, this makes quick inspection difficult: users often need specialized desktop software, access to HPC resources, or preprocessing steps that remap data to regular latitude-longitude grids. Those extra steps slow down exploratory work and may introduce interpolation artifacts.

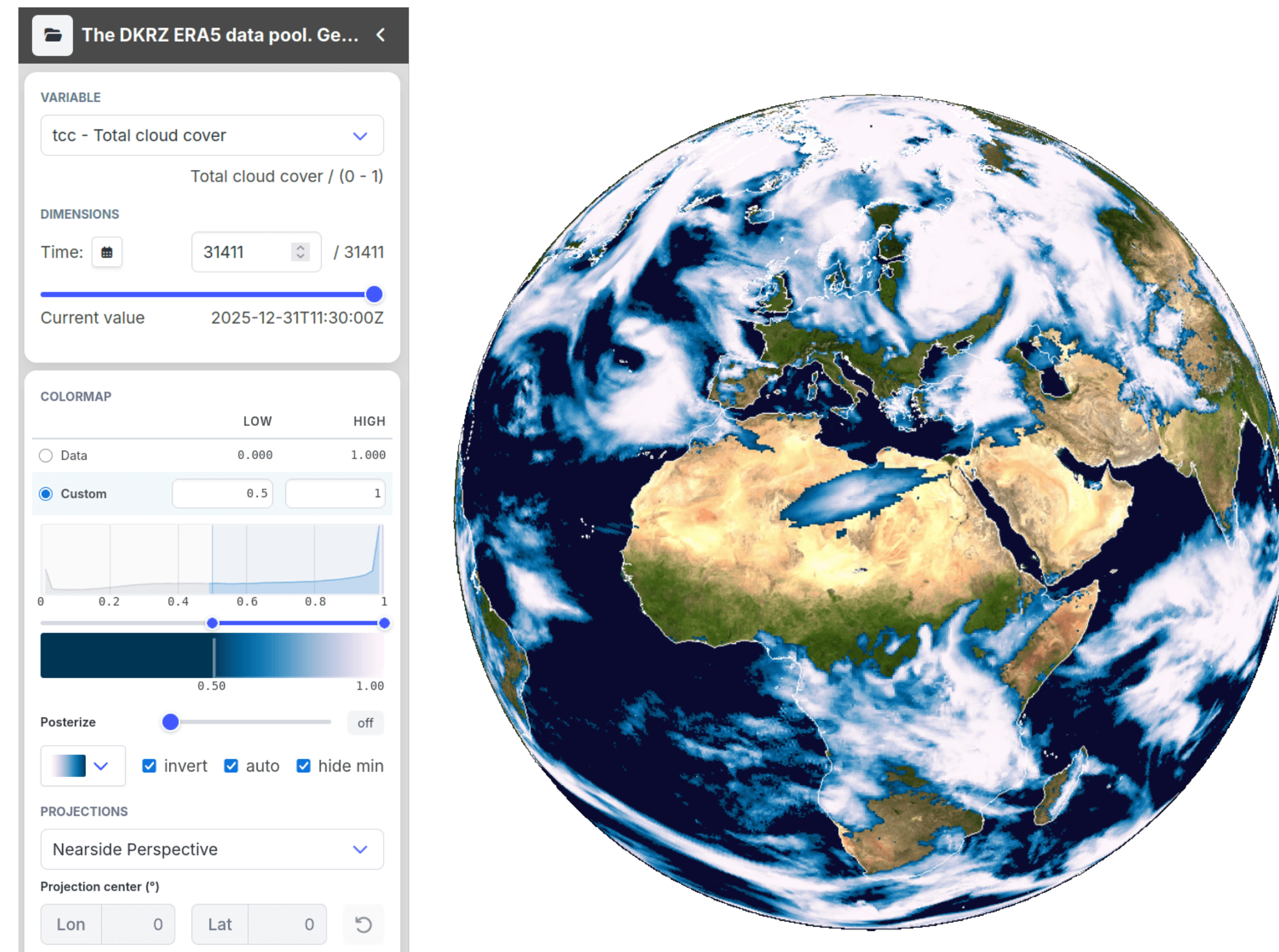


Figure 1. Gridlook displaying a native-grid dataset with controls panel. (Earth-Texture: NASA).

Gridlook addresses this gap with a browser-based WebGL viewer for cloud-hosted Zarr datasets. The tool renders data directly on the native grid in the client, requires no installation, and opens any public, CORS-enabled Zarr store. This supports FAIR-style access through stable URLs, open formats, and reusable metadata [9, 3], complementing Pangeo's cloud-native geoscience ecosystem [2]. Unlike tile-based visualization systems, Gridlook does not pre-render or serve JPG/PNG tiles and does not depend on a continuous server-side rendering connection during exploration.

Key Features

- **Shareable URLs:** dataset location, variable, projection, and view parameters are encoded in the URL for reproducible inspection.
- **Native-grid rendering:** visualize model output on its original grid without regriding.
- **Cloud-native access:** open Zarr stores (v2 and v3) from AWS, Swift, or Google Cloud directly in the browser.
- **Automatic grid detection:** infer grid type from CF-compliant metadata [5].
- **Client-side GPU rendering:** all color mapping and projection happen on the GPU via WebGL — no backend infrastructure required.
- **Multiple projections:** switch between 3D globe and 2D map views to examine spatial structure from different perspectives.
- **Exploration tools:** scientific colormaps, histograms, range selection, hover value readout, coastline and graticule overlays, and snapshot export.
- **Catalogs:** browse grouped datasets via catalog documents

Architecture and Workflow

Gridlook follows a serverless architecture. All rendering happens in the user's browser, while the data remain in cloud object storage. Zarr's chunked layout makes it possible to fetch only the slices required for the current variable, timestep, and view [3].

1. **Open a dataset or catalog:** the user supplies a Zarr URI or a catalog URL.
2. **Inspect metadata:** variable names, dimensions, and grid descriptors are read from the store by using zarrita [7].
3. **Detect the grid:** CF-style coordinates and `grid_mapping` attributes select the renderer [5].
4. **Fetch chunks on demand:** only the selected slices are requested from cloud storage.
5. **Render on the GPU:** color mapping and interaction happen client-side through WebGL.

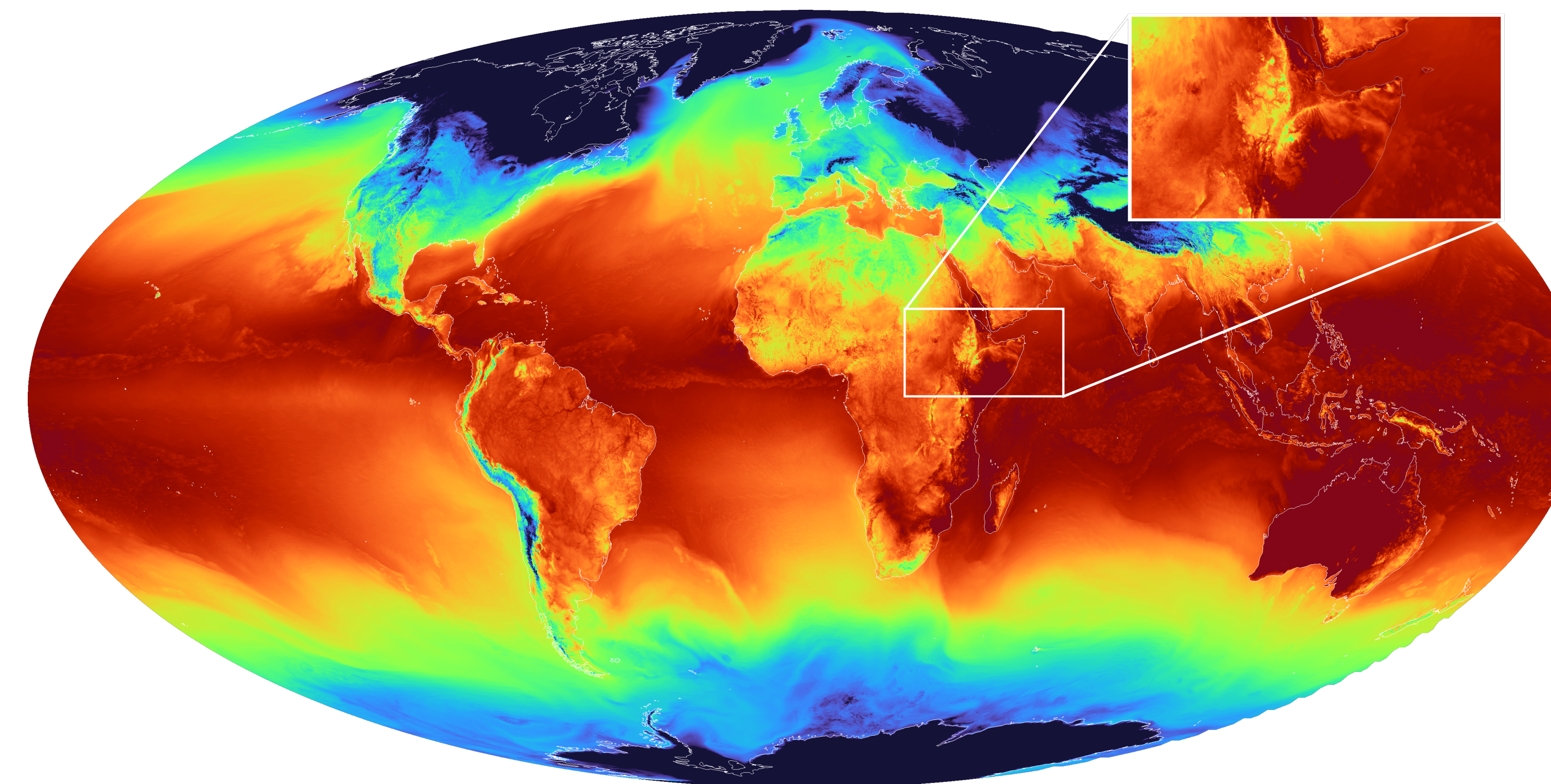
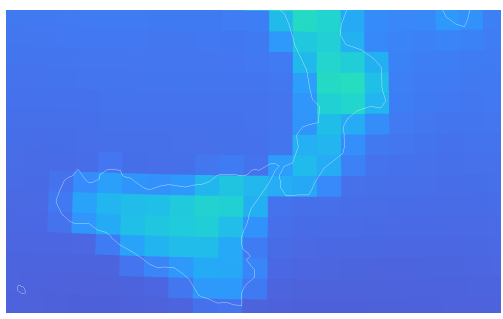

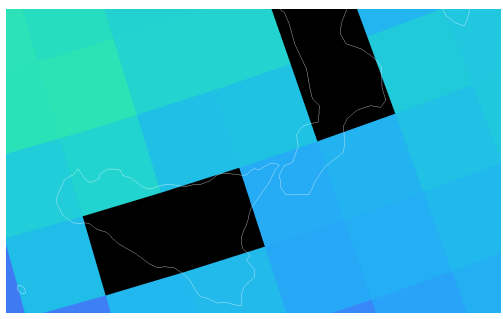

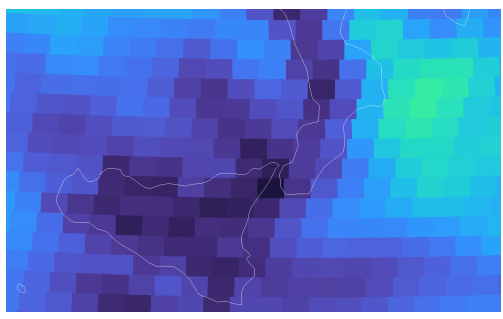

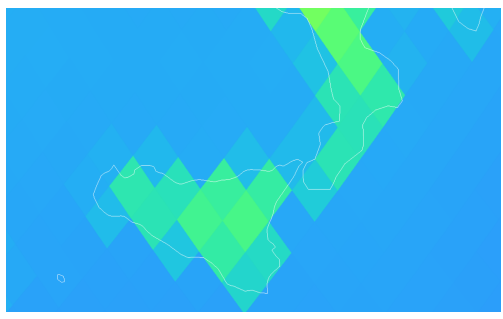

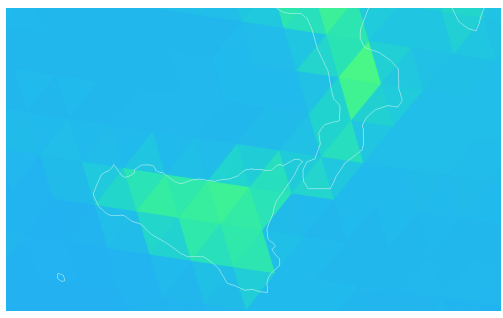

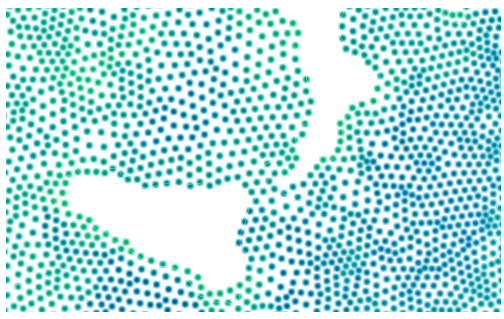



Figure 2. 2.5 km ICON DYAMOND simulation on HEALPix level 11 [1].

Table 1. Grid families currently supported by Gridlook.

Grid family	Typical examples	Preview + QR
Regular + rotated	CMIP, CORDEX	 
Curvilinear	ocean models	 
Gaussian reduced	ERA5, IFS-style products	 
HEALPix	ICON, HEALPix-remapped output	 
Triangular	ICON native output	 
Irregular / unstructured	AWI-CM, FESOM	 

Why Native-Grid Inspection Matters

For data users

Investigate events and processes interactively in unprecedented detail and speed on genuine output.

For model developers

Identify visual anomalies early. Native-grid rendering reveals artifacts, masking problems, and coastline inconsistencies that regriding can obscure.

Results, Limitations, and Outlook

Gridlook demonstrates that interactive visualization of heterogeneous ESM output can be moved much closer to the data distribution layer. A modern browser, standardized metadata, and chunked cloud storage are sufficient for many exploratory tasks that previously required heavier tooling.

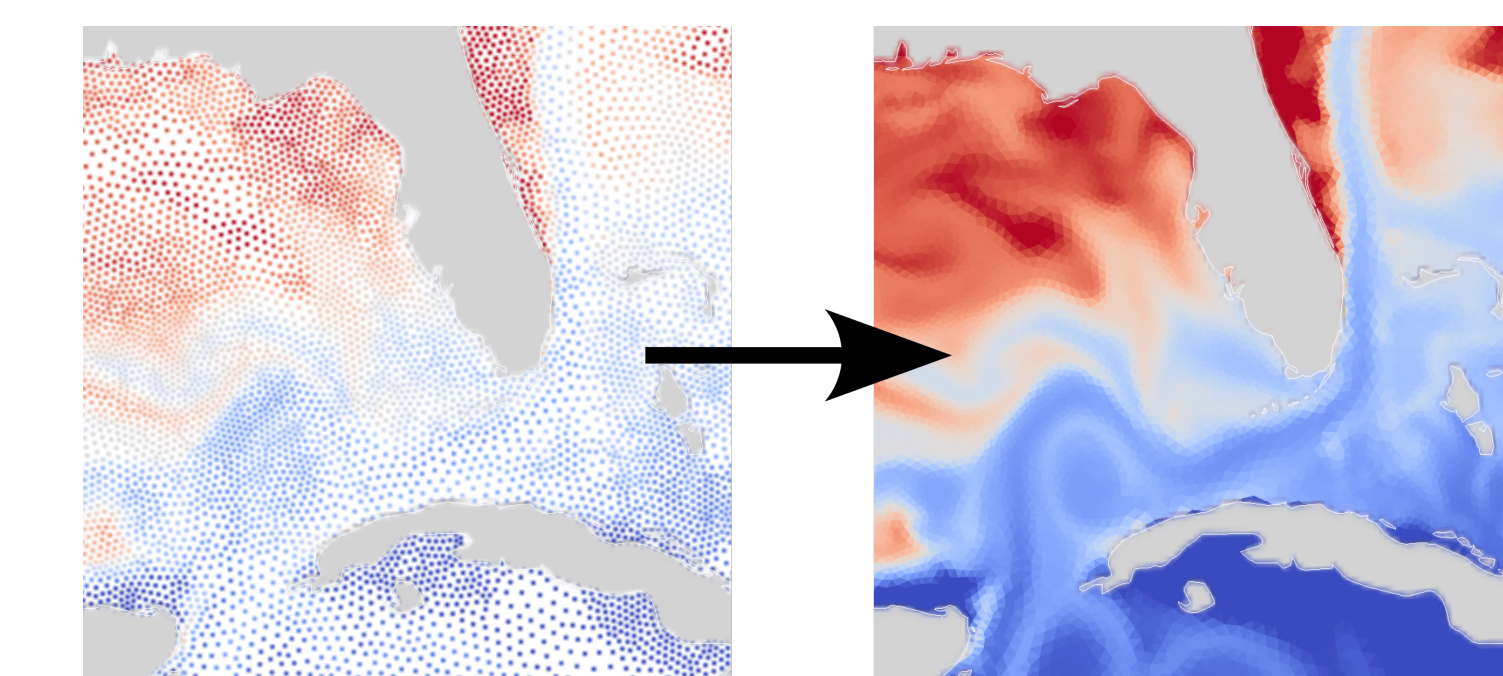


Figure 3. Unstructured FESOM grid detail: point cloud and experimental Delaunay renderer.

- **What works well:** zero-install exploration, support for multiple native grid families, and direct use of cloud-hosted Zarr datasets. Gridlook is already used for data inspection in FREVA [6] and the Kilometer-Scale Cloud at DKRZ [8].
- **Current constraints:** performance depends on chunking, network access, and metadata quality; unstructured meshes remain most demanding.
- **Near-term outlook:** broaden grid support, improve metadata and projection robustness, stabilize the experimental Delaunay triangulation renderer (fig. 3), and add Level of Detail (LoD) techniques for large grids.

The project is open source and available at `gridlook.pages.dev` with source code on GitHub. Community contributions are especially welcome for new grid types, benchmark datasets, and improved interoperability with community metadata conventions.

Meet Us at EGU

Visit the German Climate Computing Center at **Booth #23!**

Wednesday, May 6, 4:00–4:30 p.m. Fantastic Climate Model Data: a visual journey through simulation output on different grids.

Wednesday, May 6, 4:30–5:00 p.m. Interactive Live Visualization purely in the Webbrowser: a hands-on browser-based Zarr session.

References

- [1] Digital earths global hackathon catalog. <https://digital-earths-global-hackathon.github.io/catalog/>, 2024. Accessed 2026-04-08.
- [2] Pangeo: A community for open, reproducible, scalable geoscience. <https://pangeo.io/>, 2024. Accessed 2026-04-23.
- [3] Zarr specification. <https://zarr-specs.readthedocs.io/>, 2024. Accessed 2026-04-08.
- [4] R. P. Abernathy, T. Augspurger, A. Banihirwe, et al. Cloud-native repositories for big scientific data. *Computing in Science & Engineering*, 23(2):26–35, 2021.
- [5] CF Conventions Committee. Cf metadata conventions. <https://cfconventions.org/>, 2024. Accessed 2026-04-23.
- [6] M. Hadizadeh, M. Bergemann, E. Lucio Echeiza, et al. A service-oriented distributed zarr solution for climate data access across heterogeneous hpc and storage infrastructures. *EGU General Assembly 2026*, EGU26-9440, May 2026.
- [7] T. Manz. `zarrita.js`. <https://zarrita.dev/>, 2026. Accessed 2026-04-08.
- [8] F. Wachsmann and C. Wickramage. The kilometer-scale cloud by dkcz. *EGU General Assembly 2026*, EGU26-9485, May 2026.
- [9] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, et al. The fair guiding principles for scientific data management and stewardship. *Scientific Data*, 3(1), Mar 2016.