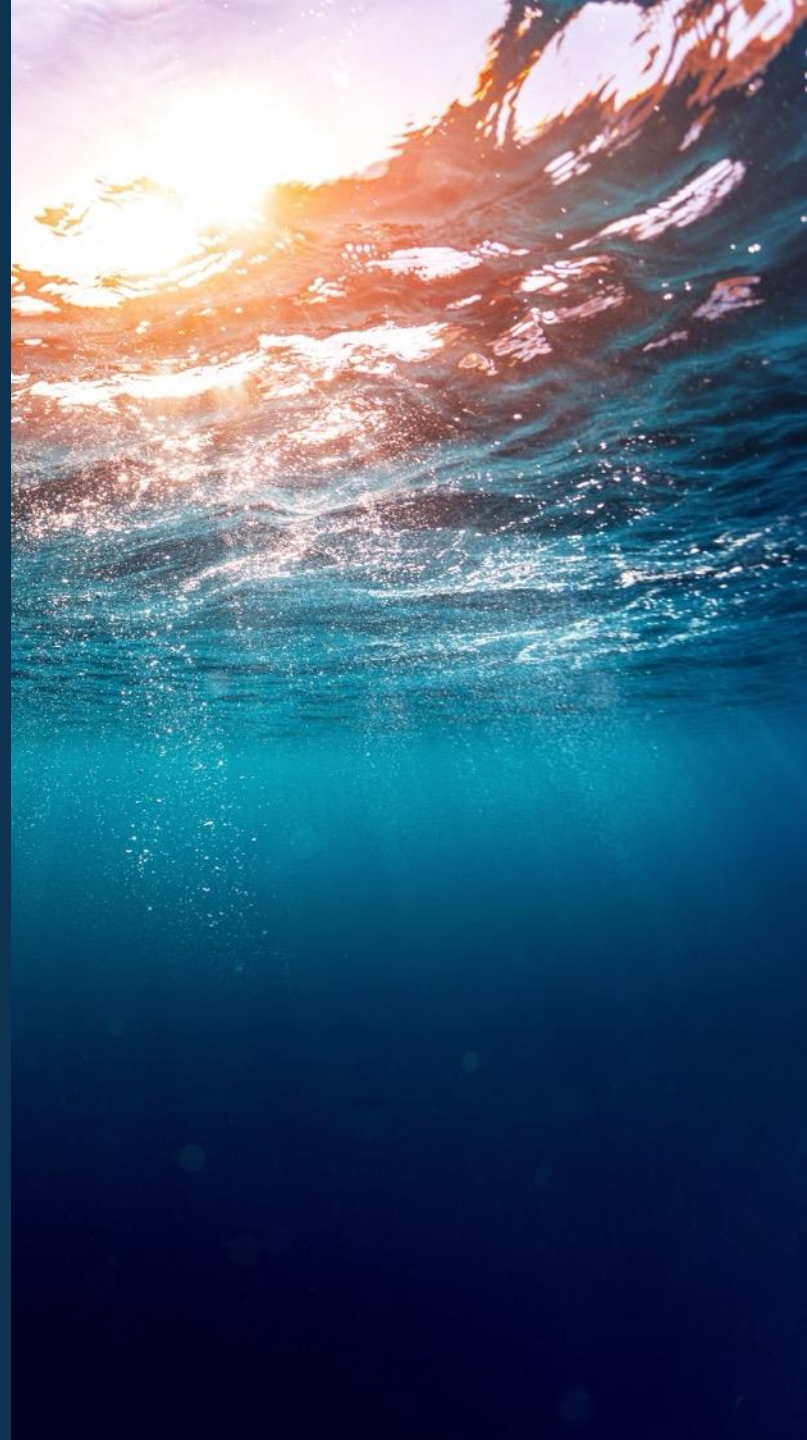
DEVELOPING HIGH FIDELITY VISUALISATIONS FOR MARINE RESEARCH USING UNREAL AND CESIUM FOR UNREAL



National Oceanography Centre

Magda Juarez Olaya, RSE, SEAI group

CHALLENGE

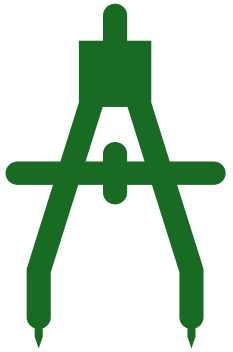


VISIBILITY OF MARINE RESEARCH IN ARCTIC REGION

Make groundbreaking work of NOC visible to partners across science, policy and industry



Translate complex oceanographic data into intuitive, immersive/high-fidelity stories.

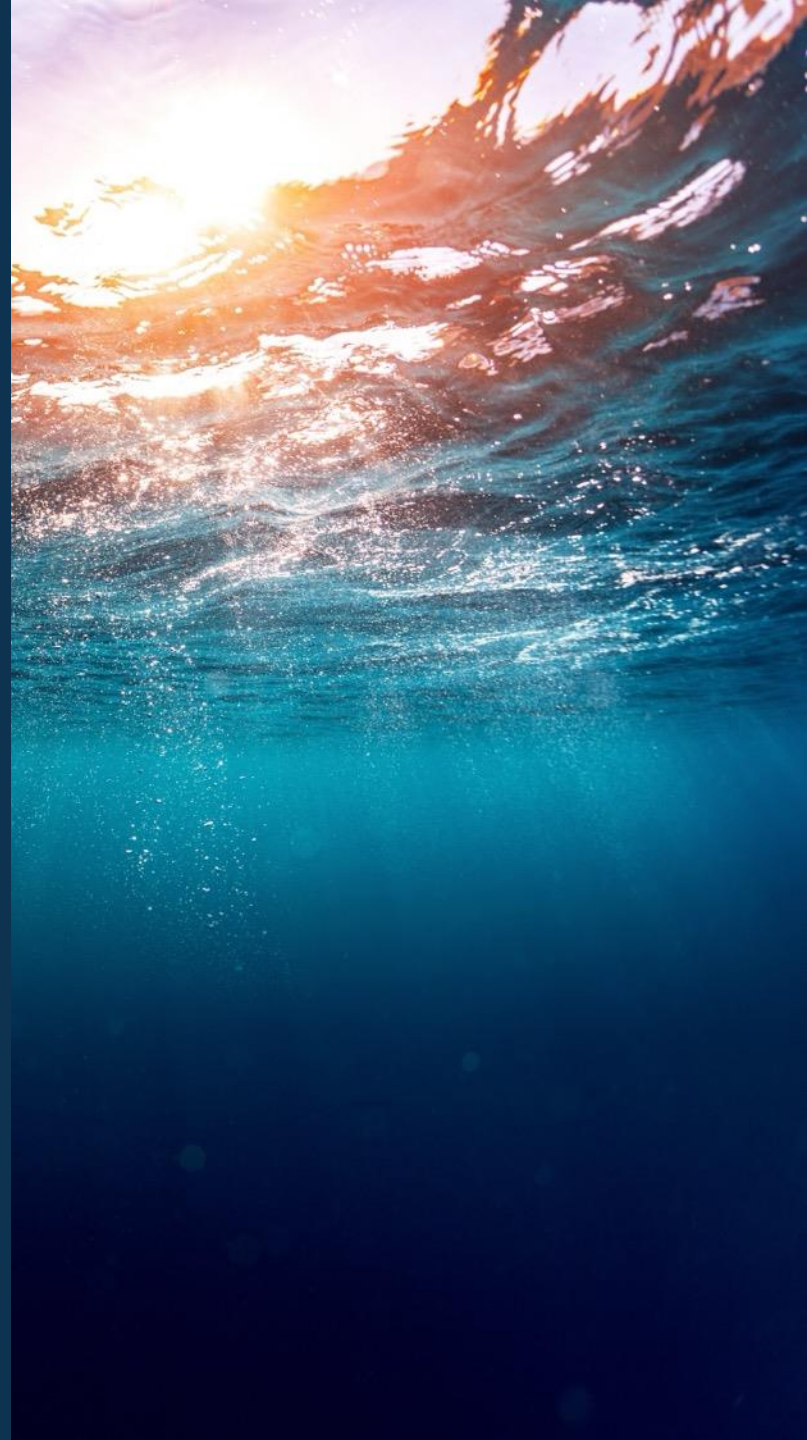


NOC IN THE ARCTIC

Group of people at NOC working on anything related to oceanography in the Arctic.



HIGH FIDELITY VISUALISATIONS WITH UNREAL AND CESIUM FOR UNREAL



UNREAL ENGINE & CESIUM FOR UNREAL

Unreal Engine

A powerful, real-time 3D creation platform developed by Epic Games, used for building video games, film, television and architectural visualisations.



Cesium for Unreal

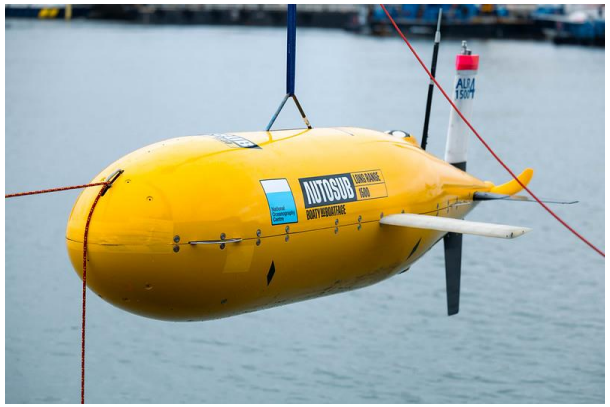
Combines the 3D geospatial capability of Cesium with the high-fidelity rendering power of Unreal Engine.



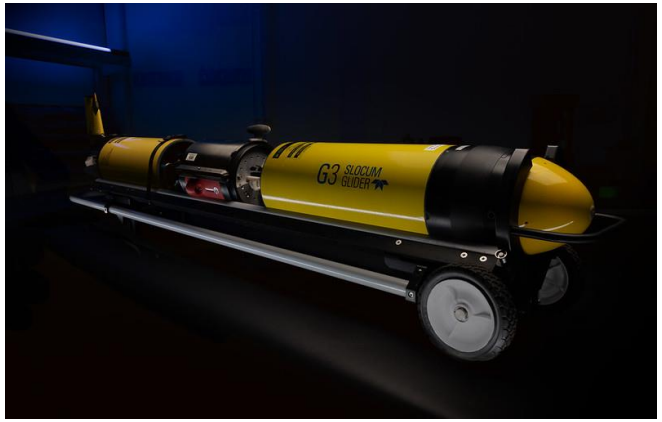


MAMMA-MIA TOOLBOX

Toolbox that simulates the payload (sensors) and trajectory of a platform (ALR, glider) that is sampling the ocean.



Autosub Long Range (ALR)



Glider

ARCHITECTURE – SYSTEM CONTEXT



3D visualization

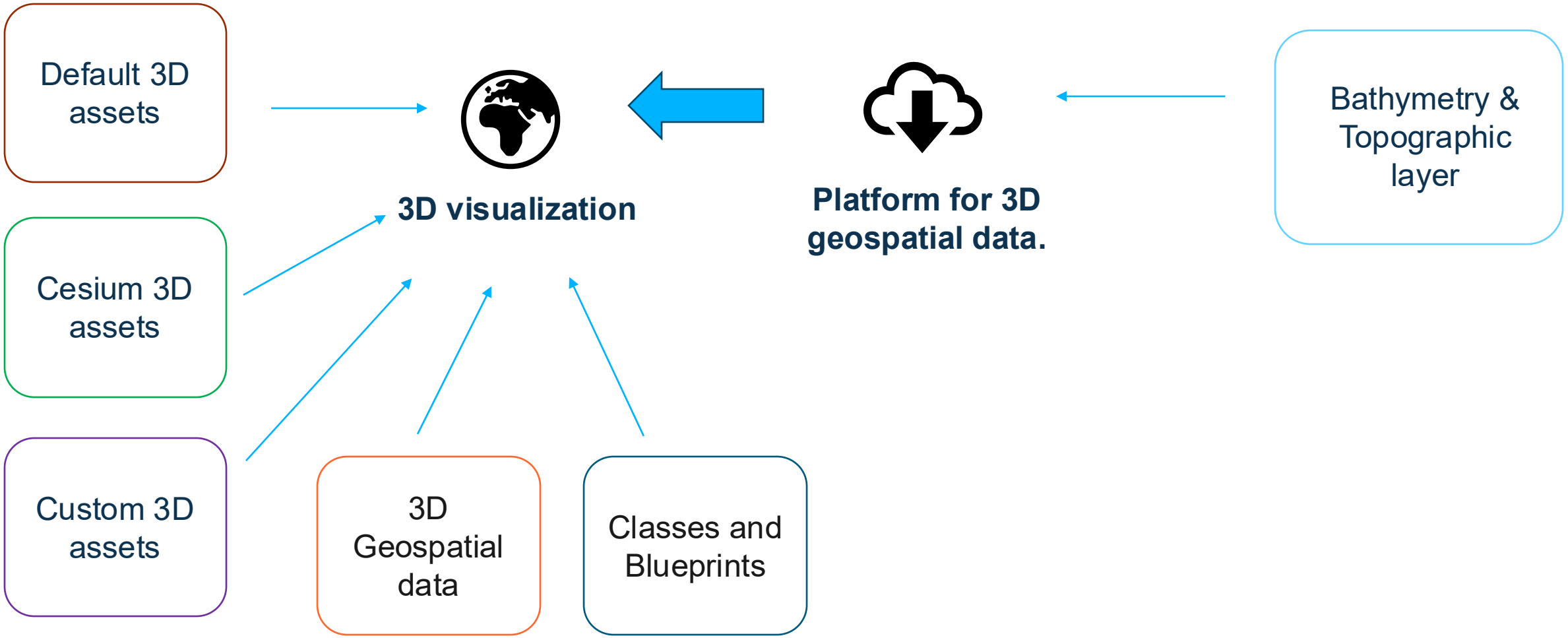




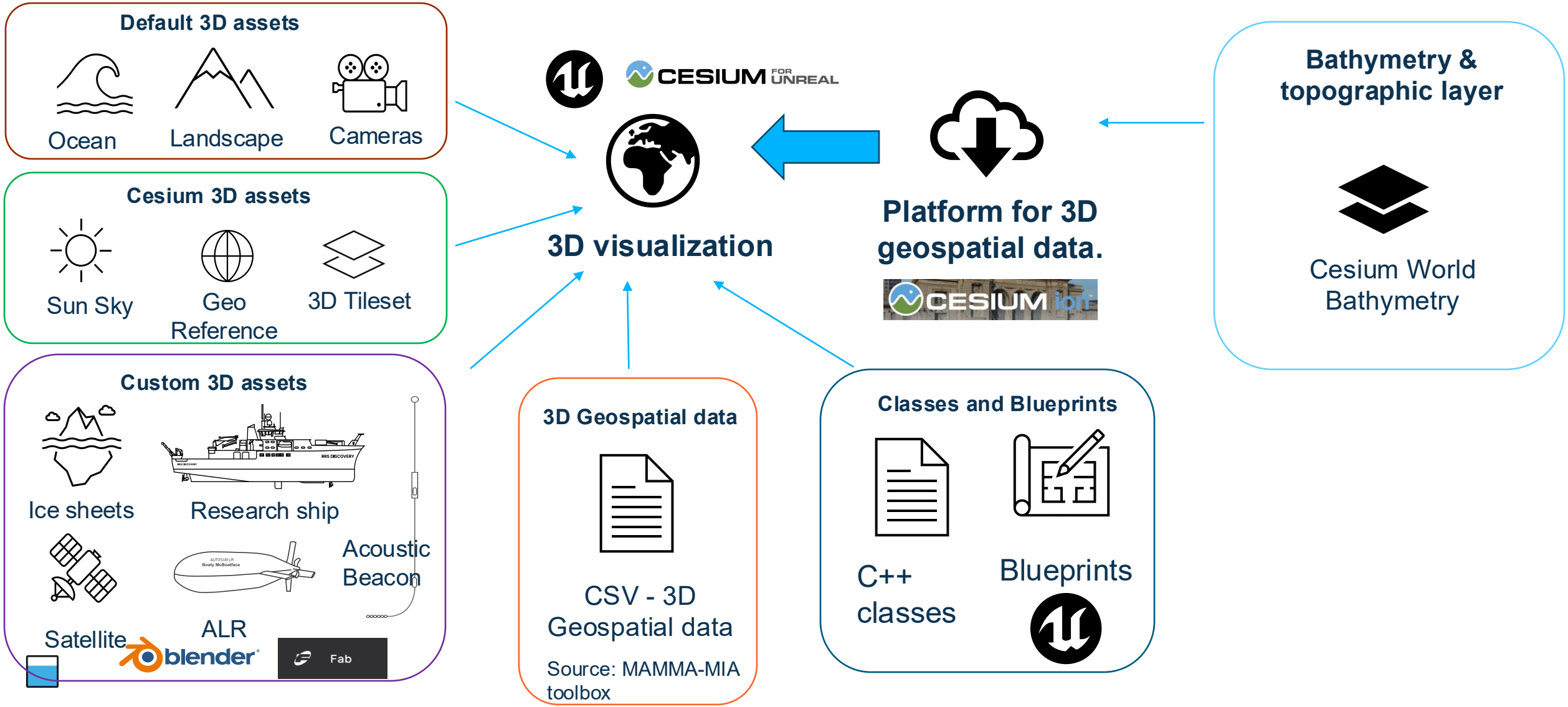
Platform for 3D geospatial data.



ARCHITECTURE – CONTAINER DIAGRAM



ARCHITECTURE – COMPONENT DIAGRAM



CESIUM WORLD BATHYMETRY – SOURCES

Source	Description
GEBCO_2023 Grid	A global terrain model for ocean and land, providing elevation data in meters on a 15 arc-second interval grid.
High-resolution depth model for the Great Barrier Reef - 30 m	A detailed bathymetric map produced by James Cook University, Geoscience Australia, and the Australian Hydrographic Service.
swissBATHY3D	A highly accurate digital elevation model (DEM) produced by swisstopo that maps the topography of Swiss lakebeds
NOAA NCEI 2023 Coastal Relief Models (CRMs)	A high-resolution, 1 arc-second (\(\approx \)30-meter) Digital Elevation Models (DEMs) providing seamless, bare-earth, integrated topographic-bathymetric data for U.S. coastal regions.
Northern GoM Deepwater Bathymetry Grid from 3D Seismic	The BOEM Northern Gulf of Mexico Deepwater Bathymetry Grid is a high-resolution, mosaicked dataset created from over 100 individual 3D seismic surveys, covering more than 90,000 square miles (approx. 230,000 km²) of the deepwater Northern Gulf of Mexico.
NOAA NCEI Continuously Updated Digital Elevation Model	High-resolution, integrated bathymetric-topographic datasets designed for coastal hazard modeling, such as tsunami and storm surge inundation

CESIUM WORLD BATHYMETRY - RESOLUTION

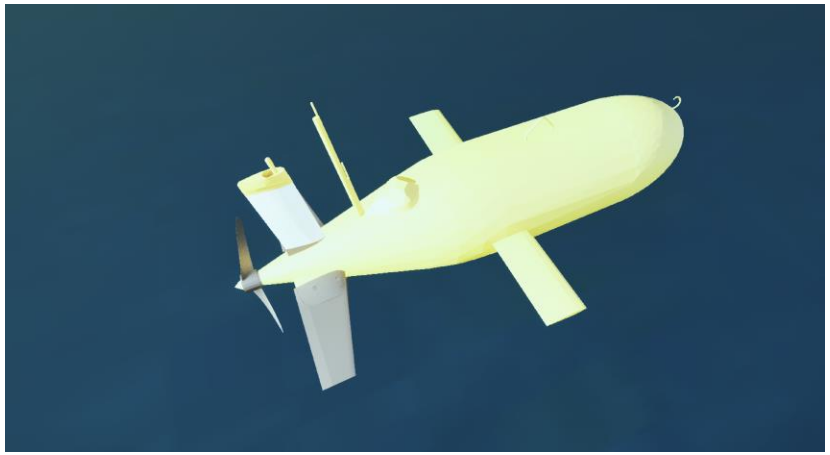
Coverage	Approximate resolution in meters
Swiss Lakes	1 m, 2 m , and 3 m
U.S. East Coast, U.S. islands, locations in AK, OR, CA	3 m and 10 m
Northern Gulf of Mexico	12 m
The Great Barrier Reef	30 m
United States West Coast	90 m
Entire Earth	450 m



3D MODEL ASSETS

Default, Cesium and custom 3D assets









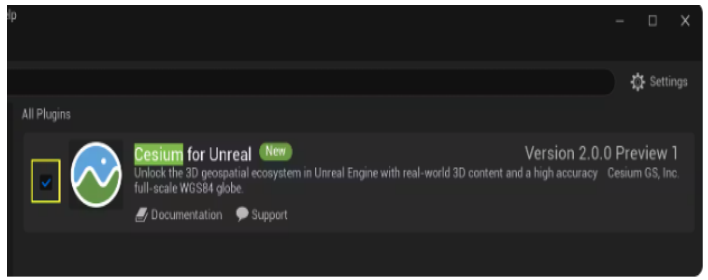
HOW TO DEVELOP A 3D VISUALISATION WITH UNREAL AND CESIUM?

1. Install and enable Cesium for Unreal plugin

2. Add default and Cesium 3D assets

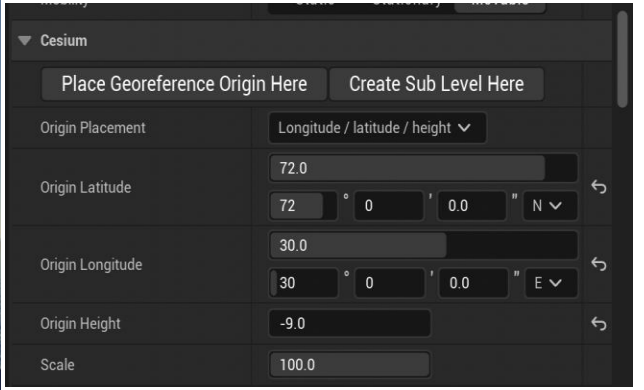
3. Config initial settings

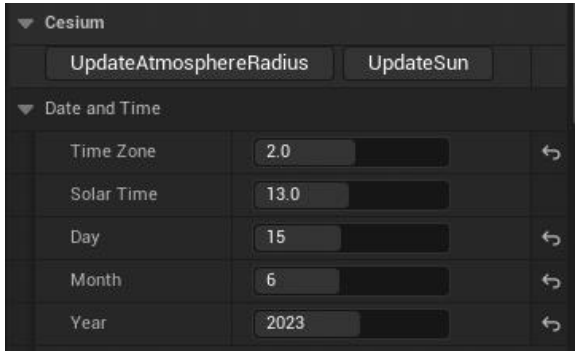
4. Import simulated 3D geospatial data












```
id,longitude,latitude,depth
1,30.0,72.0,13.483123779296875
2,30.0,72.00104094378904,13.482182600033276
3,30.0,72.00208188757807,13.481241420769674
4,30.0,72.00312283136711,13.480300241506075
5,30.0,72.00416377515614,13.479359062242473
6,30.0,72.00520471894518,13.47841788297887
7,30.0,72.00624566273422,13.477476703715297
8,30.0,72.00728660652325,13.47653552445167
9,30.0,72.00832755031229,13.475594345188071
10,30.0,72.00936849410132,13.474653165924472
11,30.0,72.01040943789036,13.47371198666087
12,30.0,72.0114503816794,13.472770807397268
```

HOW TO DEVELOP A 3D VISUALISATION WITH UNREAL AND CESIUM?

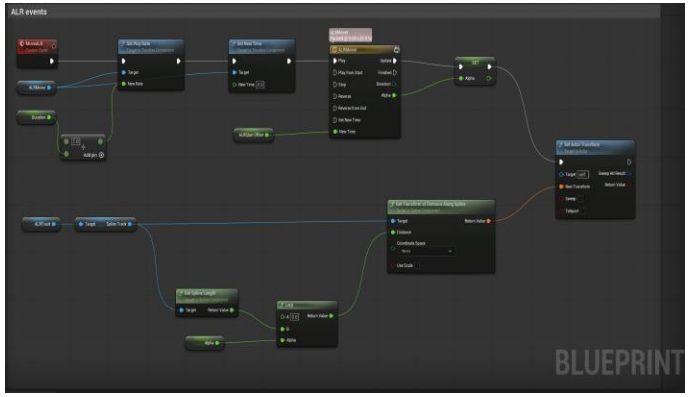
5. Create a diving path object



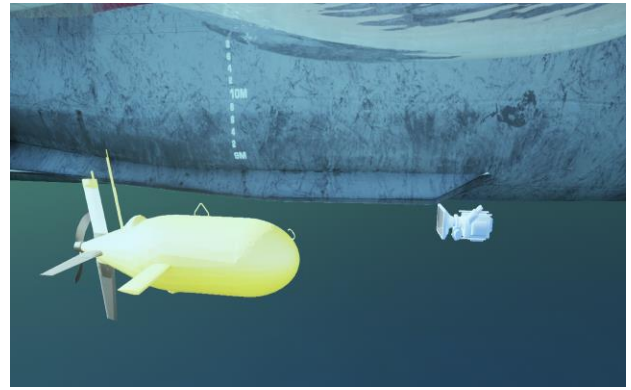
6. Add custom 3D assets



7. Add events to ALR model



8. Attach cameras to ALR model



DEMO



REFERENCES

- MAMMA-MIA toolbox: https://github.com/NOC-MDP/MammaMia
- Cesium for Unreal documentation: https://cesium.com/learn/unreal/
- Cesium World Bathymetry: https://cesium.com/platform/cesium-ion/content/cesium-world-bathymetry/
- The GEBCO_2023 Grid https://www.gebco.net/data-products/gridded-bathymetry-data/gebco2023-grid
- High-resolution depth model for the Great Barrier Reef - 30 m: https://amsis-geoscience-au.hub.arcgis.com/maps/9d9768be471e4990a3f41f669e6427a6
- swissBATHY3D: https://www.swisstopo.admin.ch/en/height-model-swissbathy3d
- NOAA NCEI 2023 Coastal Relief Models (CRMs): https://data.noaa.gov/metaview/page?xml=NOAA/NESDIS/NGDC/MGG/DEM/iso/xml/999901.xml&view=getDataView
- Northern GoM Deepwater Bathymetry Grid from 3D Seismic: https://www.boem.gov/oil-gas-energy/mapping-and-data/map-gallery/northern-goa-deepwater-bathymetry-grid-3d-seismic
- NOAA NCEI Continuously Updated Digital Elevation Model: https://www.ncei.noaa.gov/products/coastal-elevation-models

THANK YOU



National Oceanography Centre

Enhancing legacy Fortran for Modern Systems

Dr. Mark Richardson, Technical Head of
Centre for Environmental Modelling and Computation

What is CEMAC?

UNIVERSITY OF LEEDS

- Software Development Scientists
- Embedded RSE group
  - Often seen as a pool of post-doctoral researchers
  - Sometimes seen as professional services or technical support
- Research Grant Funded
  - Academics add us after discussion
  - Fractions of time
- SDS work on multiple projects
  - Broad skills: Python, Fortran, WebUI, data transformation, Linux build systems, HPC jobs
- Domain knowledge
  - Familiarity with researchers in the School
  - Experienced with the tools being used to deliver the research
  - Various levels of knowledge of the science topics

The NG-ARCH project

UNIVERSITY OF LEEDS

UK Met Office next generation modelling system Momentum® needs to be ready for next generation computer systems.

- 7 research institutions
  - 5xMOAP> Bristol, Edinburgh, Exeter, Leeds, Reading
  - Cambridge [due to UKCA chemistry model]
  - STFC as owner and developer of PSyclone.

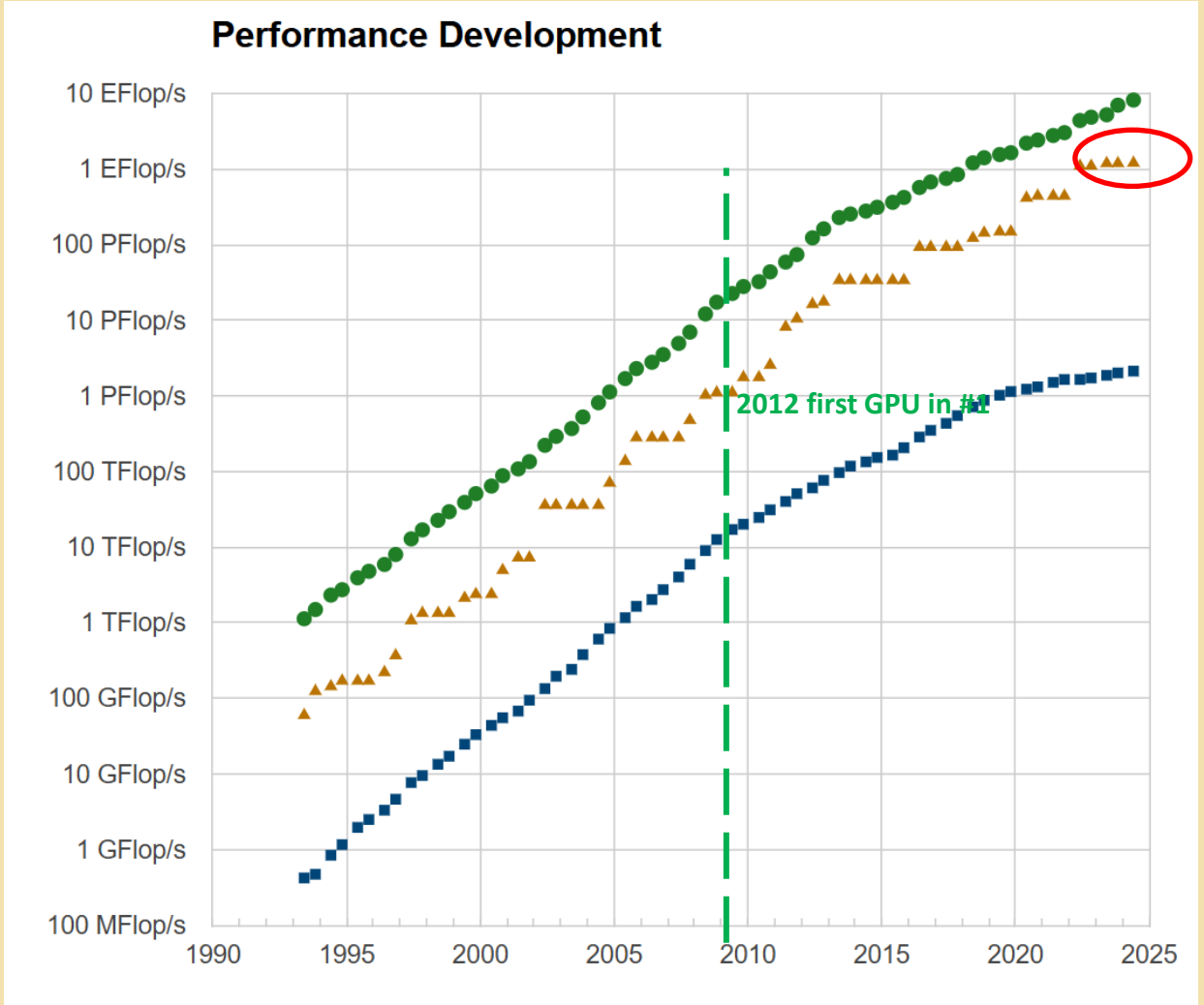
- Target GPU off load with automatic code generation

Motivation: Trends in HPC

UNIVERSITY OF LEEDS

- Next HPC systems will have more cores per node
  - CPU and/or GPU
- Over time – the fastest computer in the world has been able to achieve:
  - 1995 - 100 gigaflops ($O(10^9)$)
  - 2005 - 100 teraflops ($O(10^{12})$)
  - 2015 - 34 petaflops ($O(10^{15})$)
  - **2025 - 1.2 exaflops - ($O(10^{18})$)**
- Red Ellipse: Only Top 4 > Exaflop
  - [Nov 2025]

Green circles: accumulated
Orange triangles: number 1 spot
Blue squares: number 500 on list

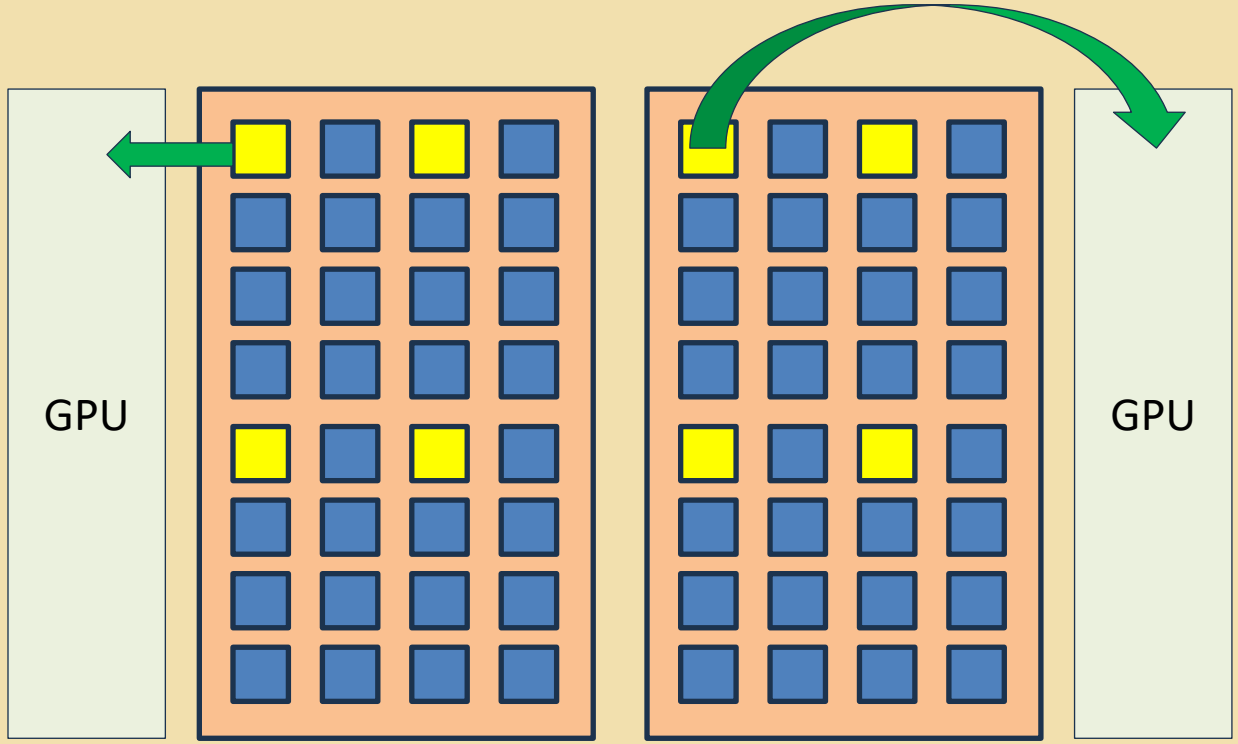


Source - https://top500.org/statistics/perfdevel/

What is needed to make best use of HPC resource

UNIVERSITY OF LEEDS

- Making a distributed memory parallel code capable of using GPUs
- Choices
  - CUDA,
  - OpenCL,
  - OpenACC,
  - OpenMP
- Chose OpenMP
- Existing MPI job configuration
- SLURM batch options to spread it out more memory per core



Case study: UKMO Momentum®

UNIVERSITY OF LEEDS

Unified Model aligns layers of Longitude-Latitude Surfaces



- Weather and climate simulations
- Two pathways
  - Modern development. Core dynamics different and build infrastructure core and apps
  - Legacy Fortran, included into the new build system
- Build system
  - Typical scientist role will develop and use the “apps” part not see a lot of the rest
- Cubed-sphere
  - 6 panels
  - improves numerical behaviour
  - Avoid tight grid spacing at poles (“singularity”)

Cube-Sphere has 6 panels; avoiding pole-singularity improves numerical time step control



https://www.metoffice.gov.uk/research/approach/modelling-systems/momentum
https://www.metoffice.gov.uk/research/approach/modelling-systems/unified-model

Momentum build system

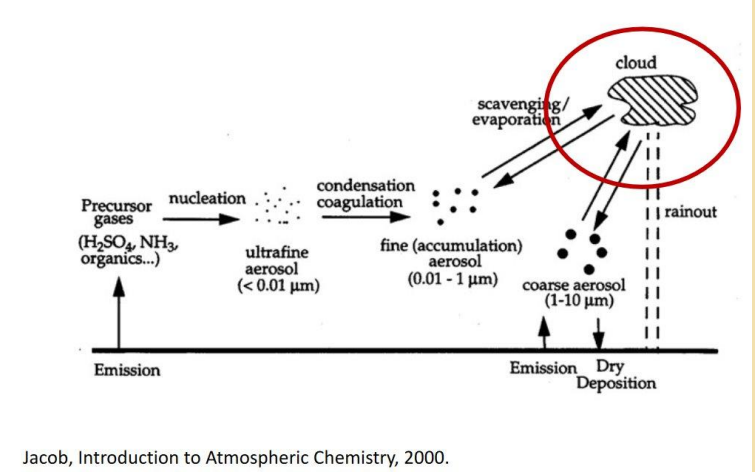
UNIVERSITY OF LEEDS

- Lfric_core repository provides infrastructure for build
- Lfric_apps repository provides the base for science
- One of the apps is “lfric_atm”
  - Brings together the atmospheric science components
- Science has been continuously developed since 1990 within “unified Model”
  - There are a lot of person years of investment
  - Writing (replicating) from scratch a huge task
- Automate the process to occur during the build
- Extra step of pre-processing ahead of compilation

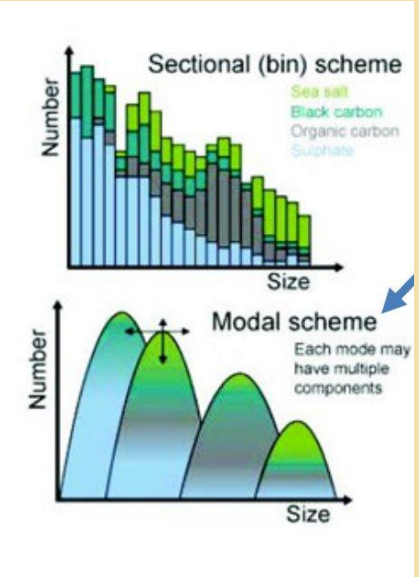
UKCA – GloMAP mode Aerosols

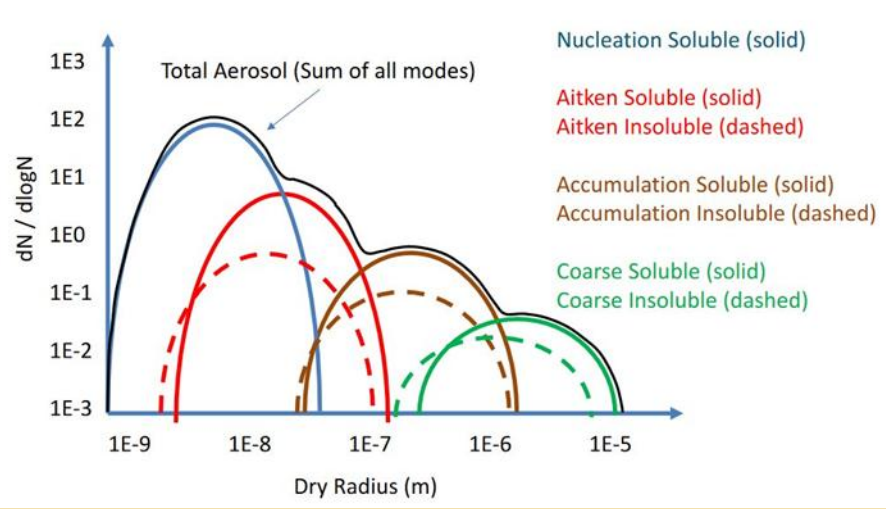
UNIVERSITY OF LEEDS

- Leeds work is focused on UKCA **GLOMAP** (Global Model of Aerosol Processes) acceleration using PSyclone – started in September 2024.
- Modelling aerosol size distribution (via *modes*), mass, radius, chemical composition – important for their effects on clouds and radiation. These are subject to change due to lots of microphysical processes –these make up much of the GLOMAP code.
- Many operations that take place in a independent grid boxes (i.e. no transport between them) – *ideal for GPU parallelisation* (lots of loops over the model domain, and collapsible loops over the model domain/different modes)



Jacob, Introduction to Atmospheric Chemistry, 2000.





Images from
https://www.ukca.ac.uk/images/c/c8/UKCA_Training_Cambridge_MODE.pdf

Enable Parallel Threads with PSyclone

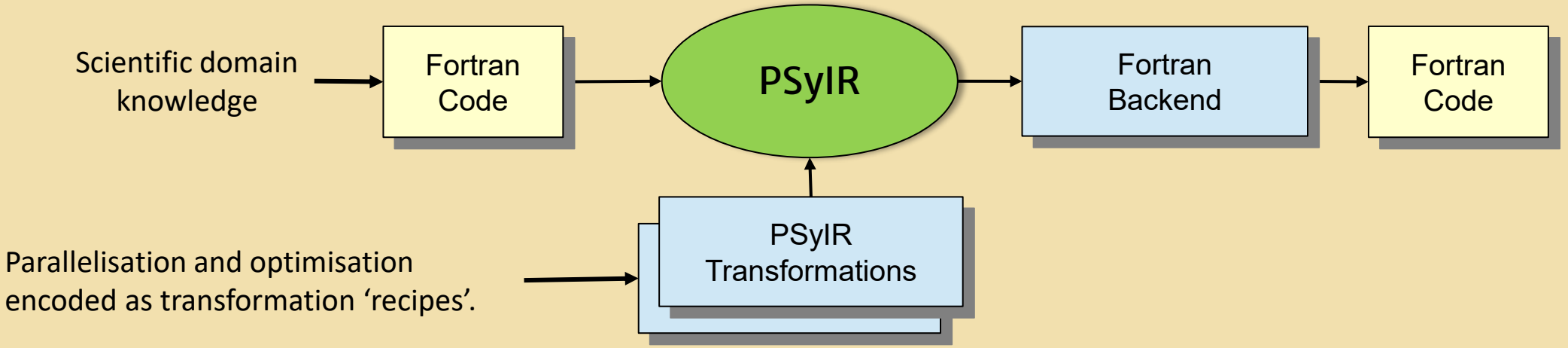
UNIVERSITY OF LEEDS

Science and Technology Facilities Council
Hartree Centre

- Third party tool being developed at STFC
  - NEMO ocean model organised by columns
- Expected approach is for blank sheet development
- Secondary function is to process legacy code
- Requires a specialist to “teach” or “inform” tool about the existing code and target computer system
- **Transformer** to convert pure serial domain specific code into parallel aware code ahead of compilation.
- **Transmute** processes legacy Fortran in a similar manner.
- Excellent response and interaction from that team.
- https://psyclone.readthedocs.io/en/stable/

PSyclone

As a source-to-source compiler:

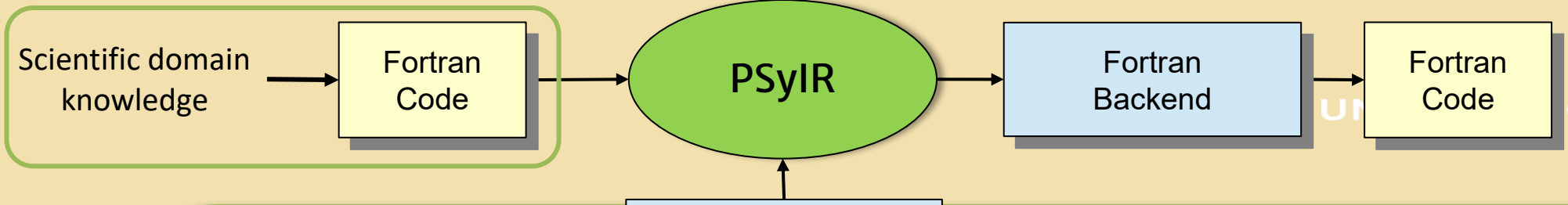


Separation of concerns

Perf portability: diff. recipes for each architecture

Visible "*readable*" output
Standard debugging/profiling

Standard output composable w. other s2s, vendor compilers



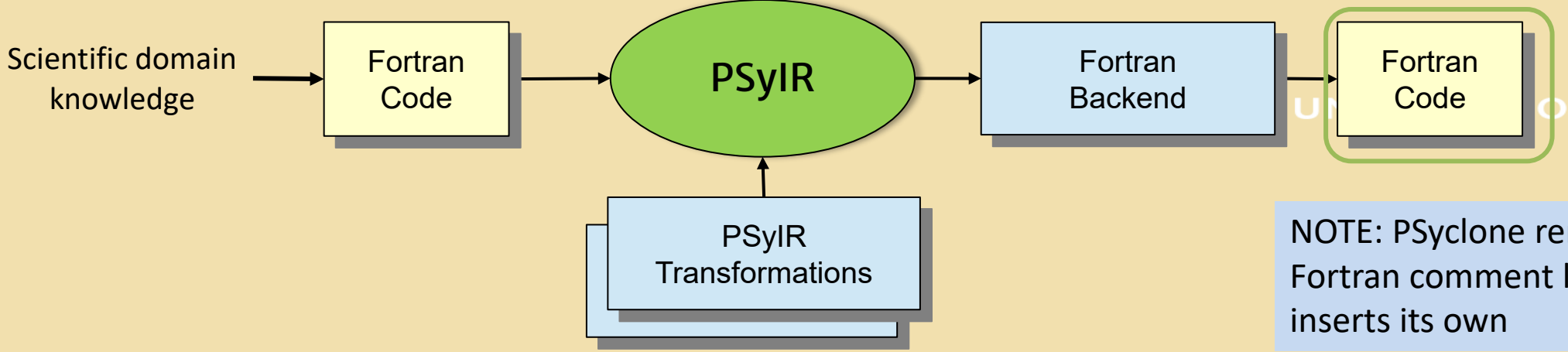
Original code:

```
DO jn = 1, kjpt                                                    ! ==========
   !                                                               ! tracer loop
   !                                                               ! ==========
   !!----------------------------------------------------------
   !!   I - masked horizontal derivative
   !!----------------------------------------------------------------------
: bug.... why (x,:,:)?   (1,jpj,:) and (jpi,1,:) should be sufficient....
   zdit (1,:,:) = 0._wp     ;     zdit (jpi,:,:) = 0._wp
   zdjt (1,:,:) = 0._wp     ;     zdjt (jpi,:,:) = 0._wp
   !!end

   ! Horizontal tracer gradient
   DO jk = 1, jpkm1
      DO jj = 1, jpjm1
         DO ji = 1, jpim1   ! vector opt.
            zdit(ji,jj,jk) = ( ptb(ji+1,jj  ,jk,jn) - ptb(ji,jj,jk,jn) ) * umask(ji,jj,jk)
            zdjt(ji,jj,jk) = ( ptb(ji  ,jj+1,jk,jn) - ptb(ji,jj,jk,jn) ) * vmask(ji,jj,jk)
         END DO
      END DO
   END DO
   IF( ln_zps ) THEN      ! botton and surface ocean correction of the horizontal gradient
      DO jj = 1, jpjm1                ! bottom correction (partial bottom cell)
```

Note comments

Identifies DO loop for parallel



NOTE: PSyclone removes Fortran comment lines and inserts its own

Generated Fortran with OpenMP directives added, those are for CPU threads

```
jn = 1, kjpt
zdit(1, :, :) = 0._wp
zdit(jpi, :, :) = 0._wp
zdjt(1, :, :) = 0._wp
zdjt(jpi, :, :) = 0._wp
!$OMP parallel do default(shared), private(ji,jj,jk), schedule(static)
DO jk = 1, jpkm1
  DO jj = 1, jpjm1
    DO ji = 1, fs_jpim1
      zdit(ji, jj, jk) = (ptb(ji + 1, jj, jk, jn) - ptb(ji, jj, jk, jn)) &
           * umask(ji, jj, jk)
      zdjt(ji, jj, jk) = (ptb(ji, jj + 1, jk, jn) - ptb(ji, jj, jk, jn)) &
           * vmask(ji, jj, jk)
    END DO
  END DO
END DO
!$OMP end parallel do
```

Removed comments

GPU target

UNIVERSITY OF LEEDS

- Several approaches possible
  - CUDA
  - OpenACC
  - OpenCL
  - OpenMP
- Bottom-up
  - Find the low-level “long” do loops and apply threading to those
- Top-down
  - Choose the highest-level subroutine that could be off-loaded to the accelerator
- Compiler independence
  - Choose OpenMP over “OpenACC” (NVIDIA), OpenCL (AMD), CUDA…
- Vendor collaboration
  - Had very good collaboration with NVIDIA (nvhpc compiler) in early stage of project
  - Their vision was for each MPI task to use one GPU and offload huge sections of code

Platforms available

UNIVERSITY OF LEEDS

- Per University systems
  - Software stack variety
  - Use of “Spack”
- Existing system Archer2
  - Reasonable for development but not cutting-edge GPU
  - Software versions unable to build the software
  - Challenge of getting a working test case ported from inside UKMO
- Challenging collaborations.
  - 7 other organisations
- Funder provided a MS Azure Sandbox system and 2x(software stack)
  - NVIDIA compilers nvhpc
  - Gnu GCC

Challenges

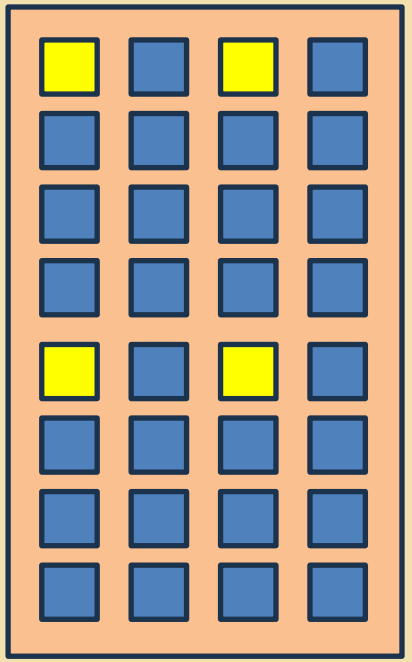
UNIVERSITY OF LEEDS

- Computer systems
  - Initially per institute choice but outside Met Office so some complications
  - Archer2 could attract NCAS CMS support, i.e using PUMA2, had some GPU
  - Later were provided with a **MS Azure** “sandbox” with some contemporary hardware
  - A ***singularity*** image file was created (now known as “***apptainer***”)
- Repositories
  - At outset were using FCM, Rose and cylc
  - Ended just as code transitioned to git.
- Compiler versions
  - Cray compiler on Archer2 not capable of building LFRic, not modern enough
  - UKMO systems team favour “*spack*” and will share “*recipes*” with collaborations.
- Valid simulation cases availability on different systems
  - Data sets not easily supplied
- Dependency software libraries
  - These must be built with same versions of compilers.

Change of Goal to CPU target

UNIVERSITY OF LEEDS

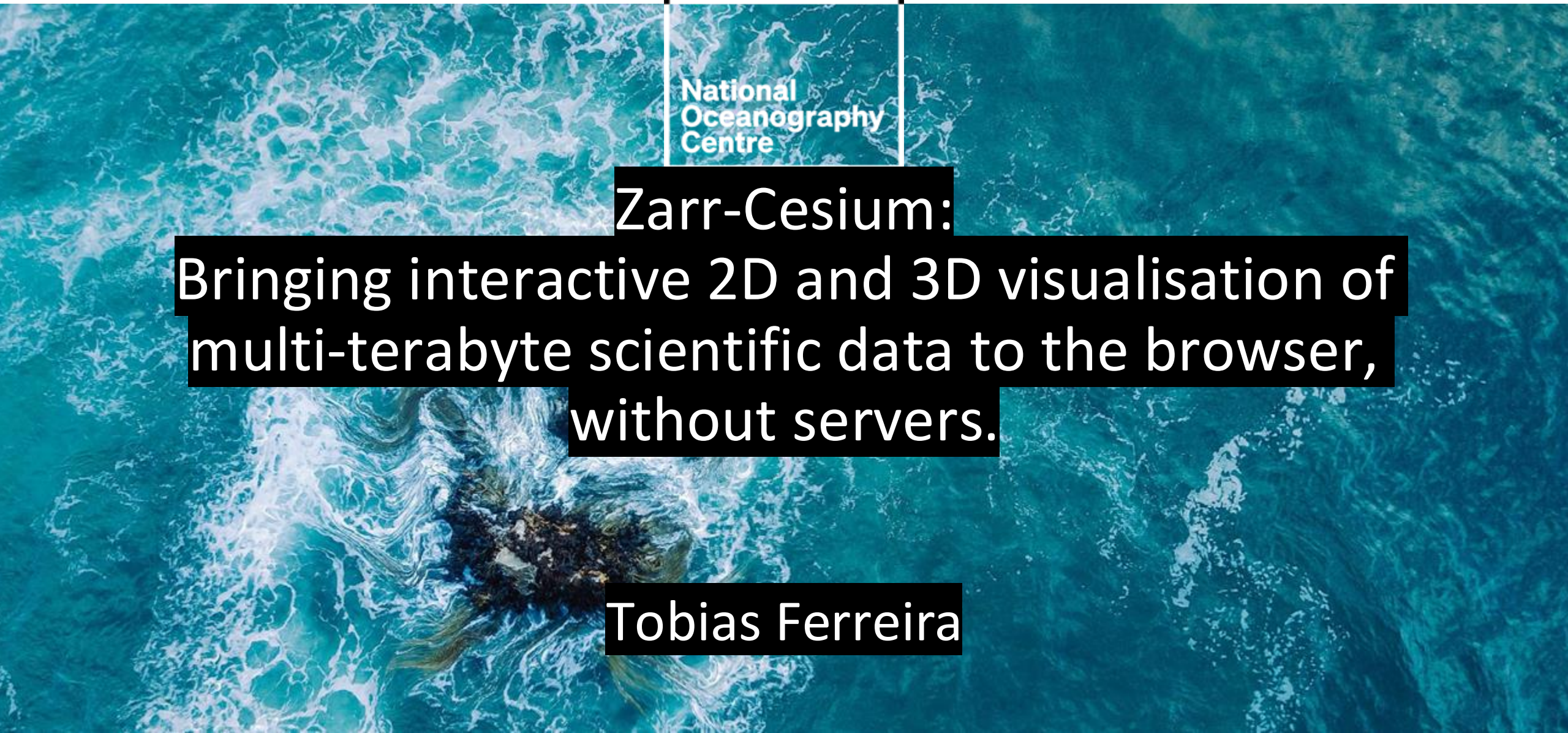
- In March 2025 (12 months in)
  - Code **works** on GPU with some issues to be resolved (slower than unmodified)
  - Timing is poor due to transfer of lots of data from CPU to device
  - Lots of disconnect low level regions initiate the parallelism
  - Potential of most recent hardware to avoid that but at time of development unavailable.
- Project owner identified unlikely to choose a GPU system
  - Revise goal to be CPU parallel
  - Still use OpenMP to work with threads
  - Typically, the distributed compute model will use fewer cores per node as increased memory needed
- Parallelism has been applied at a higher level for CPU target
  - Delivered for UKCA end of Sept 2025
  - Similar to work completed in 2016 with UM
  - Performance matches purely MPI configuration for fixed resource



Summary

UNIVERSITY OF LEEDS

- Complex system of software, inherited legacy functionality
- Tool to automate transformations and enable acceleration by multiple threads either GPU or CPU.
- Skills required include
  - excellent Python to inform tool of nature of Fortran and differing strategies for each of those targets
  - Good Fortran to be sure modifications are correct
  - Linux skills for building software to understand the process and software stack
- Domain familiarity for conversation with software's owner
- Good working practices for versioning and revision control



Zarr-Cesium:
Bringing interactive 2D and 3D visualisation of multi-terabyte scientific data to the browser, without servers.

Tobias Ferreira

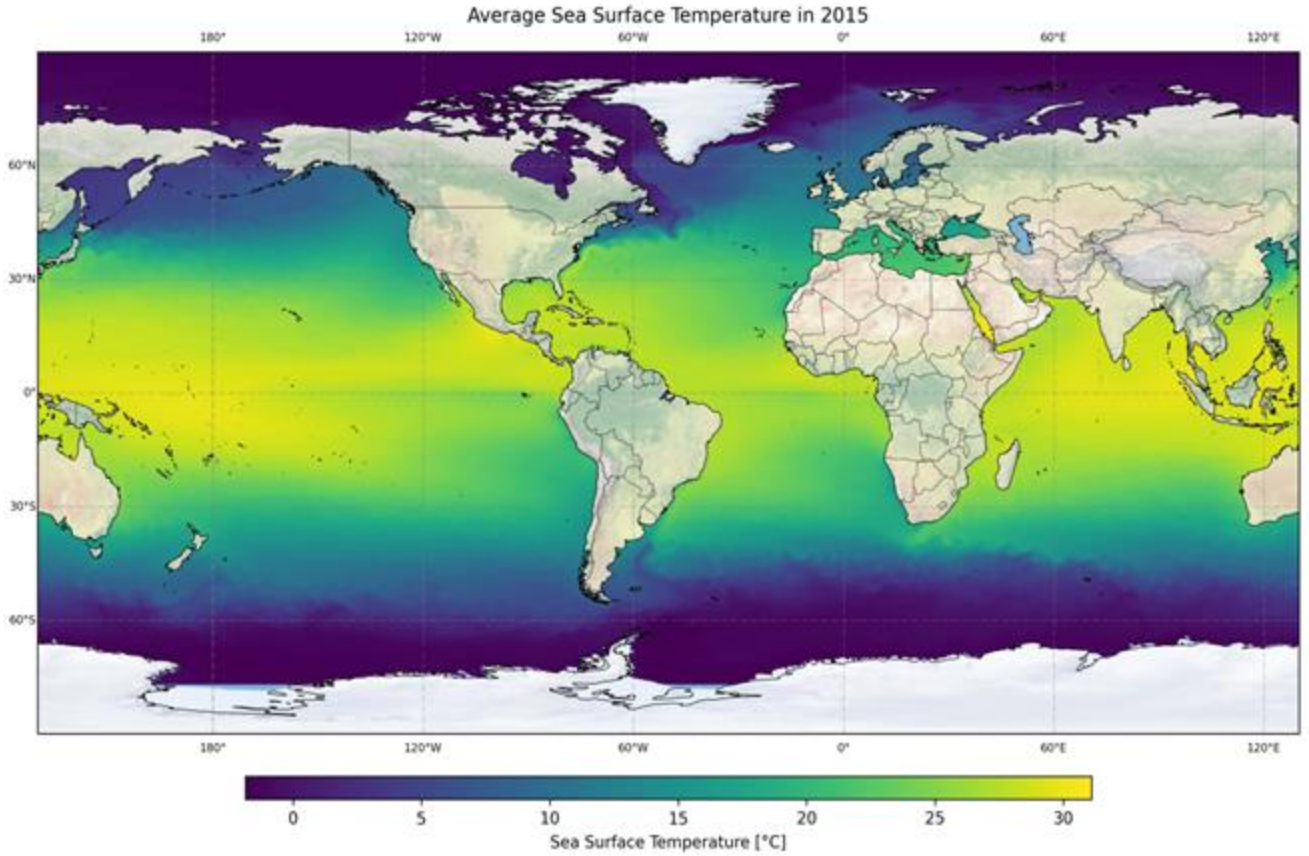
Introduction

- We are building a visualisation tool for the AtlantiS project
- Data from different sources and formats:
  - fixed point observations
  - numerical models
  - gridded datasets
  - autonomous platforms
  - ...

AtlantiS
ATLANTIC CLIMATE AND ENVIRONMENT IMPACTS

WP2: Model Outputs

- NEMO Near-Present-Day simulations developed by the NOC
- Global ocean variables: different resolutions, temporal dimensions
- Available in Icechunk (Zarr) Format
- Store in JASMIN Object Store
- Dataset has more than 70 TB

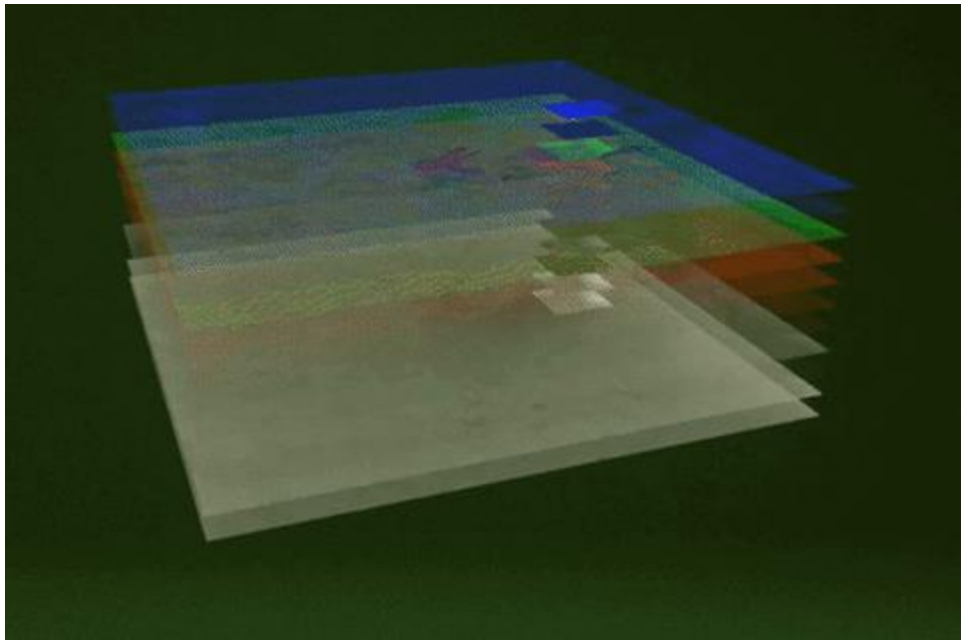


How can we visualise this data in a scalable and high-performance way?

Zarr visualisation

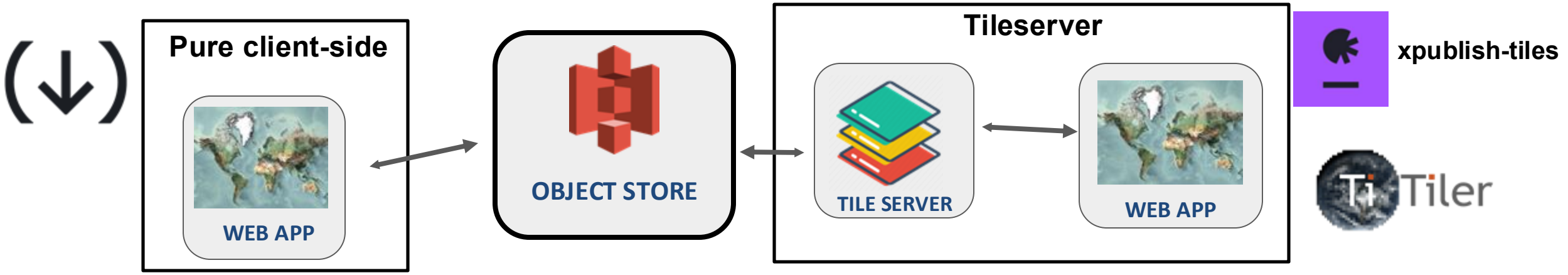
National Oceanography Centre

- Zarr: great for scalable storage & analysis
- Challenging for browser visualisation: users expect fast, interactive tools
- Chunking, metadata, dimensionality and projection bring complexity



USGS animation

Zarr visualisation



They are great solution, but:

- 2D only, but the ocean is 3D
- Data is flattened into slices
- We are not exploring the data as it exists

What's missing?
An open, browser-native, 3D-first way to visualise Zarr datasets

Zarr-Cesium:

a client-side solution for interactive 2D and 3D visualisation of geospatial Zarr datasets in CesiumJS

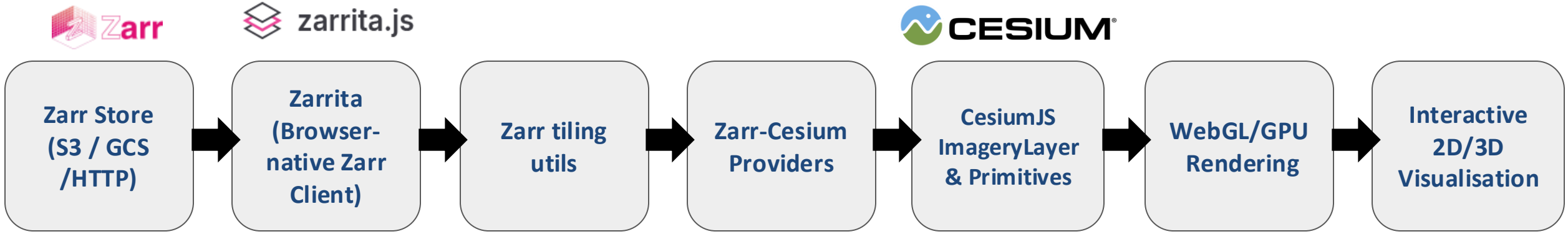
Pure client-side: No preprocessing. No tile servers. No custom infrastructure.

You only need a browser and a Zarr dataset in the Object store

- Supports multi-tera/petabyte Zarr datasets
- Generate 2D raster maps, 3D volumetric cubes and animated U/V vector fields
- WebGL + GPU acceleration
- Dynamic slicing and custom styling



Architecture (High-Level)



Fetches only needed chunks

Works with multiscale datasets

No pre-processing is necessary (it affects performance).

Almost everything happens on the GPU

How to make Zarr “Renderable”?

Making Zarr “Renderable”

National Oceanography Centre

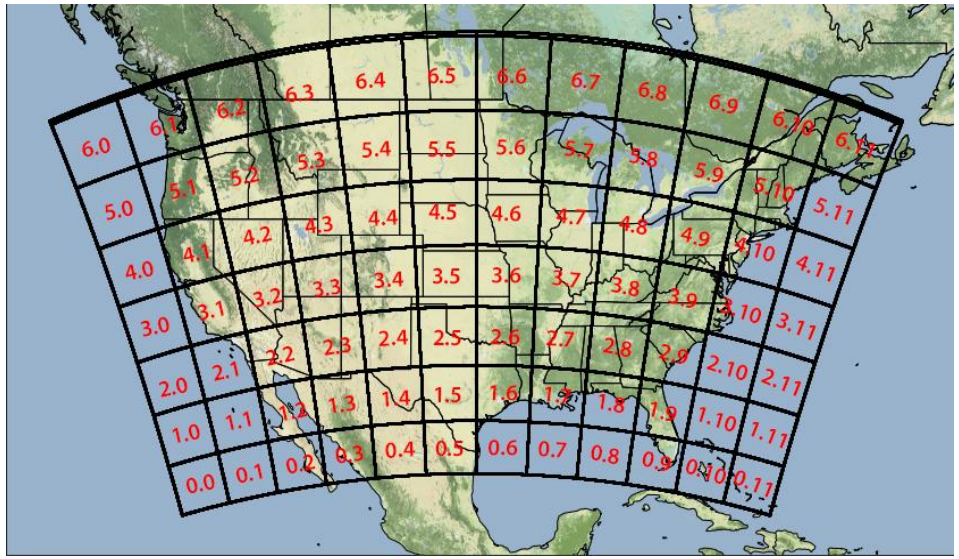
Zarrita.js: enables direct, client-side access to Zarr datasets from a web browser. zarrita.js

Zarr-cesium: Translate scientific Zarr datasets into renderable structures

- CF-compliant dimension detection (including CF time)
- CRS detection (EPSG:4326 / 3857)
- Multiscale pyramid discover

What happens per frame / tile?

- Cesium camera → geographic bounds
- Bounds → pixel-space indices
- Indices → Zarr slice objects → Browser memory



https://home.chpc.utah.edu/~u1269218/HrrrZarr/

Magic Sauce: GPU Rendering in the Browser

National Oceanography Centre

- Terabyte-scale numeric arrays
- Continuous interaction (pan, zoom, slice, recolor)

Browser CPU is not enough

The GPU is the only place where:

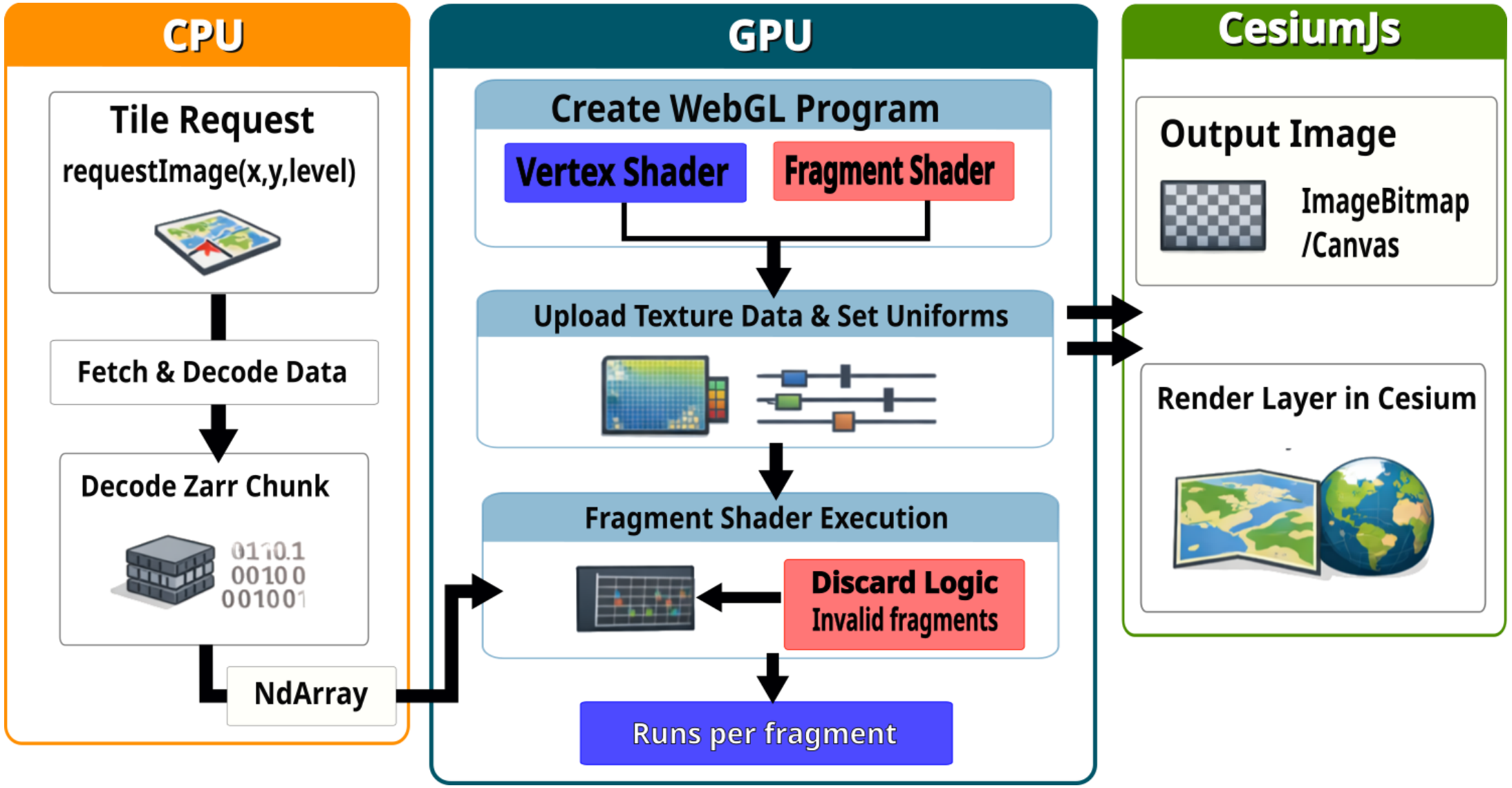
- Per-pixel math is cheap
- Colormaps can change instantly
- Rendering scales with screen resolution, not with data size

Zarr-Cesium pushes all visualisation math onto the GPU, by using WebGL



GPU Paint | **CPU Think**

WebGL: GPU Rendering in the Browser



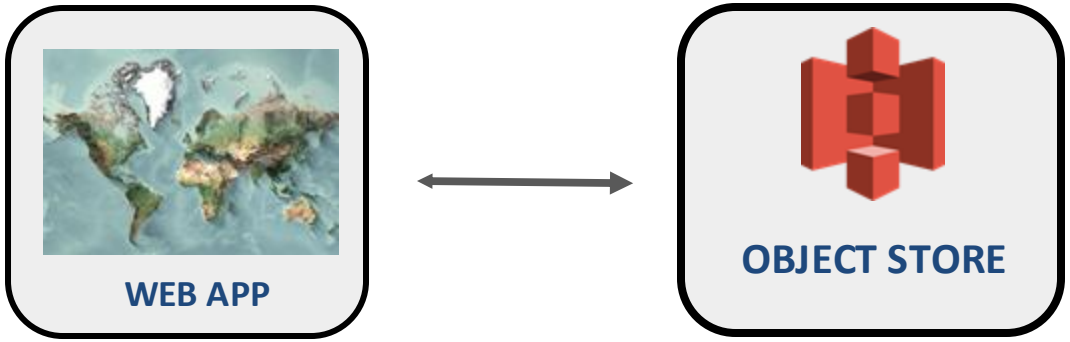
Data Providers

THREE DATA PROVIDERS:

ZarrLayerProvider

ZarrCubeProvider

ZarrCubeVelocityProvider



https://noc-oi.github.io/zarr-cesium/

Everything runs client-side.

ZarrLayerProvider (2D Scalar Fields)

National Oceanography Centre

Cesium ImageryLayer directly from Zarr

- Like WMS, titiler, carbonplan/maps layers

No tiles, no PNGs: **Zarr → WebGL.**



```
const layer = await ZarrLayerProvider.createLayer(viewer,{
    url: 'https://example.com/data.zarr',
    variable: 'salinity',
    colormap: 'viridis',
    scale: [30, 40]
});
viewer.imageryLayers.add(layer);
```

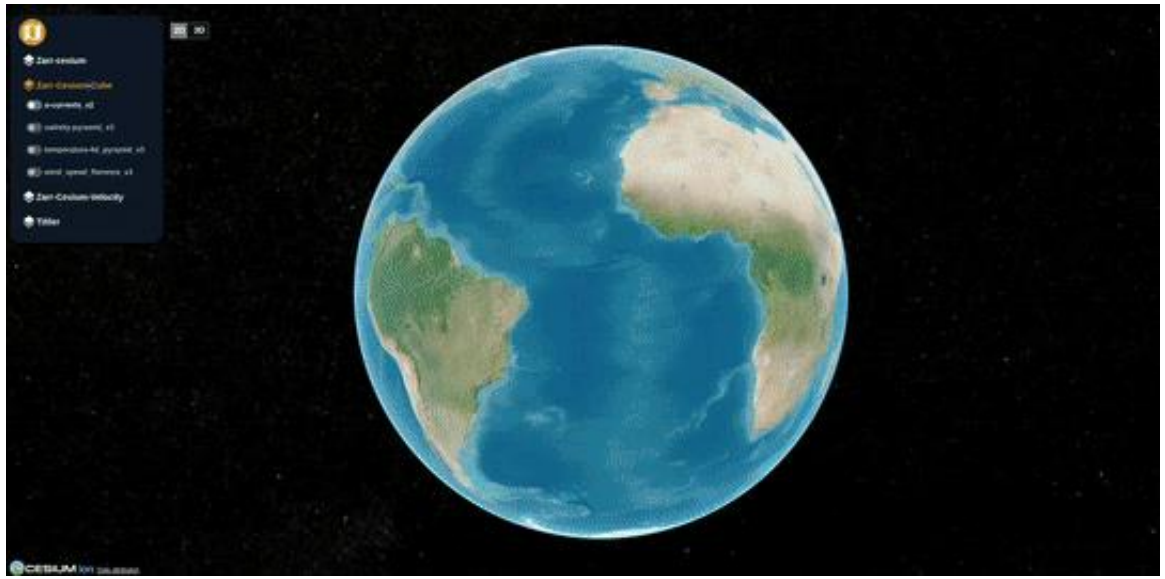
ZarrCubeProvider (3D Volumes)

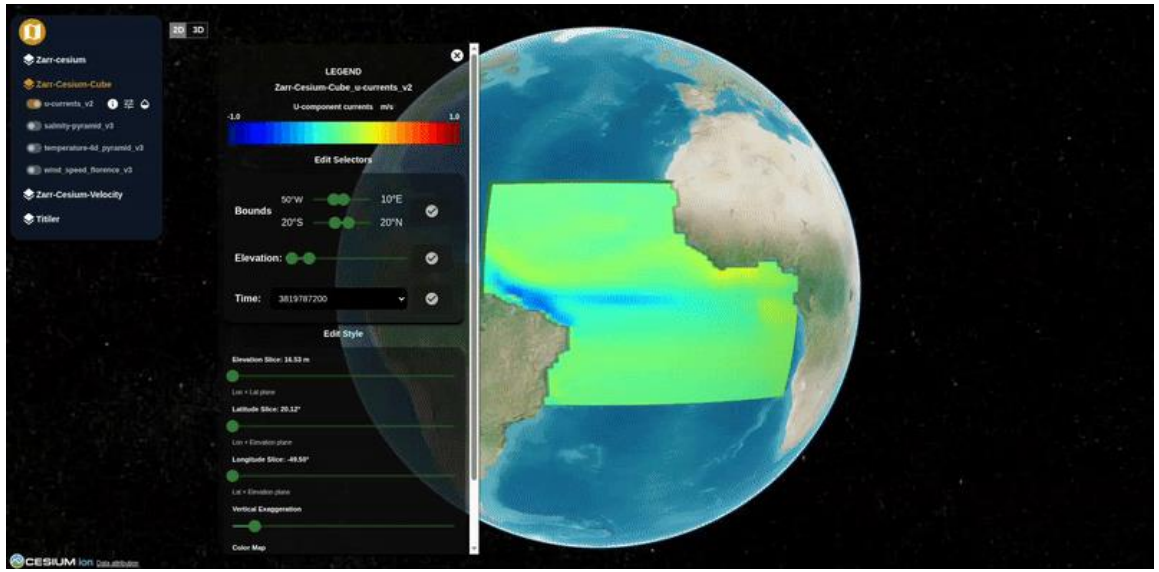
3D exploratory analysis right in the browser of scalar cubes (lon × lat × depth)

- Slice cubes horizontally & vertically
- Style: vertical exaggeration, opacity, colormap shading

A real 3D cube, with no shaders trickery and no pre-rendered images.

```
const cube = new ZarrCubeProvider(viewer, options);
await cube.load();
cube.updateSlices({
  latIndex: 20,
  lonIndex: 15,
  elevationIndex: 5
});
```





ZarrCubeVelocityProvider (3D Vector Fields)

National Oceanography Centre

Fully animated particle-flow layers rendered directly from Zarr U/V velocity components

- Uses **cesium-wind-layer**
- Fly through currents & winds in 3D





```
const options = { urls: {
  u: 'https://example.com/uo.zarr',
  v: 'https://example.com/vo.zarr'},
  variables: { u: 'uo', v: 'vo' },
  bounds: { west: -10, south: 30, east: 10, north: 45 }
};
const windCube = new ZarrCubeVelocityProvider(viewer, options);
await windCube.load();
```

Next Steps

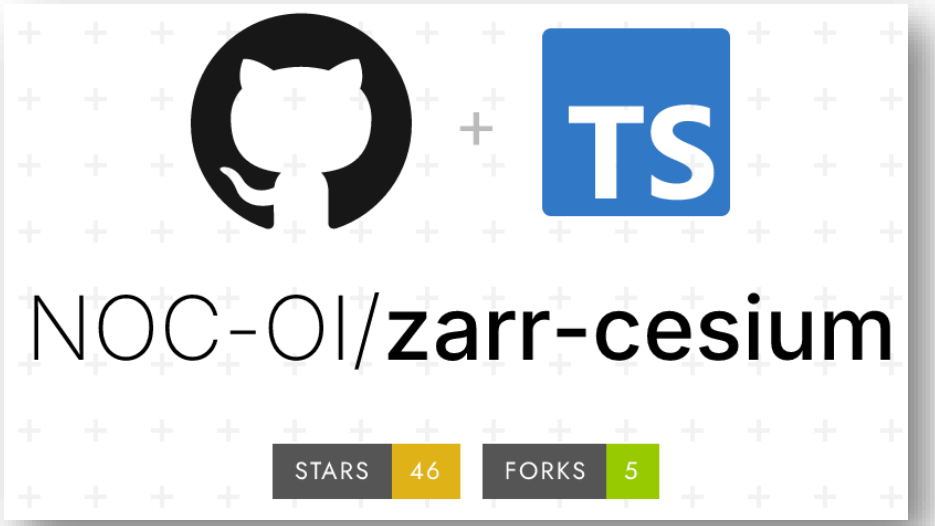
National Oceanography Centre

- Test integration with different Zarr datasets
- Increase performance
- Increase compatibility with other projections using proj4
- Allow the integration with non-Zarr datasets

Closing remarks

Zarr-Cesium aims to “just work”, being easy to integrate with existing Cesium dashboards.

```
npm install zarr-cesium
```



DOCS: https://noc-oi.github.io/zarr-cesium/docs

DEMO: https://noc-oi.github.io/zarr-cesium/

GITHUB: https://github.com/NOC-OI/zarr-cesium

AtlantiS
ATLANTIC CLIMATE AND ENVIRONMENT IMPACTS

National Oceanography Centre



Thanks you!

TOBIAS FERREIRA

TOBIAS.FERREIRA@NOC.AC.UK



@ RSE Geoscience SIG Spring Seminar | May 18, 2026

Daniel Nüst | https://orcid.org/0000-0002-0024-5046

https://github.com/nuest/geoextent/

Slides: t1p.de/geoextent-rse

CC BY

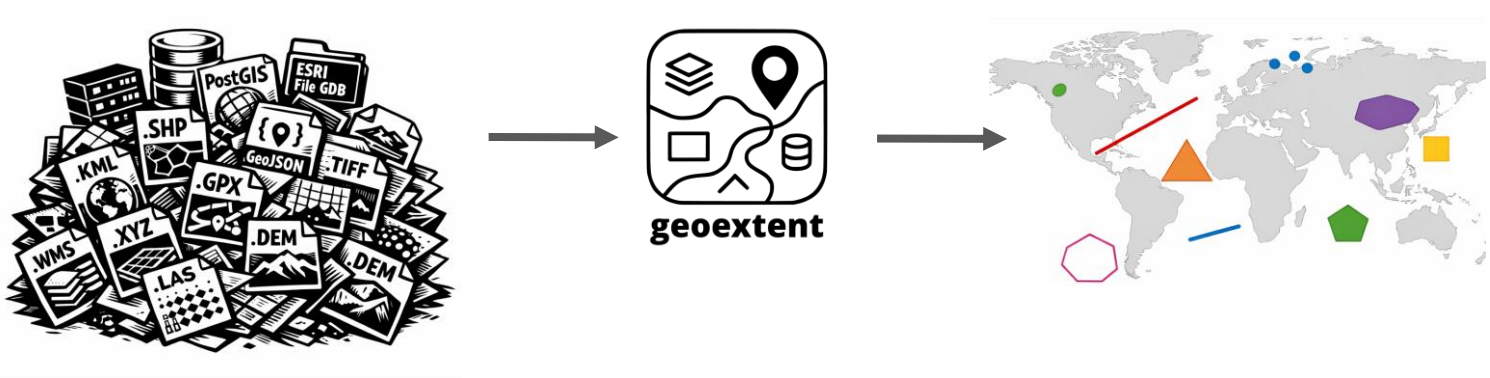


the tool



use case

geoextent is a Python library + CLI for **extracting geospatial extent** and temporal extent of data files and directories with multiple data formats.
files can be local, or remote, identified via URL **or identifier** (DOI).
why? geospatial metadata can improve **data discovery and assessment** in research.



Illustrations: you.com

features

geoextent

multiple input format support:
GeoJSON, CSV, Shapefile, GeoTIFF, GeoPackage, GPX, KML, GML, FlatGeobuf, ...

repository integration (handles, DOIs)
with metadata/API variants (prefer, fallback):
Zenodo, PANGAEA, OSF, Figshare, Dryad, GFZ Data Services, ...

flexible **extent types**:
bounding boxes, temporal extents, and convex hulls;
using advanced filtering: (soft) size limits, file type filtering, and parallel downloads;

rich **output options**: GeoJSON, WKT, WKB formats with optional geojson.io URL generation or static map preview (in CLI); save to file;
geographic context via placename lookup using multiple gazetteer services;

advanced: multi-processing/workers, metadata enrichment;

repositories



35 research repositories are supported:

Zenodo, PANGAEA, OSF, Figshare, 4TU.ResearchData, Dryad, GFZ, RADAR, Arctic Data Center, DataONE, B2SHARE, MDI-DE, GDI-DE, NFDI4Earth, SEANOE, GeoScienceWorld, UKCEH, GBIF, DEIMS-SDR, HALO DB, GitHub, GitLab, Software Heritage, Dataverse [Harvard, DataverseNL, DataverseNO, UNC, UVA, Recherche Data Gouv, ioerDATA, heiDATA, Edmond], Pensoft, TU Dresden Opara, Senckenberg, BGR, BAW, Mendeley Data), Wikidata, any STAC catalog, and any CKAN instance (e.g. data.gov.uk, GovData.de, data.gov.au, data.gov.ie

open an issue if your favourite is **missing**; **try** out the ones you use, please

foundations & project basics

geoextent

github.com/nuest/geoextent/
MIT license

DOTADIW principle*

pypi.org/project/geoextent/

gdal.org/





examples

example CLI commands: Zenodo

geoextent

```
$ python -m geoextent -b -t 10.5281/zenodo.4593540
Downloading files: 100%|████████████████| 2.84M/2.84M [00:00<00:00, 3.87MB/s, files=1/1]

{"type": "FeatureCollection",
  "features": [{"type": "Feature", "geometry": {"type": "Polygon", "coordinates": [[[-
80.71456319678893, 39.642802545572735], [-74.78657735361809, 39.642802545572735], [-
74.78657735361809, 42.256308231814586], [-80.71456319678893, 42.256308231814586], [-
80.71456319678893, 39.642802545572735]]]},
  "properties": {
    "tbox": ["2006-02-02", "2018-08-27"],
    "geoextent_extraction": {
      "version": "0.12.1.dev75+gc696a0e5b", "inputs": ["10.5281/zenodo.4593540"],
      "statistics": { "files_processed": 1, "files_with_extent": 1, "total_size": "2.71 MiB" },
      "format": "remote", "crs": "4326", "extent_type": "bounding_box" }
}
```

input record: https://doi.org/10.5281/zenodo.4593540

more examples: https://nuest.github.io/geoextent/examples.html

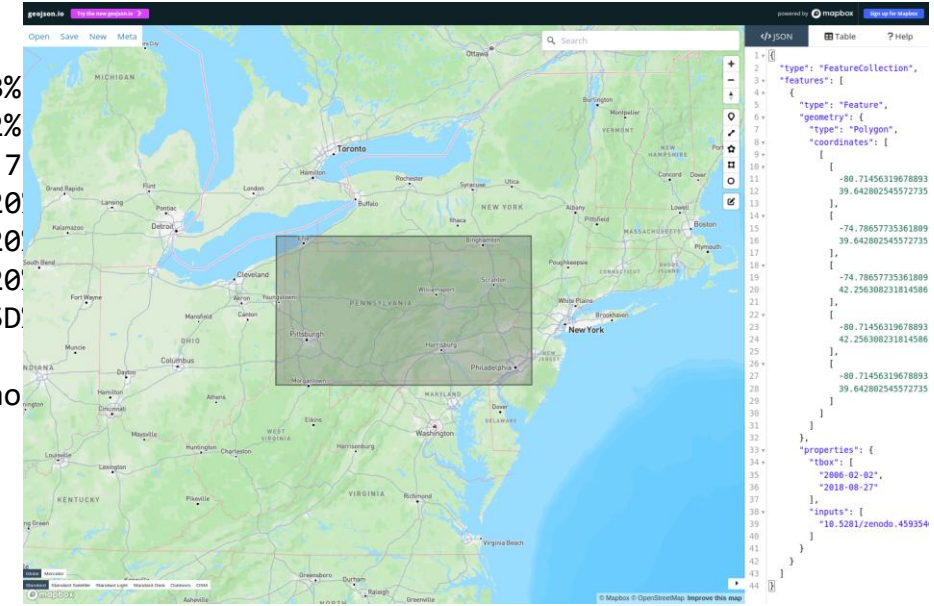
example CLI commands: quick explore

```
$ python -m geoextent -b -t --geojsonio --format wkt 10.5281/zenodo.4593540
Downloading files: 100%|████████████████| 2.84M/2.84M [00:00<00:00, 4.81MB/s, files=1/1]
POLYGON((39.642802545572735 -80.71456319678893,42.256308231814586 -
80.71456319678893,42.256308231814586 -74.78657735361809,39.642802545572735 -
74.78657735361809,39.642802545572735 -80.71456319678893))
```

🌎 View spatial extent at:

```
http://geojson.io/#data=data:application/json,%7B%
tures%22%3A%20%5B%7B%22type%22%3A%20%22Feature%22%
ygon%22%2C%20%22coordinates%22%3A%20%5B%5B%5B-80.7
74.78657735361809%2C%2039.642802545572735%5D%2C%20
74.78657735361809%2C%2042.256308231814586%5D%2C%20
80.71456319678893%2C%2042.256308231814586%5D%2C%20
80.71456319678893%2C%2039.642802545572735%5D%5D%5D
%5B%222006-02-02%22%2C%20%222018-08-
27%22%5D%2C%20%22inputs%22%3A%20%5B%2210.5281/zeno
```

click link



example CLI commands: outputs & pipes




```
# Extract extent as WKB (hex)
BBOX_WKB=$(python -m geoextent -b --format wkb --quiet path/to/data.geojson)

# Insert into PostGIS using WKB (more precise than WKT)
psql -d mydb -c "INSERT INTO extents (name, geom)
  VALUES ('my_dataset', ST_GeomFromWKB(decode('${BBOX_WKB}', 'hex'), 4326));"
```

example CLI commands: places & metadata




```
$ python -m geoextent -b -t --placename --convex-hull --ext-metadata 10.5281/zenodo.4593540
… {"placename": "United States", …

… [{"type": "Feature", "geometry": {"type": "Polygon", "coordinates": [[[-80.34454763988143, 39.72868251600846], [-80.34763849112053,
39.72897890634001], [-80.34873085917073, 39.72912432653343], [-80.3559400455179, 39.73032666388894], [-80.35987496535228, 39.73107328325734], [-80.36324008343719, 39.73281610925867], [-80.45279095682073,
39.78341366333026], [-80.45527625638229, 39.78549094163208], [-80.45680220780203, 39.78806110199079], [-80.53025476739768, 39.959172458011544], [-80.5308229029596, 39.96193947346666], [-80.5315051625855,
41.950536121220686], [-80.53149818959047, 41.97720544819606], [-80.53092368551297, 41.979964137822925], [-80.5292418046151, 41.98245800086316], [-80.5266148836029, 41.98444614560855], [-80.52593478101528,
41.98482118889403], [-80.50610908498477, 41.992717279096446], [-80.50335477962628, 41.99372675998329], [-80.49916888766874, 41.995167508977545], [-80.4979057927458, 41.99554893036651], [-80.49768181093057,
41.99561640560673], [-80.47267836507052, 42.00231289723687], [-80.4707020661498, 42.002835609751116], [-79.8458803746571, 42.153802476968636], [-79.84206765917911, 42.15419329664985], [-75.36068786967296,
42.00818951404845], [-75.35726242222862, 42.007805336326676], [-75.33677193862827, 42.000802367973385], [-75.3359297899013, 42.00045918254918], [-75.33476163655337, 41.99991646846765], [-75.3340986285742,
41.9995678058514], [-75.33066054019068, 41.99678801538373], [-74.79756704364871, 41.44407416969962], [-74.79065993960597, 41.43419199660388], [-74.78943083448655, 41.43158674556256], [-74.78930956418844,
41.42882365893981], [-75.08601388494928, 40.50665757713053], [-75.08621469118276, 40.50613508421368], [-75.08773082112455, 40.503660947489315], [-75.09014849689757, 40.501647302054394], [-
75.55544251663868, 40.1930119323641 …

"properties": { "external_metadata":
  [{"source": "DataCite", "doi": "10.5281/zenodo.4593540",
  "title": "Pennsylvania SGL with 1km buffer (GEOJSON)",
  "authors": ["Conner, Weston"], "publisher": "Zenodo", "publication_year": 2021,
  "url": "https://zenodo.org/record/4593540", "license": …
```

input record: https://doi.org/10.5281/zenodo.4593540

more examples: https://nuest.github.io/geoextent/examples.html

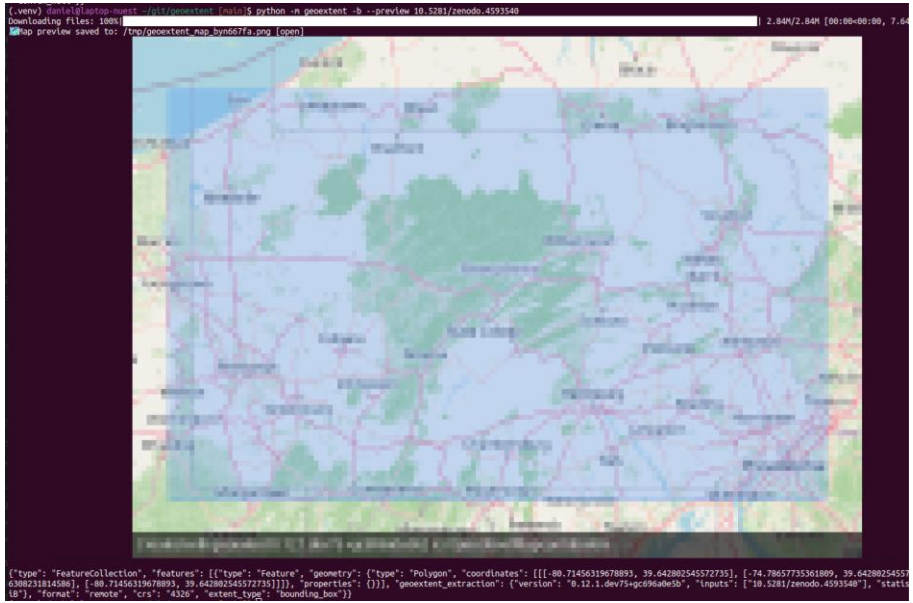
example CLI commands: placenames (cont.)

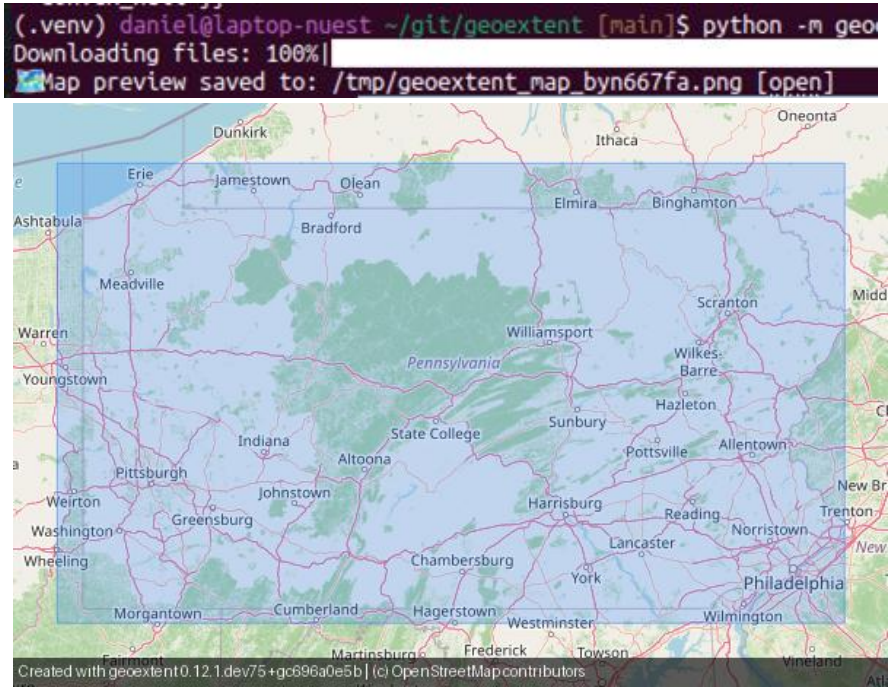



```
$ pyt…
```

click link

example CLI commands: map output CLI & file





input record: https://doi.org/10.5281/zenodo.4593540

more examples: https://nuest.github.io/geoextent/examples.html

example CLI commands: NER places and times



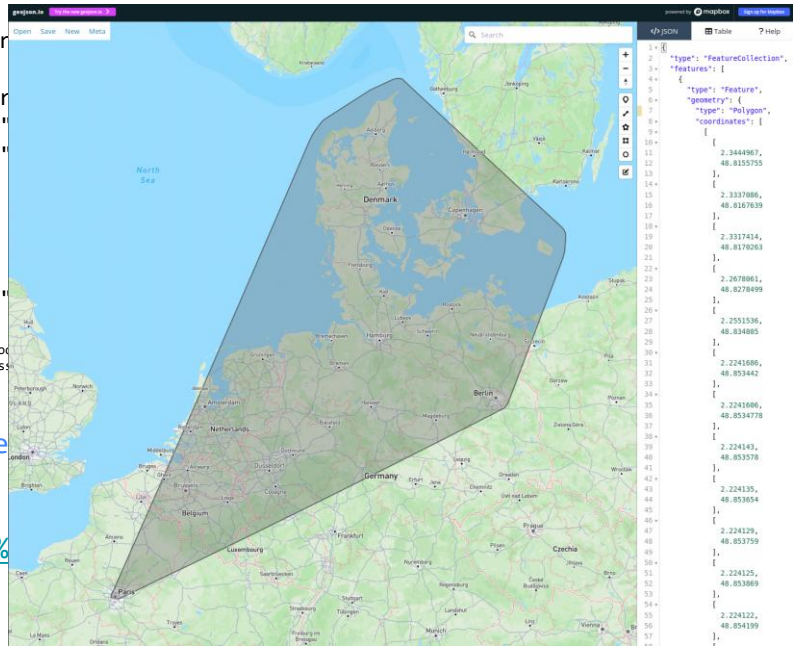

```
$ geoextent -b -t --convex-hull --ner-ambiguity top --text "Field campaigns in Berlin in
Spring 2020, Paris in 2024, and Denmark from 2025 to 2026" --geojsonio
```


```
{"type": "FeatureCollection", "features": [{"type": "Feature", "geometry": {"type": "Polygon", "coordinates":
[[[2.3444967, 48.8155755], … … [2.3444967, 48.8155755]]]},
"properties": {"extraction_method": "ner", "ner_model": "en_core_web_sm", "r
"period_gazetteer": "ics", "place_names":
[ {"name": "Berlin", "label": "GPE", "char_start": 19, "char_end": 25, "scor
"nominatim", "matched": true, "candidate_count": 1, "lat": 52.5173885, "lon"
"Berlin, Deutschland", "gazetteer_id": "osm:relation:62422", "gazetteer_url"
"https://www.openstreetmap.org/relation/62422"},
  {"name": "Paris", …},
  {"name": "Denmark", …}],

"date_entities": [{"text": "Spring 2020", …}, {"text": "2024", …, "start": "
31"}, {"text": "2025", …}],
"source_text": "Field campaigns in Berlin in Spring 2020, Paris in 2024, and Denmark from 2025 to 2026", "source_offset_unit": "python_co
01", "2025-12-31"]}]}, "geoextent_extraction": {"version": "0.13.1.dev3+g79eb7c0f4.d20260518", "inputs": [], "statistics": {"files_proces
"format": "text", "geoextent_handler": "handle_text", "crs": "4326", "extent_type": "convex_hull"}}
```


```
---annotated source (ansi)---
Field campaigns in Berlin in Spring 2020, Paris in 2024, and De
```

🌎 View spatial extent at:
http://geojson.io/#data=data:application/json,%7B%22type%22%3A%



based on **SpaCy**

more examples: https://nuest.github.io/geoextent/howto/text-extraction.html

Python API

geoextent

current public API functions:

- from_file() - extract extent from individual files
- from_directory() - extract extent from directories
- from_remote() - extract extent from remote sources (repositories, journals, preprint servers)

https://nuest.github.io/geoextent/howto/api.html

example Python code

geoextent

```python
from geoextent.lib import extent

# Retrieve extent and metadata
result = extent.from_remote(
    '10.5281/zenodo.4593540',
    bbox=True,
    ext_metadata=True
)

# Access metadata
metadata_list = result['external_metadata']
for metadata in metadata_list:
    print(f"Source: {metadata['source']}")
    print(f"Title: {metadata['title']}")
    print(f"Authors: {', '.join(metadata
    print(f"Year: {metadata['publication
```

```python
identifiers = [
    '10.5281/zenodo.4593540',
    '10.25532/OPARA-581',
    'https://osf.io/abc123/'
]

geoextent.from_remote(identifiers, True, True, True)
```



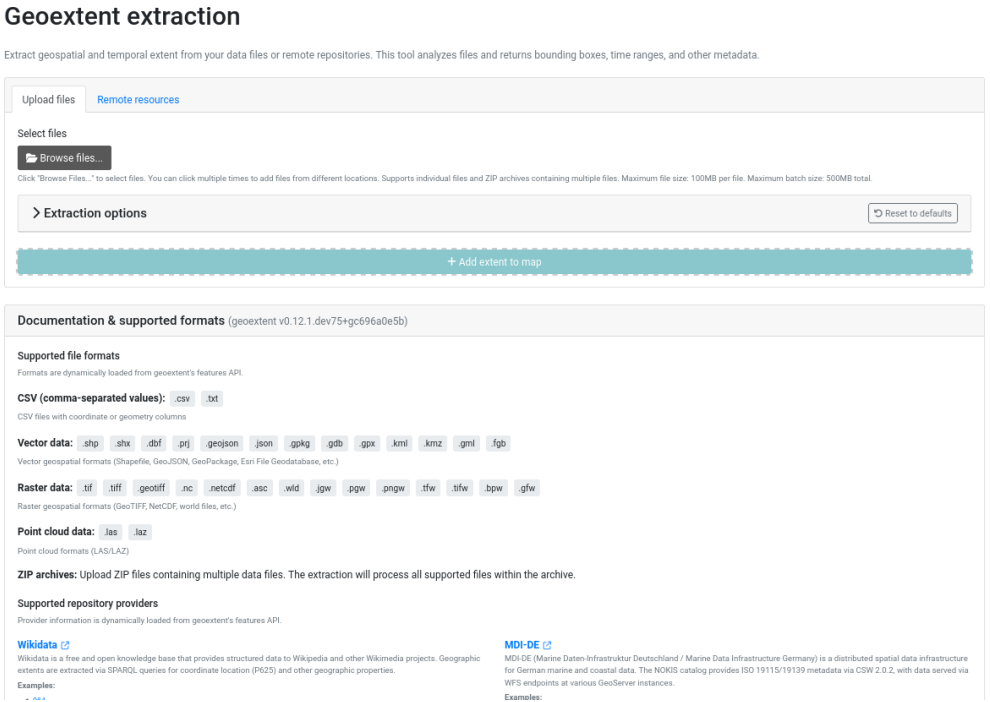
applications

try out online

geoextent

https://optimap.geo.tu-dresden.de/geoextent/

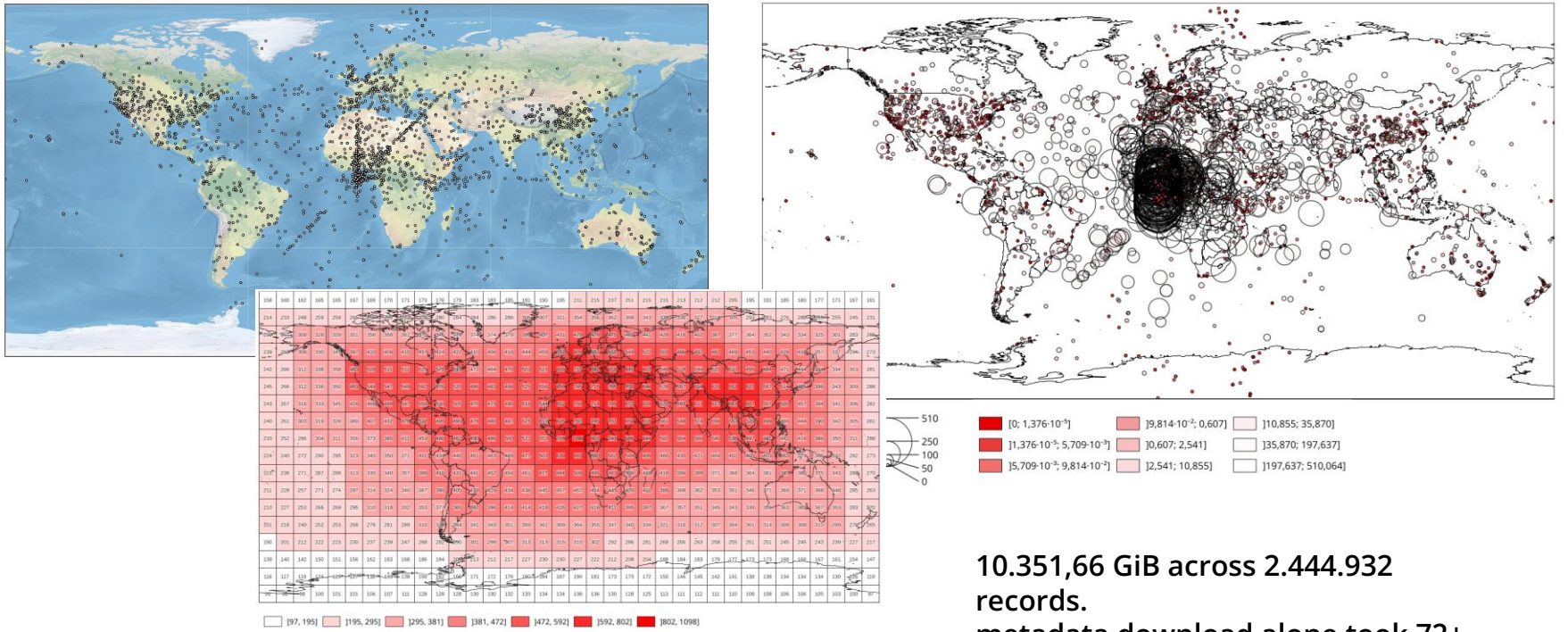
(limitations)



geospatial data in general purpose repositories

geoextent

BSc thesis (research manuscript under development)

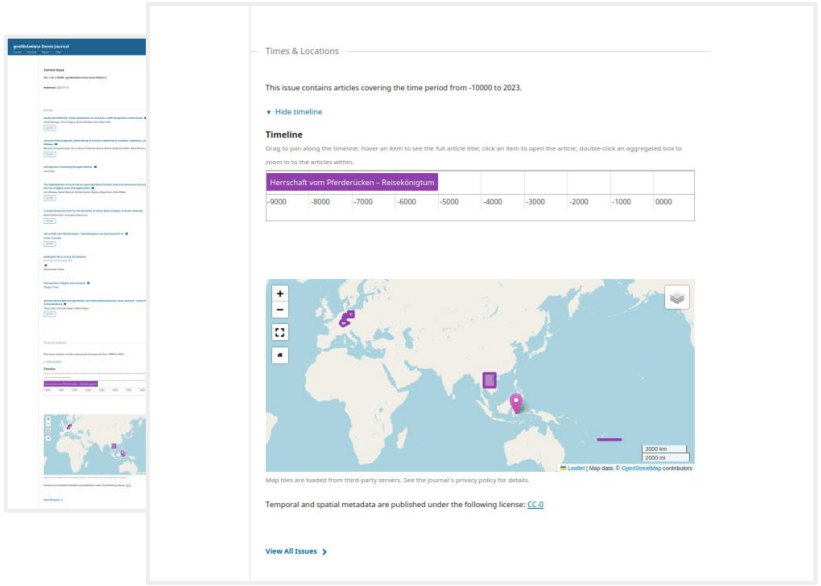


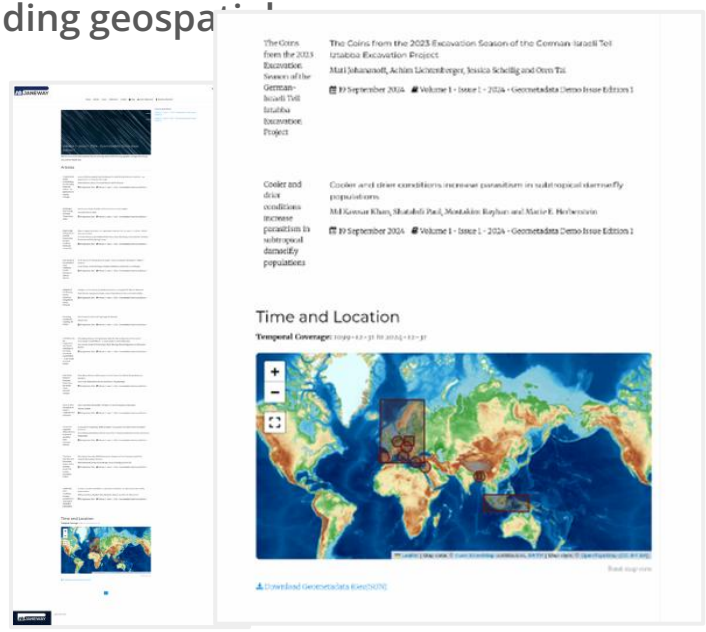
10.351,66 GiB across 2.444.932 records.
metadata download alone took 72+

plan: geospatial metadata for research works

User assistance/automation for adding geospatial metadata to research papers

Open Journal Systems (OJS) and Janeway plugins for adding geospa





Nüst, D., Niers, T., Yücel, G., & Anette Coordts. (2023). TIBHannover/optimetaGeo: Ushuaia (v1.0.1.0-beta). Zenodo. https://doi.org/10.5281/zenodo.8199005 | **https://github.com/TIBHannover/geoMetadata**
Nüst, D. (2026). Geometadata Plugin for Janeway (v0.2.0). Zenodo. https://doi.org/10.5281/zenodo.20259785 | **https://github.com/GeoinformationSystems/janeway_geometadata/**

next



examples for using with different languages/tools (R, QGIS) ?

S2 for geometric calculations?

WASM? (tricky with GDAL)

Zenodo explorer?

your repository?

full text data, maybe LLMs, ...

API endpoint as a service to downstream tools (e.g., publishing)

All contributions are welcome!

https://github.com/users/nuest/projects/1?query=sort%3Aupdated-desc+is%3Aopen



the development

coding assistant dev



Anthropic's Claude



straightforward tasks:

adding a content provider

(example: https://github.com/nuest/geoextent/commit/49e9c0d97812bc7920692c5911ad5f657b456f0d)

adding **test cases** (sometimes too extensive, great for catching misunderstandings)

tricky tasks:

documentation generation and test generation can get out of hand

git repo as **insurance**, only commit yourself (but let AI stage specific lines and write commit messages - which must be shortened most of the time)

the many things I would not have implemented manually

geoextent

NER feature > https://github.com/nuest/geoextent/commit/abc257ad4134a821a327805d8899243fb96e975b

number of repositories (not as fast)

massive amount of **tests and cross OS pipelines** (housekeeping, knowledge)

```
investigate the failing tests in CI: https://github.com/nuest/geoextent/actions/runs/26023252113
apply both fixes
```

documentation with examples for each feature

documentation for installation with 5 different **package managers** (fiddling)

AppImage variant (no personal use)

static map & CLI preview & coloured output in CLI (research demand)

machine-readable features list (maintenance)

multiple gazetteers (no immediate benefit)



thank you!

what are your questions?

This project was initially developed as part of the German Research Foundation (DFG)-funded research project Opening Reproducible Research (o2r, DFG project no. PE 1632/10-1 and PE 1632/17-1). Continuing development is supported by the DFG-funded NFDI4Earth consortium (DFG project no. 460036893) and by the German Federal Ministry of Research, Technology and Space (BMFTR) through the KOMET project (BMFTR grant no. 16KOA009B).

geoextent

RSE Geoscience Spring Seminar, May 18, 2026
https://github.com/nuest/geoextent/

Daniel Nüst | https://orcid.org/0000-0002-0024-5046

Slides: **https://t1p.de/geoextent-rse**

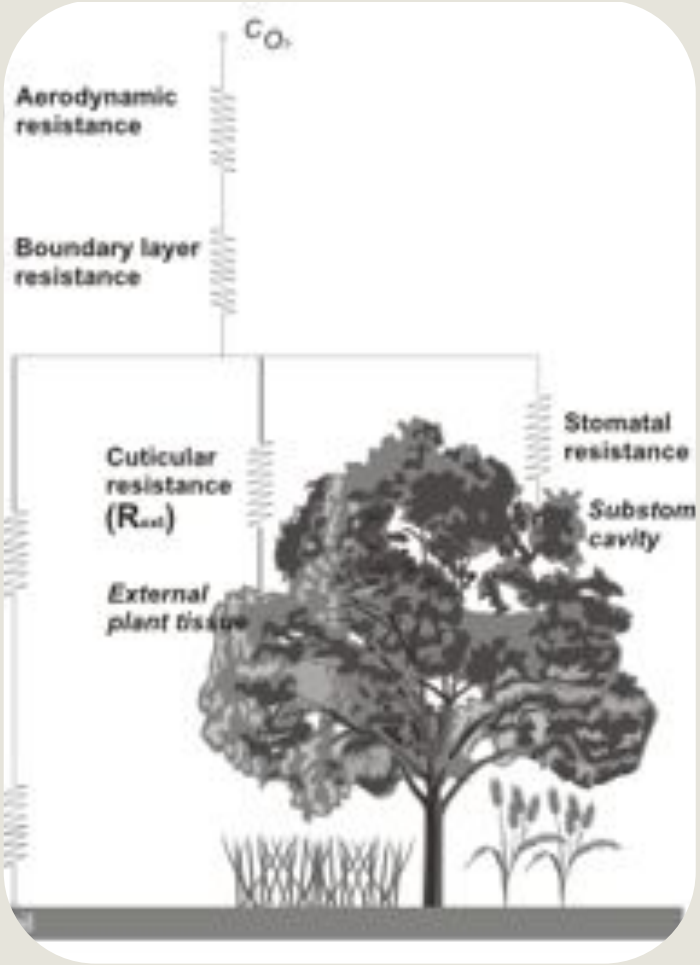
CC BY

Sam Bland



DO3SE Nextflow Pipeline

Development Challenges and Successes



DO3SE model

Primarily an ozone deposition model with soil moisture and crop growth modules.

- Works on hourly met data
- Used for both single site and large scale grid runs

DO3SE: Deposition of Ozone for Stomatal Exchange

Model development history

1 PhD

30+ Years

5 separate developers
+
1 PI Researcher



Fortran CLI

Fortran + Python GUI

Python only

Python + Nextflow

Moving to gridded data

Input Data Progression

- Single location (< 1 mb)
- Regional grid data with single year (6-10GB)
- Global grid + many years + many scenarios + many cultivars (2+ TB)

Compute Platform Progression

- Local Machine
- Local HPC with bash script & Jupyter Notebooks
- Local HPC + Nextflow Pipeline

Early pipeline limitations

Human Factors

Every rerun of the model after input or model changes meant an error prone data pipeline between local and hpc

- Error prone
- Time consuming
- Hard to debug
- Not always reproducible
- Lack of transparency

Compute Limitations

- Difficult to scale up
- Lack of cache meant significant unnecessary re-runs

Choosing a pipeline



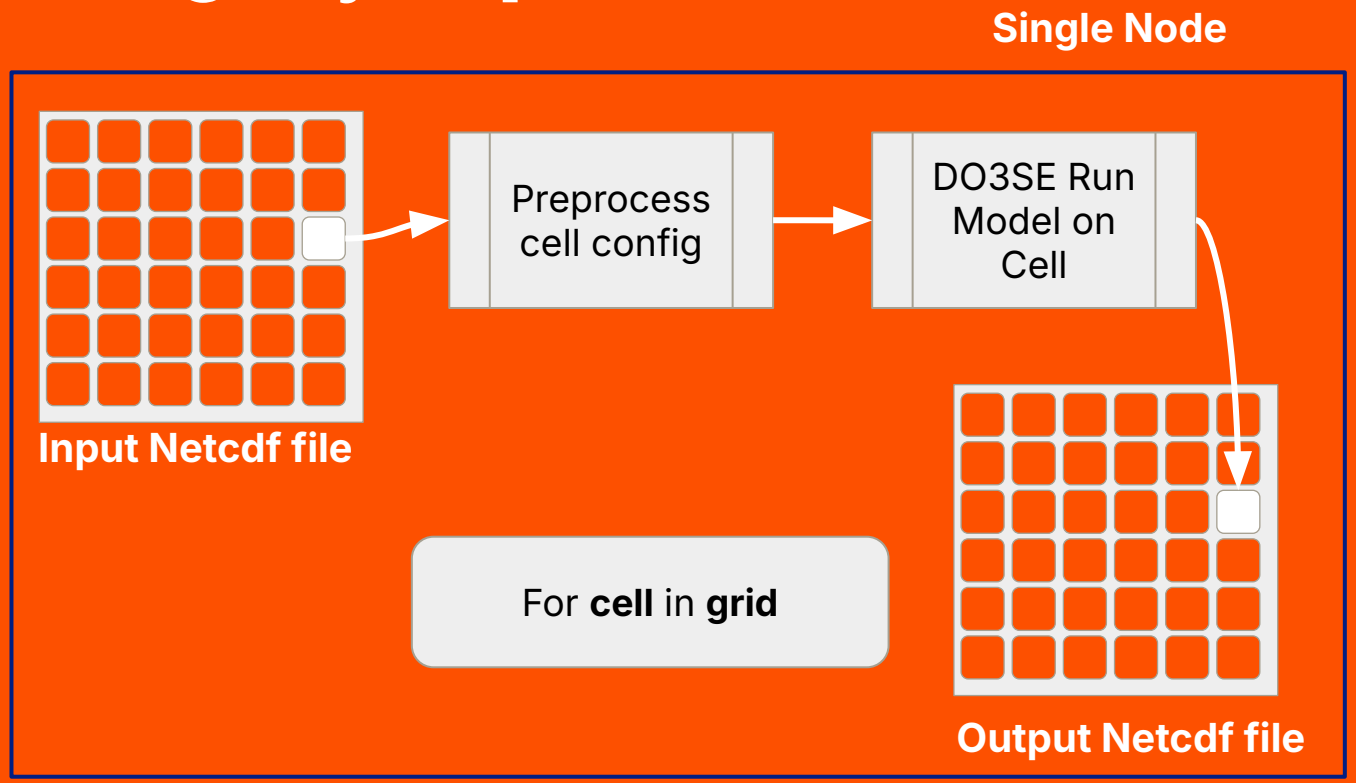


- Internal UoYork expertise
- After some initial testing it was clear it could do what we need

Pipeline steps

Download Data
Preprocess Data
Prepare any static inputs such as masks
Save the metadata that will be used to run the model
Run DO3SE model on each cell
Combine outputs
Create plots from outputs

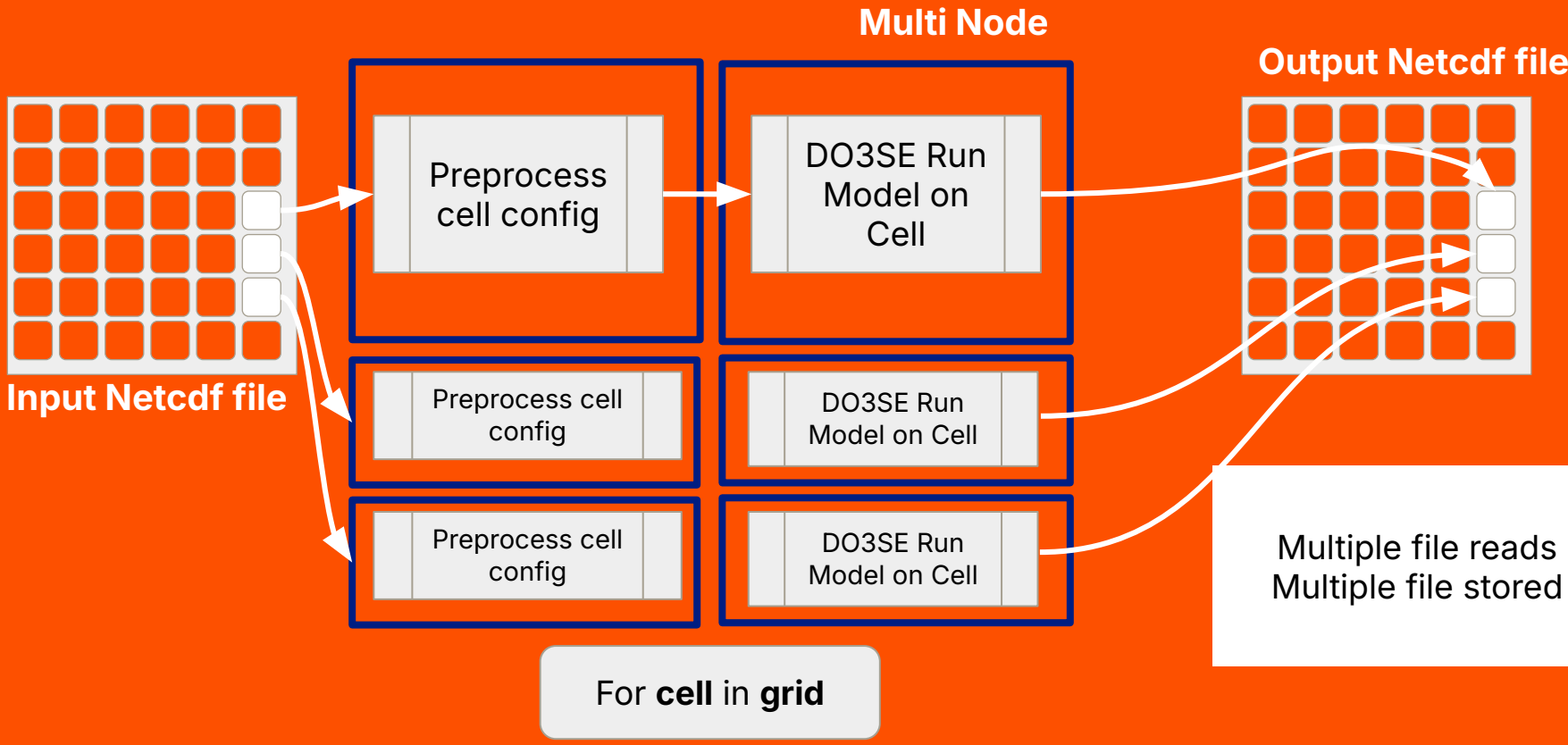
Legacy Pipeline

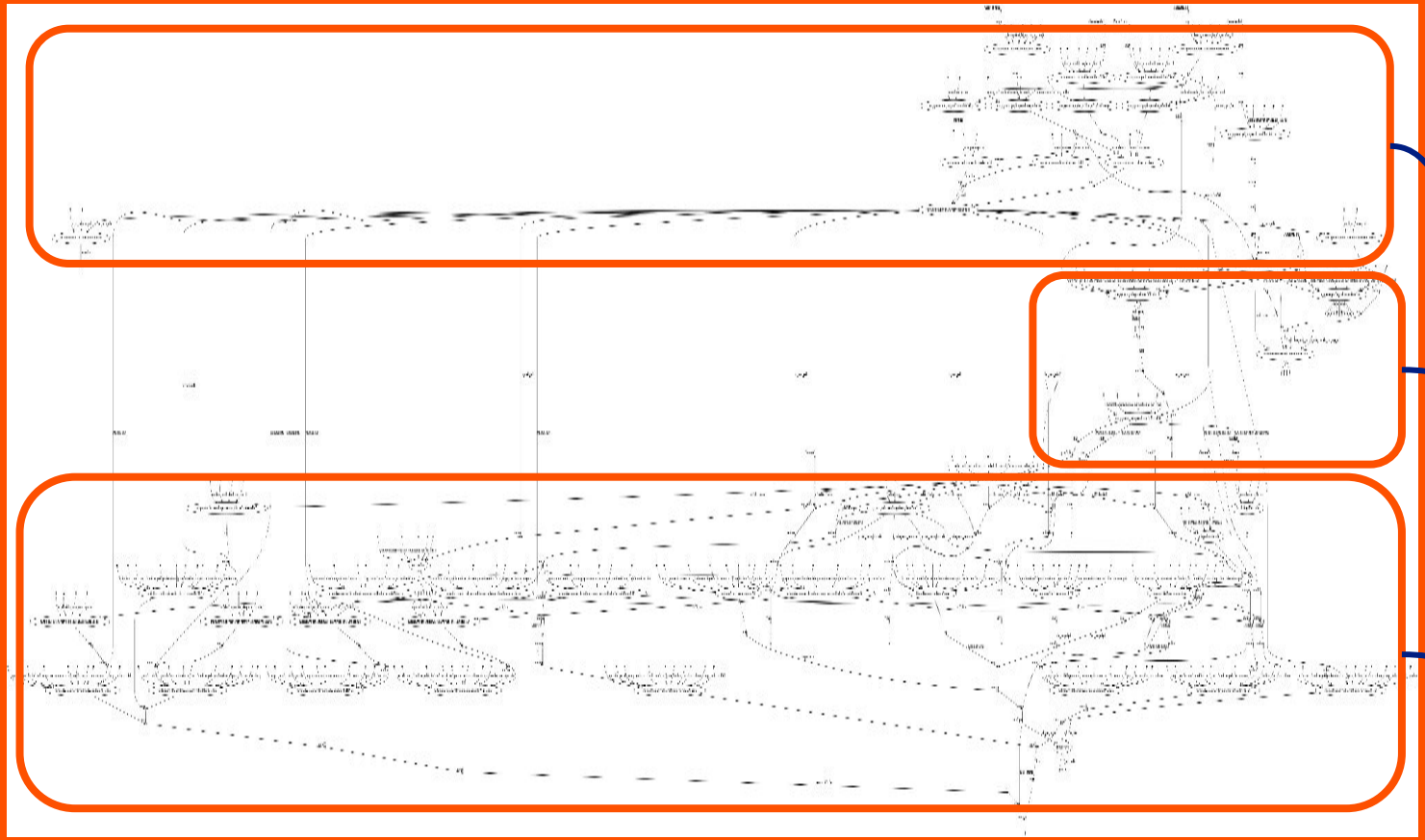


Single Python script execution

Single File Read
Single file stored

Nextflow Pipeline





Preprocessing

DO3SE Model Cell Process

Post process and output plots

Almost immediate benefits

Early wins

+ Learning
+ Development
+ Testing

a few days ⇒ to a couple of hours and a single click.



New Challenges - Data

Data storage requirements

Caching = less CPU duplication

Caching = More data storage

Changing approach

Where possible running the pipeline in a separate directory so that cache could easily be cleared and was not shared between dissimilar model runs (I.e. between different input datasets)

New Challenges - Testing

Fragmented Code

To enable more parallelisation with Nextflow some logic from the model was separated between pipeline processes. This meant it was no longer tested

Future testing approach

We are hoping to implement some end to end testing and reducing python code kept in Nextflow processes.

New Challenges - Compute

More parallelisation meant more compute.

- Much easier to send runs meant significant scaling up of compute
- More IO increased the overall compute of the pipeline compared to single node in memory approach

Solutions

- A number of sampling options were included in the pipeline to run smaller test runs
- Communicating between RSE, PI and HPC service providers
- Using the pipeline cache capability

Pipeline strategies



One pipeline to rule them all



Scaling to multiple projects

The challenge of generalisation

- Variable input Netcdf configurations
- Input source(file or API)
- Data pre processing
- DO3SE configurations
- Non met-data sources (e.g. phenology data)

Handling different datasets

- Wrfchem Outputs
- ERA5 outputs
- GISS
- CESM
- EMEP

Pipeline splitting - By Process

Tall or wide?

More processes

=

Leverage Workflow Manager benefits

=

Greater complexity and IO

Module or bespoke

Generalisation of the processes involves the same decision making process to generalising any software function.

Workflow modularised and shareable

Grid Processing

Masking

Cropping

Regridding

Merging

Plotting

Annual Variable Mean

Spatial annual max

Produce markdown report

Pipeline Splitting - By workflow

Single workflow pipeline for

All cells

Full season

Separate pipelines for

Cultivar (Crop Type)

Season

Input Data Source

Reproducible



Meta data

All config files

Model and dependency versions

Additional run notes

Pipeline git commit id

FAIR workflow principles

(Findable, Accessible, Interoperable, Reusable)

Applying the FAIR Principles to computational workflows | Scientific Data

Continue the discussion

Do you...
...have a similar pipeline?
...use a workflow management tool with spatial data?
...have a similar pipeline?

FAIR workflow principles

Findable	Accessible	Interoperable	Reusable
Meta data provision	Documentation	Standard formats	Clear licence

Brand theme

How to use

These are example slides, but they're all available in a range of colours. Just right-click your chosen slide and select 'apply layout'.

For image layouts, click on the icon to add your chosen image from Google or your computer.

Tips

- Use the brighter colours sparingly for impact
- Use this neutral background for longer text
- Try these combinations:
  - Grey and dark blue
  - Light and dark green
  - Orange and yellow
  - Purple and dark green
- Ensure you test the accessibility of your presentation using Grackle Docs

Presentation subject
Date

TITLE

Text

- List item 1
- List item 2
- List item 3

Text

- List item 1
- List item 2
- List item 3

Title

Text

- List item 1
- List item 2
- List item 3

Title

100

£1m

10th

Add a bold

message

Title

Text

- List item 1
- List item 2
- List item 3

Subtitle

Text

- List item 1
- List item 2
- List item 3

Subtitle

Text

- List item 1
- List item 2
- List item 3

Subtitle

Text

- List item 1
- List item 2
- List item 3

Add a bold

message

Title

Text

- List item 1
- List item 2
- List item 3

Title

Text

- List item 1
- List item 2
- List item 3

Title

Text

- List item 1
- List item 2
- List item 3

Instruction or section title



Url or call to action