Tutorial: Using NSDF for End-to-End Analysis of Scientific Data

Heberth Martinez[1], **Aashish Panta**[2], Paula Olaya[1], Gabriel Laboy[1], Jay Ashworth[1], Giorgio Scorzelli[2], Valerio Pascucci[2], Michela Taufer[1]

[1]University of Tennessee Knoxville, [2]University of Utah

Feb 28, 2024 - San Diego, California, USA - **AHM 2024**

Acknowledgments

The authors of this tutorial would like to express their gratitude to:

- NSF through awards 2138811, 2103845, 2334945, 2138296, and 2331152
- The Dataverse team
- The Seal Storage team
- Vargas Lab led by Dr. Rodrigo Vargas

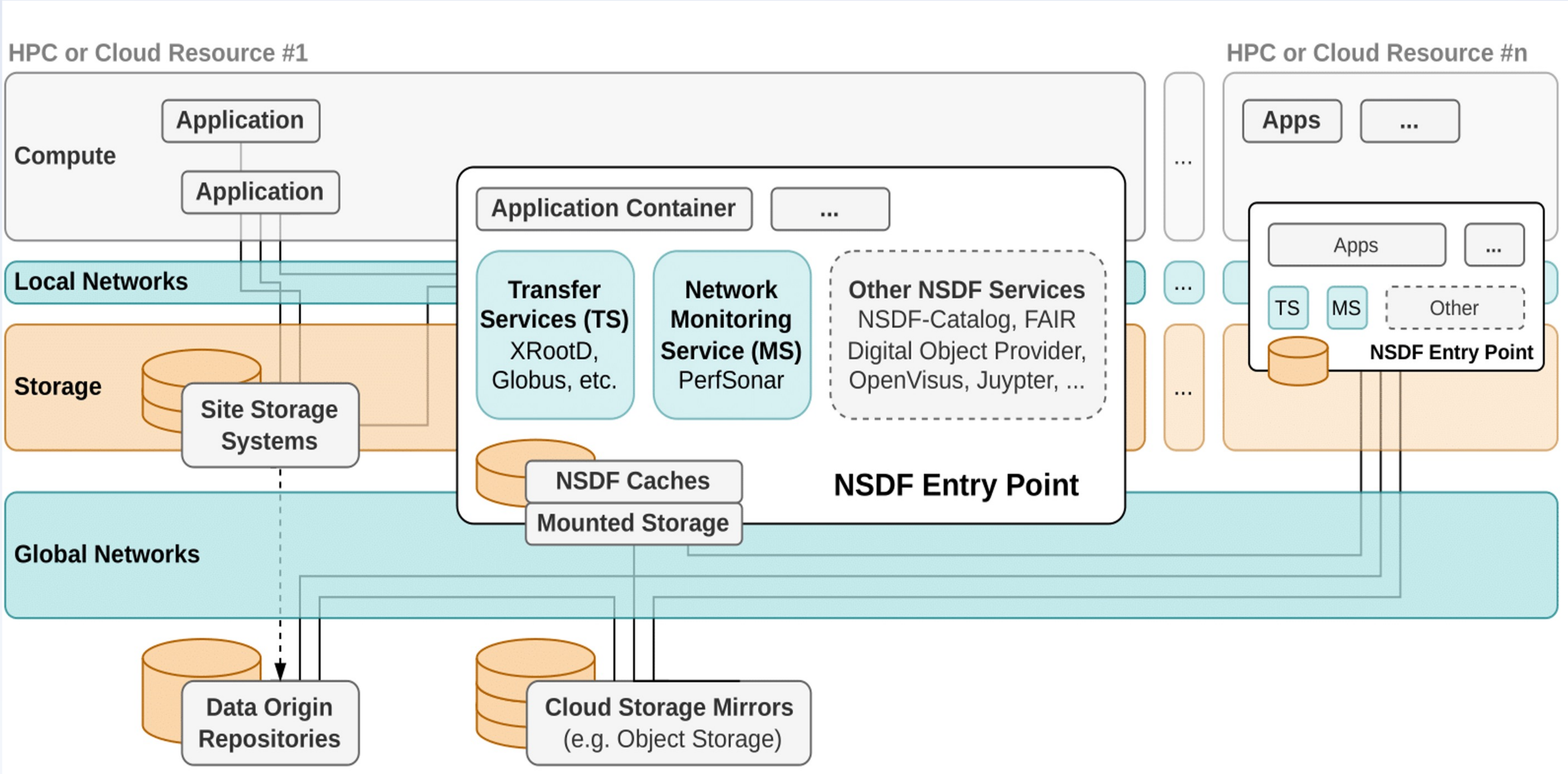
Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

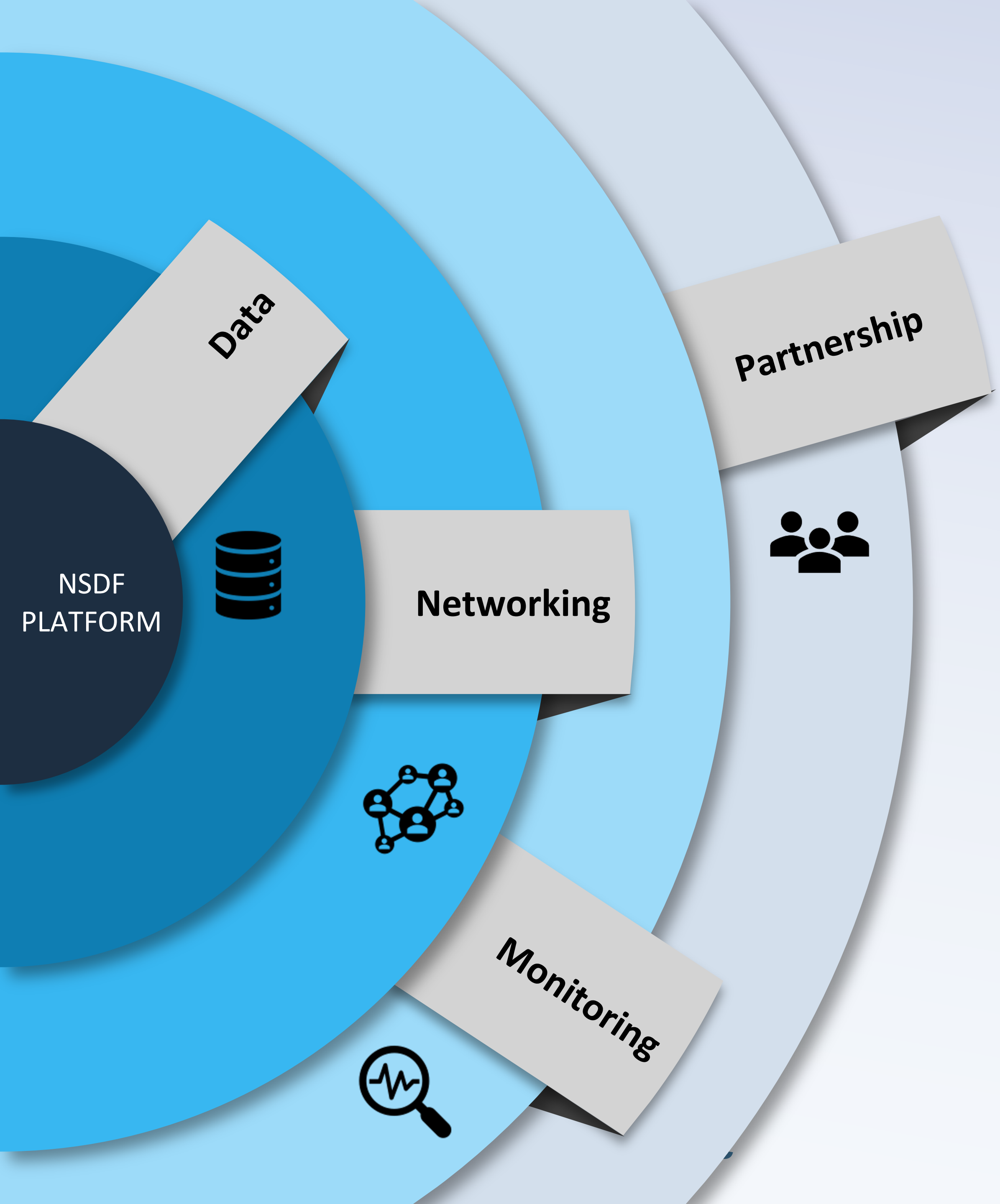
NSDF: The National Science Data Fabric

NSDF provides easy access and use of data through a platform agnostic software stack

NSDF Entry Points allow connecting and interacting with a wide range of research infrastructure

NSDF platform agnostic testbed for democratizing data delivery





Services & Building Blocks

Monitoring

Service monitoring

Partnership

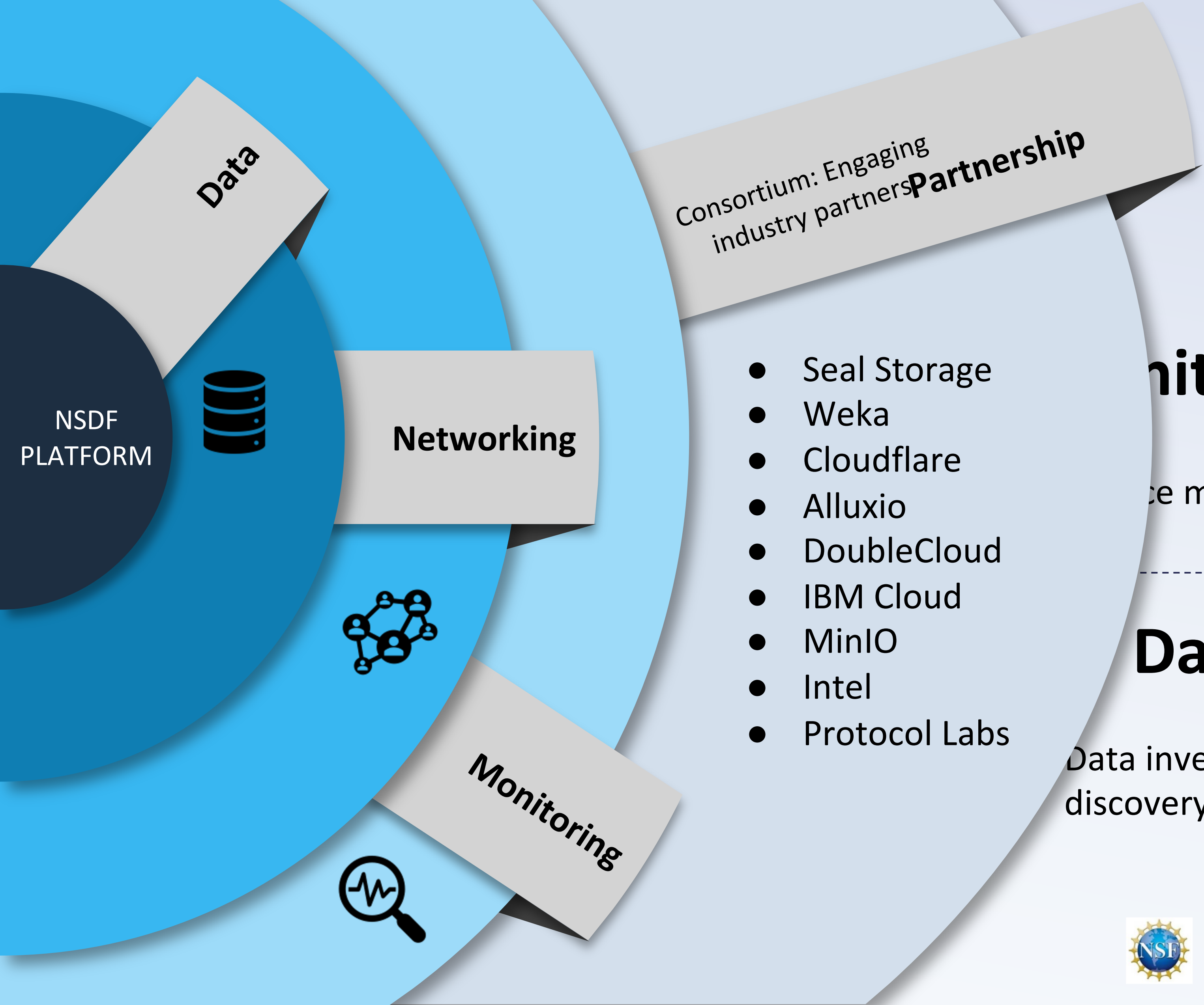
Users and commercial partners

Data

Data inventory and discovery

Networking

Network characterization



Services & Building Blocks

…nitoring

…ce monitoring

Partnership

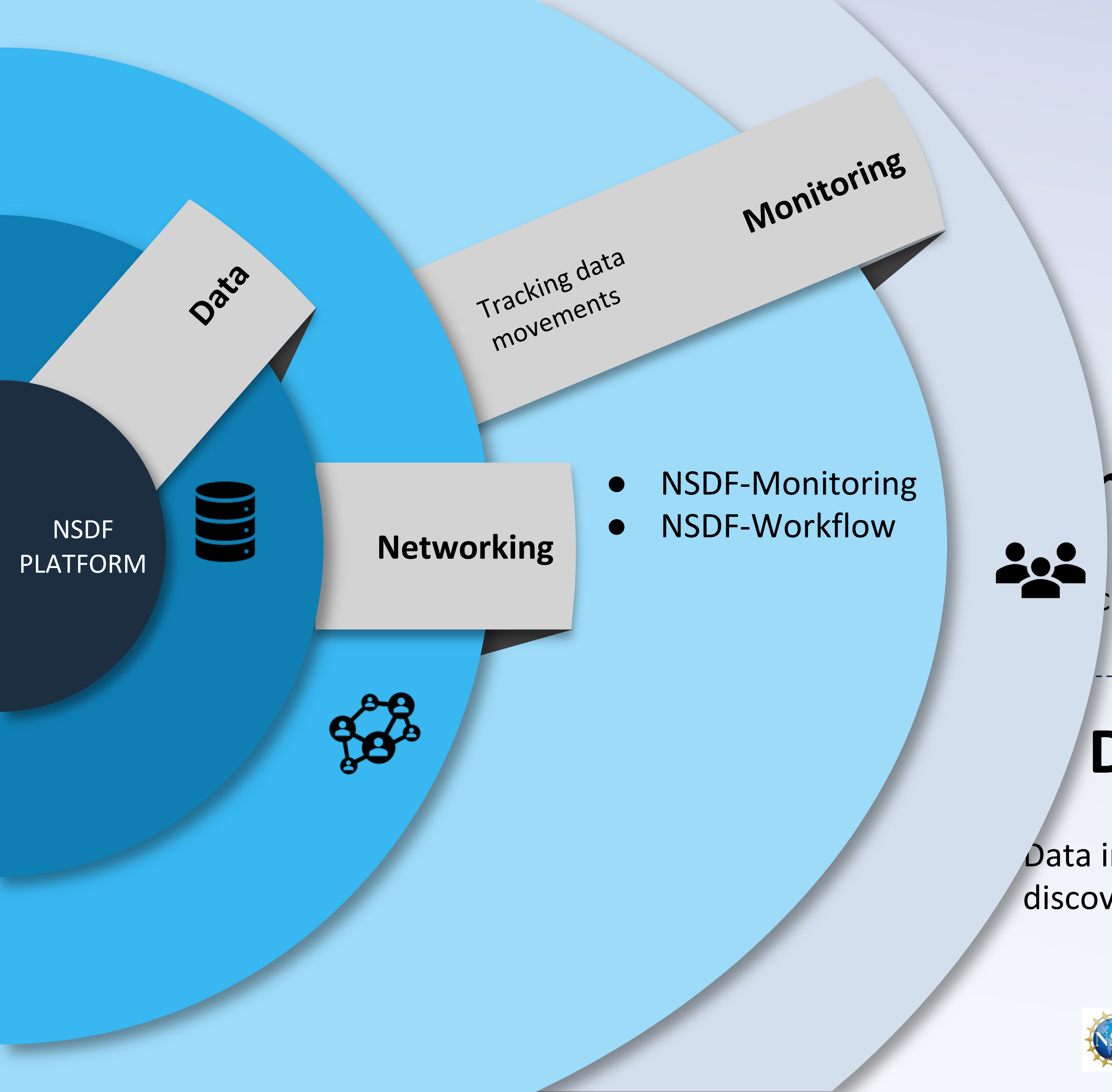
Users and commercial partners

Data

Data inventory and discovery

Networking

Network characterization



Services & Building Blocks

...nitoring

...ce monitoring

Partnership

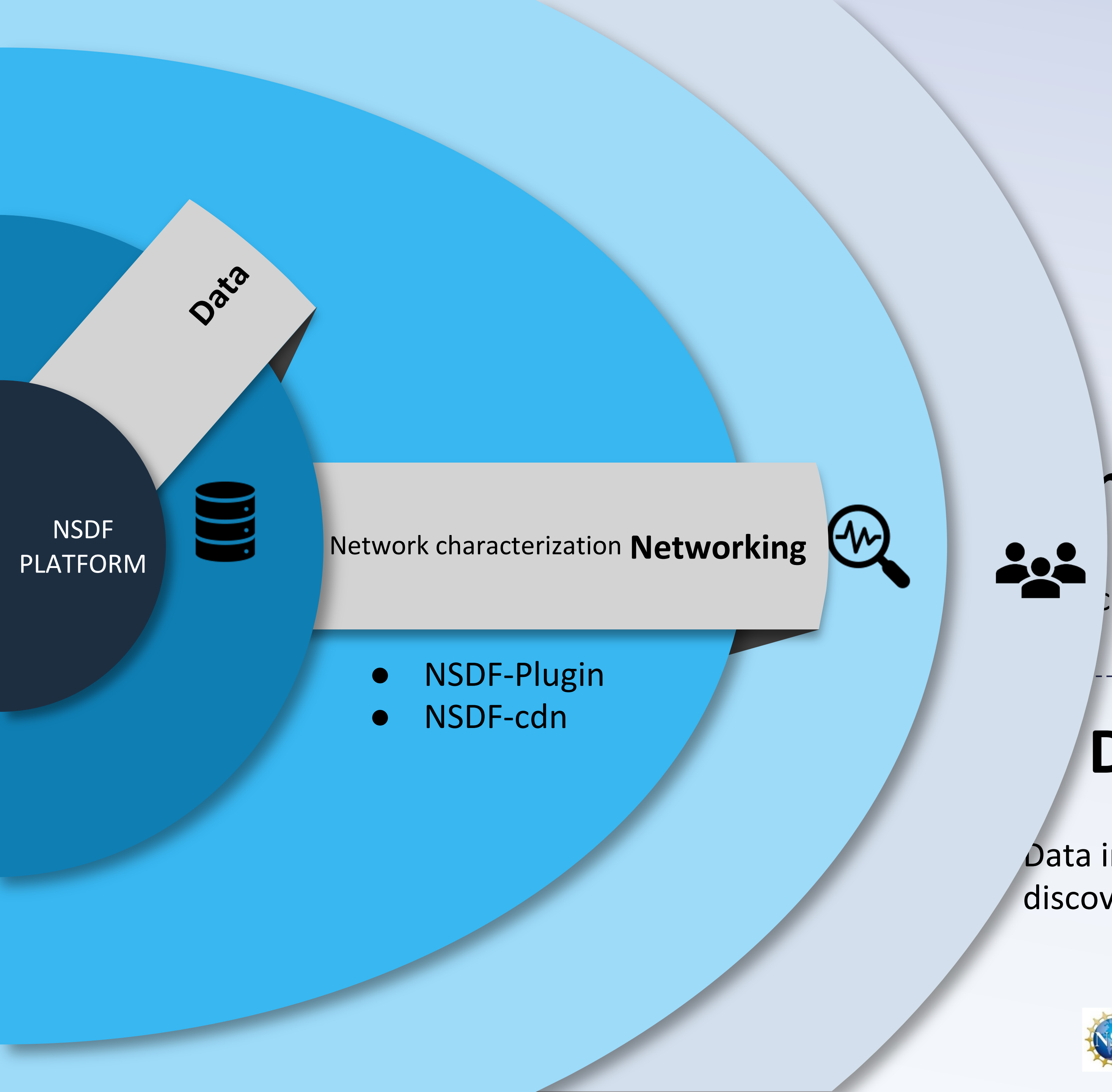
Users and commercial partners

Data

Data inventory and discovery

Networking

Network characterization



Services & Building Blocks

...nitoring

...ce monitoring

Partnership

Users and commercial partners

Data

Data inventory and discovery

Networking

Network characterization

Services & Building Blocks

Data
Storage, transfer, visualization

NSDF PLATFORM

- NSDF-Catalog
- NSDF-Fuse
- NSDF-Stream
- **NSDF-OpenVisus**

nitoring

ce monitoring

Partnership

Users and commercial partners

Data

Data inventory and discovery

Networking

Network characterization

Tutorial Goals

Tutorial

This tutorial demonstrates end-to-end analysis of scientific data through NSDF services

Tutorial Goals

- **Construct a modular workflow** that combines your application components with NSDF services
- **Upload, download, and stream data** to and from **public and private storage** solutions
- Deploy the NSDF dashboard for large-scale **data access, visualization,** and **analysis**

Four-Step Workflow Tutorial

Tutorial

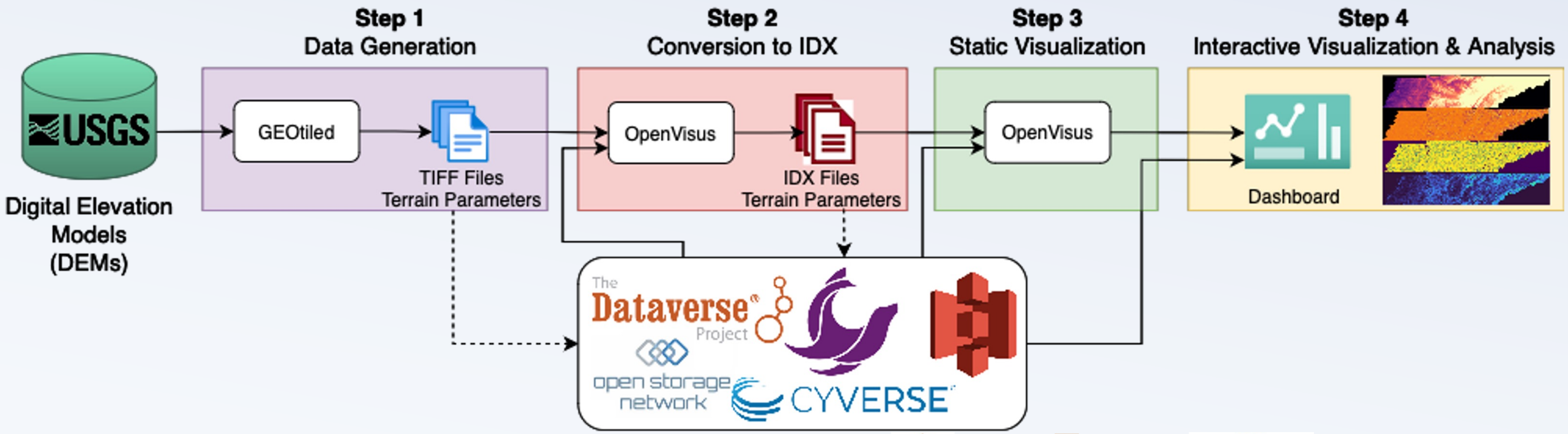
This tutorial showcases the capabilities of NSDF, guiding you through a **four-step modular workflow** that leverages OpenVisus services to analyze a geospatial dataset generated with GEOtiled.

Step 1: Data Generation
Collect DEMs from the United States Geological Survey (USGS). **Process** them with GEOtiled or upload the data from public or private storage.

Step 2: Conversion to IDX Format
Convert files from TIFF to IDX (the format used by OpenVisus), preserving accuracy but reducing size. **Store** IDX files in public or private storage.

Step 3: Static Visualization
Statically visualize the terrain parameters in OpenVisus. **Validate** accuracy of IDX-based images with the TIFF-based images.

Step 4: Interactive Visualization & Analysis
Launch dashboard for interacting with large-scale data to access subregions of the original dataset for ad hoc analysis



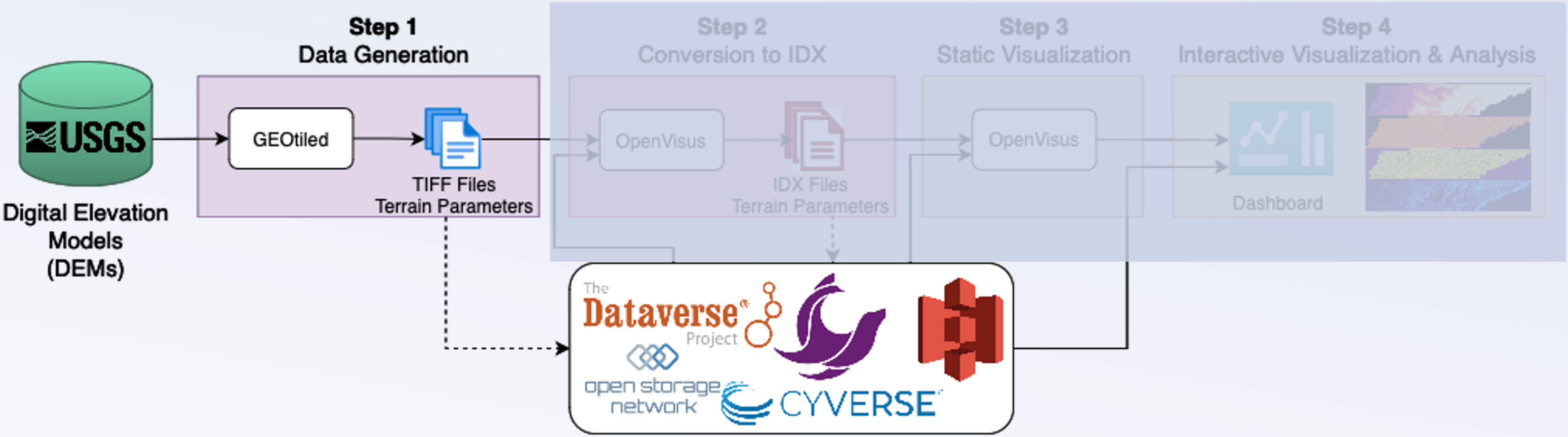
Step 1: Data Generation with GEOtiled

Step 1: Data Generation
Collect DEMs from the United States Geological Survey (USGS) and process them with GEOtiled or upload the data from public or private storage.

Step 1 provides **two options** to obtain data and generate the TIFF files before proceeding with Step 2

Option A
Generating Data Using the SOMOSPIE Application Module

→Option B
Accessing data from Dataverse public commons

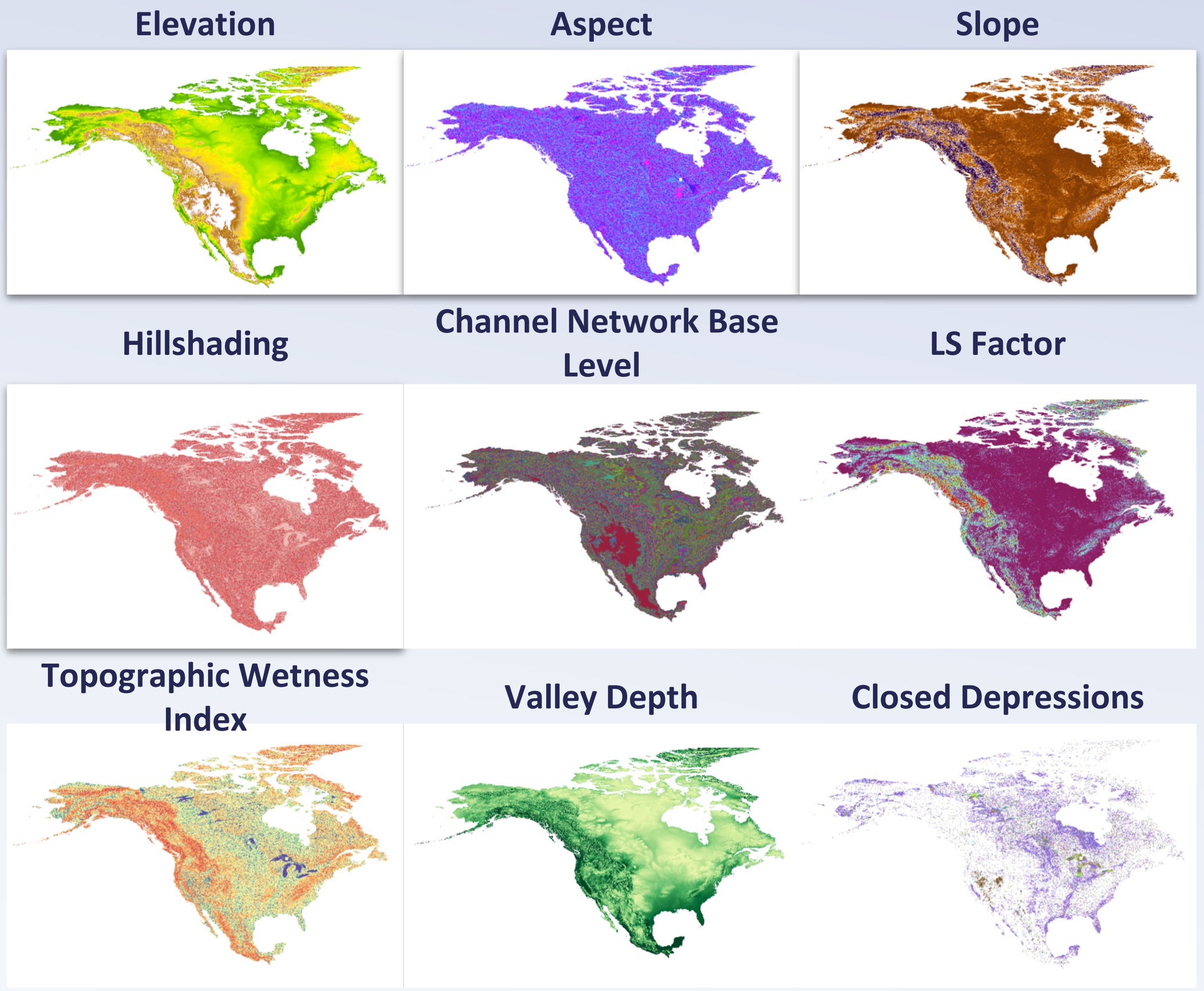


Step 1: What are Terrain Parameters?

Terrain parameters (e.g., slope, aspect, hillshading, etc.) are **descriptions of surface form derived from Digital Elevation Models (DEM).**

They play a **fundamental role** in applications such as **precision forestry and agriculture, and hydrology for landscape ecology.**

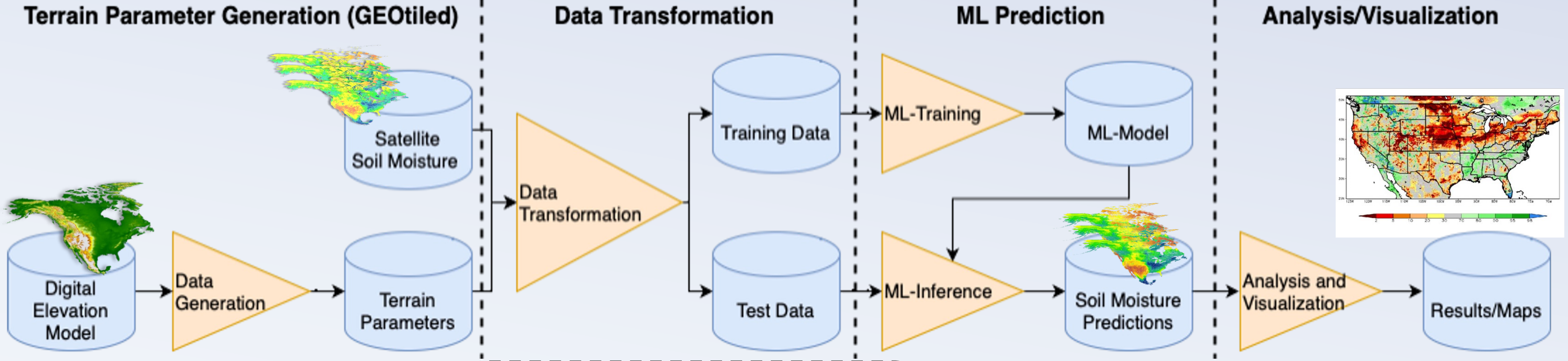
Generating **terrain parameters** at **high-resolution is computationally expensive,** hindering their accessibility by the scientific community



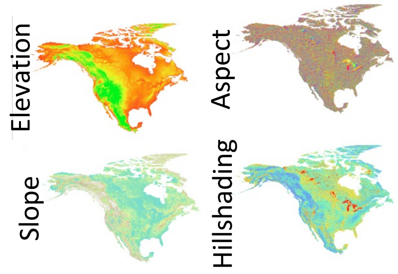
Step 1: SOMOSPIE Components

SOMOSπ

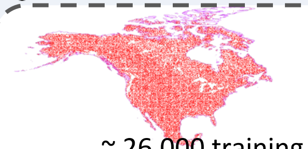
SOMOSPIE (SOil MOisture SPatial Inference Engine) has **four components** that empower scientists to generate, predict, and analyze **high-resolution topographic data**



Terrain Parameters



Training Data

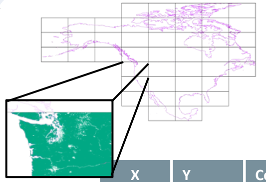


~ 26,000 training points

Sample points every ~27 km

X	Y	Soil Moisture	Covariate 1	Covariate 2	Covariate 3	Covariate... 4
X 1	Y 1	SM 1	A 1	B 1	C 1	N 1
X 2	Y 2	SM 2	A 2	B 2	C 2	N 2
X 3	Y 3	SM 3	A 3	B 3	C 3	N 3
X 4	Y 4	SM 4	A 4	B 4	C 4	N 4
X 5	Y 5	SM 5	A 5	B 5	C 5	N 5
X... N	Y... N	SM... N	A... N	B... N	C... N	N... N

Test Data

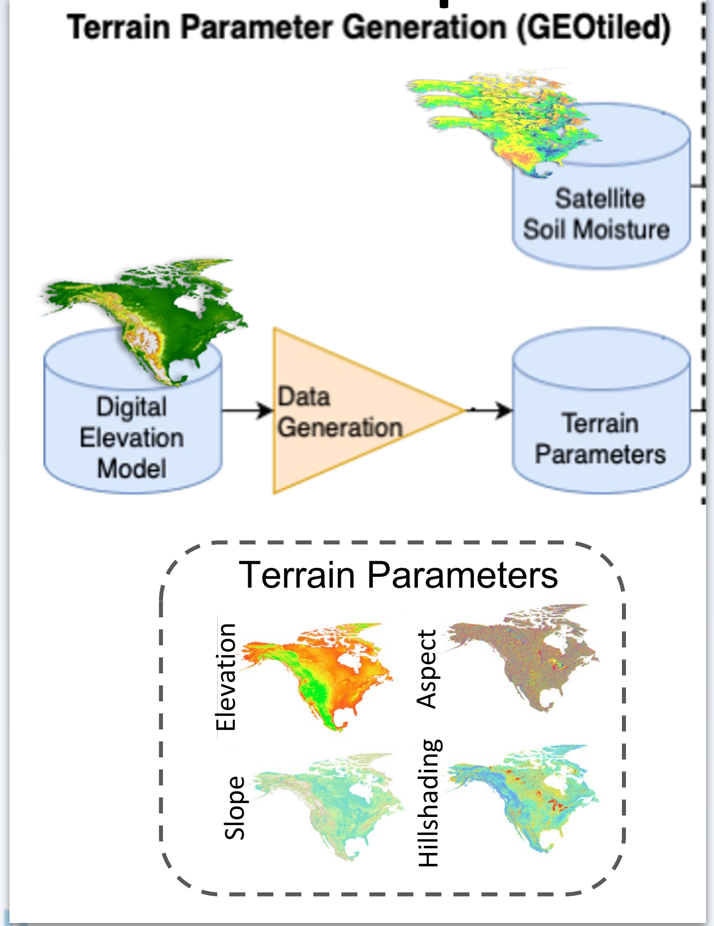


Sample points every X meters

X	Y	Covariate 1	Covariate 2	Covariate 3	Covariate ... 4
X 1	Y 1	A 1	B 1	C 1	N 1
X 2	Y 2	A 2	B 2	C 2	N 2
X 3	Y 3	A 3	B 3	C 3	N 3
X 4	Y 4	A 4	B 4	C 4	N 4
X 5	Y 5	A 5	B 5	C 5	N 5
X... N	Y... N	A... N	B... N	C... N	N... N

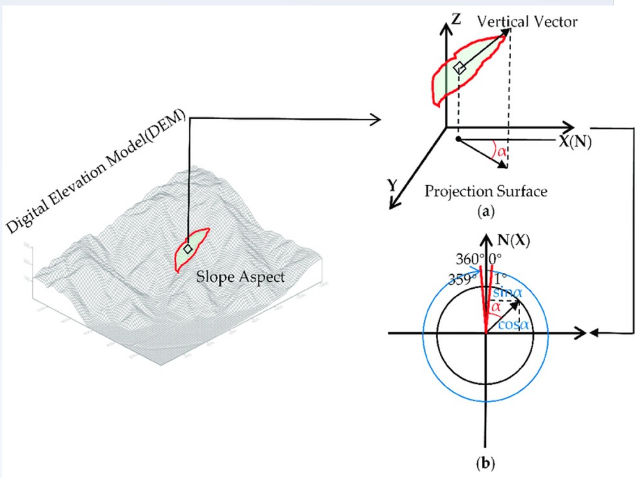
Step1: GEOtiled Terrain Generation

We expand on the first component, **GEOtiled, that computes high-resolution terrain parameters** using Digital Elevation Models (DEMs)

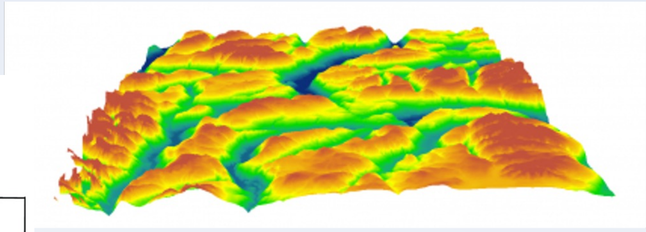


GEOtiled leverages data partitioning to accelerate the computation of terrain parameters from DEMs while preserving accuracy

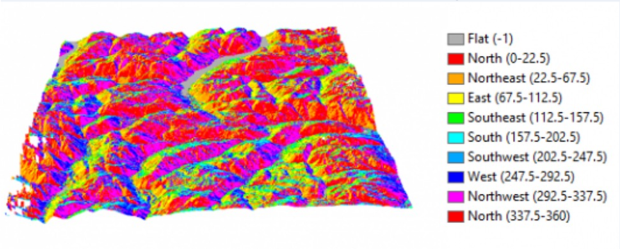
Computing High-resolution Terrain Parameters



Digital Elevation Model (DEM)



Terrain Parameter 1: Aspect

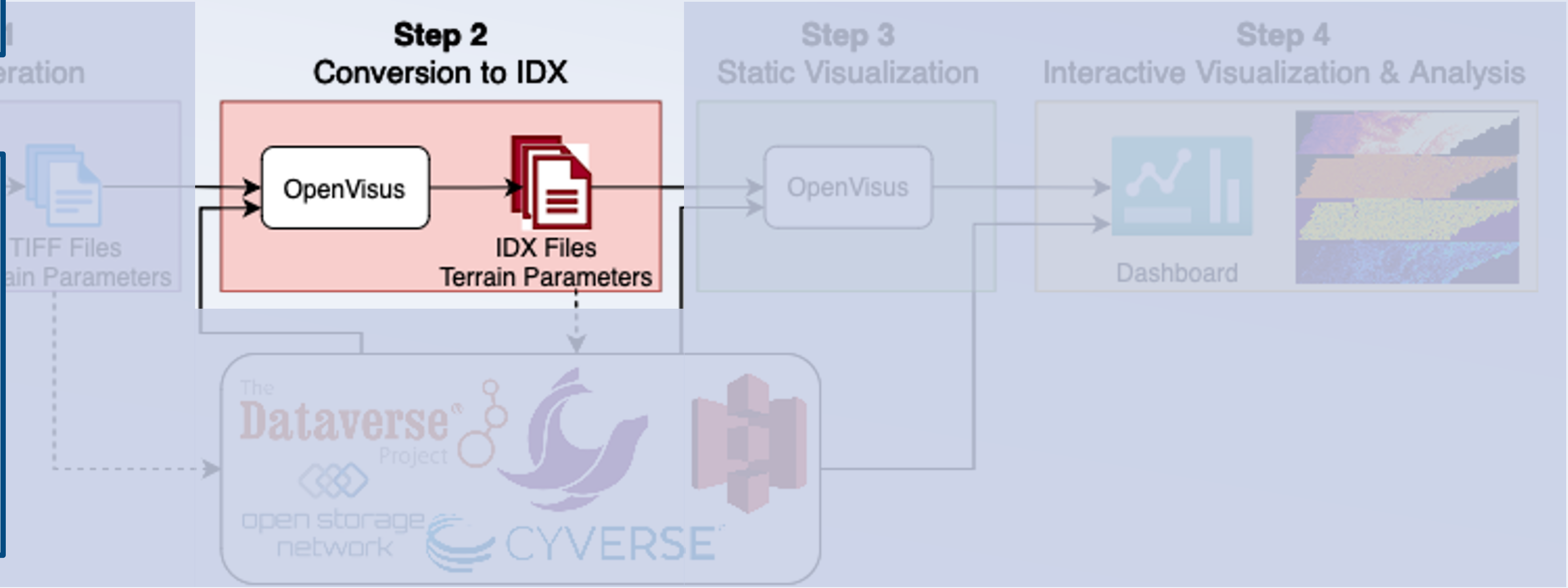


Step 2: Conversion to IDX

OpenVisus is a progressive cache-oblivious framework for large-scale data visualization

Step 2: Conversion to IDX Format
Convert files from TIFF to IDX (the format used by **OpenVisus**), preserving accuracy but reducing size. Store **IDX** files in public or private storage.

Converting **to IDX** from TIFF format **reduces file size by 20%** while preserving accuracy



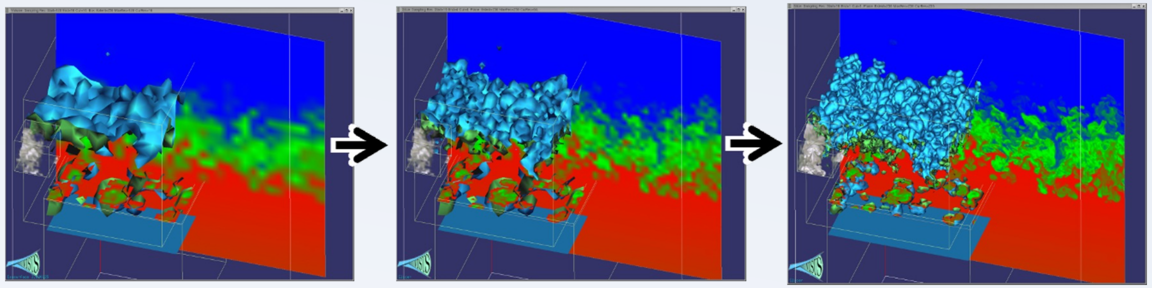
The **IDX** data format stores the data in a hierarchical Z (HZ) order, providing efficient, progressive access to large-scale scientific datasets

Step 2: IDX Data Format

Why IDX?

- The IDX data format provides **efficient, cache-oblivious, and progressive access** to large-scale scientific datasets.
- Data stored in IDX format can be visualized in an **interactive environment** allowing for meaningful explorations with **minimal resources.**
- IDX provides **scalability** across a wide range of running conditions like personal computers to distributed systems.

- Conversion to IDX is **not limited** to TIFF; it will work on other data formats like **NetCDF, HDF5, RGB, raw/binary,** and so on.
- IDX supports industry-standard lossless and lossy compression algorithms such as zlib, zfp, lz4.



Fine Resolution →

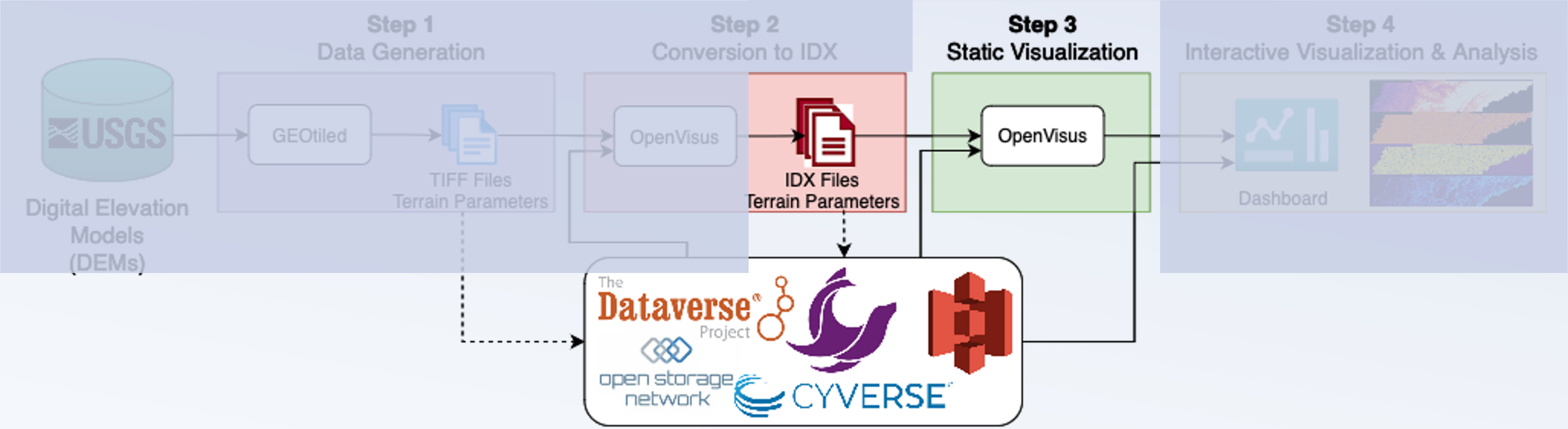
Step 3: Static Visualization

Step 3 provides **two options** to obtain data and collect the IDX files

→Option A
From local storage

Option B
From Seal Storage

Step 3: Static Visualization
Statically visualize the terrain parameters in OpenVisus. Validate the accuracy of IDX-based images with TIFF-based images.



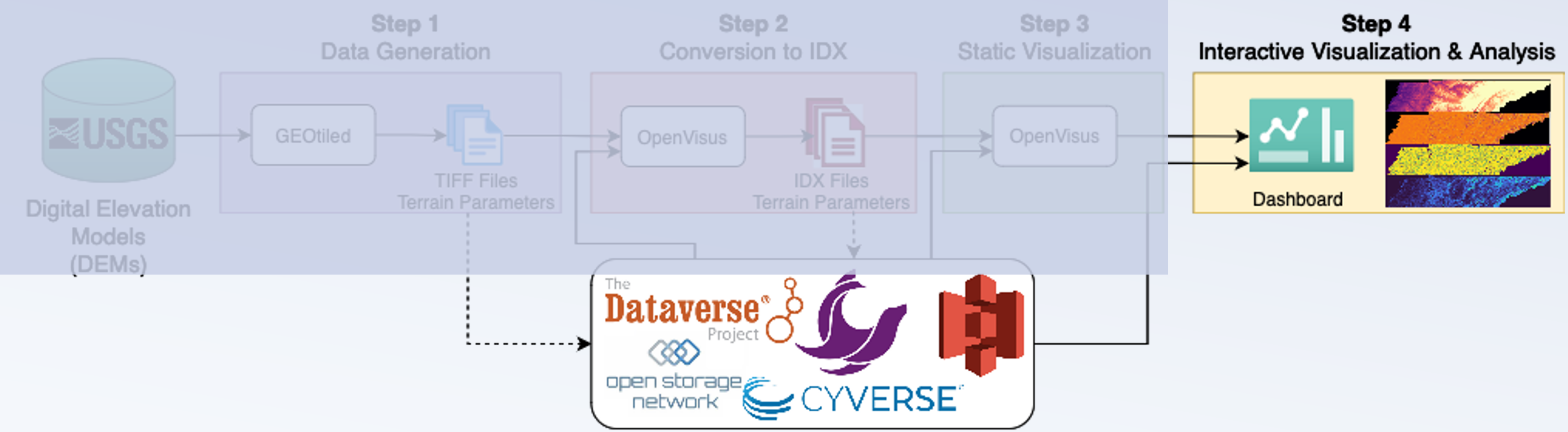
Step 4: Interactive Visualization & Analysis

Remotely **access** large datasets, **zoom** into specific areas, **select** and **crop** subregions of interest, **save** data locally in a Python-compatible format, and **analyze** the data for scientific discovery.

Step 4: Interactive Visualization & Analysis
Launch dashboard for interacting with large-scale data to access subregions of the original dataset for ad hoc analysis.

Step 4 provides **two options** to obtain data and collect the IDX files

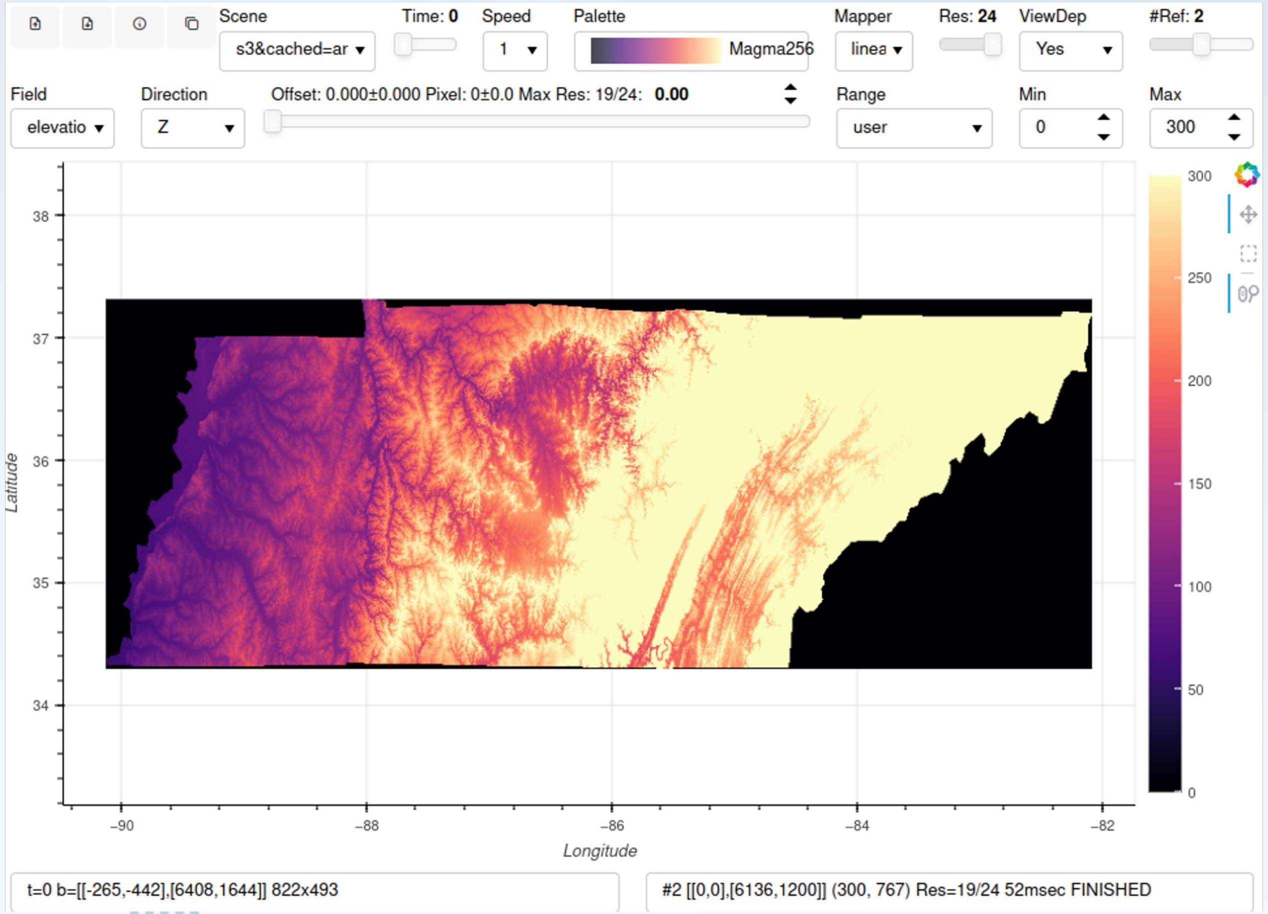
Option A	**Option B**
From local storage	From Seal Storage



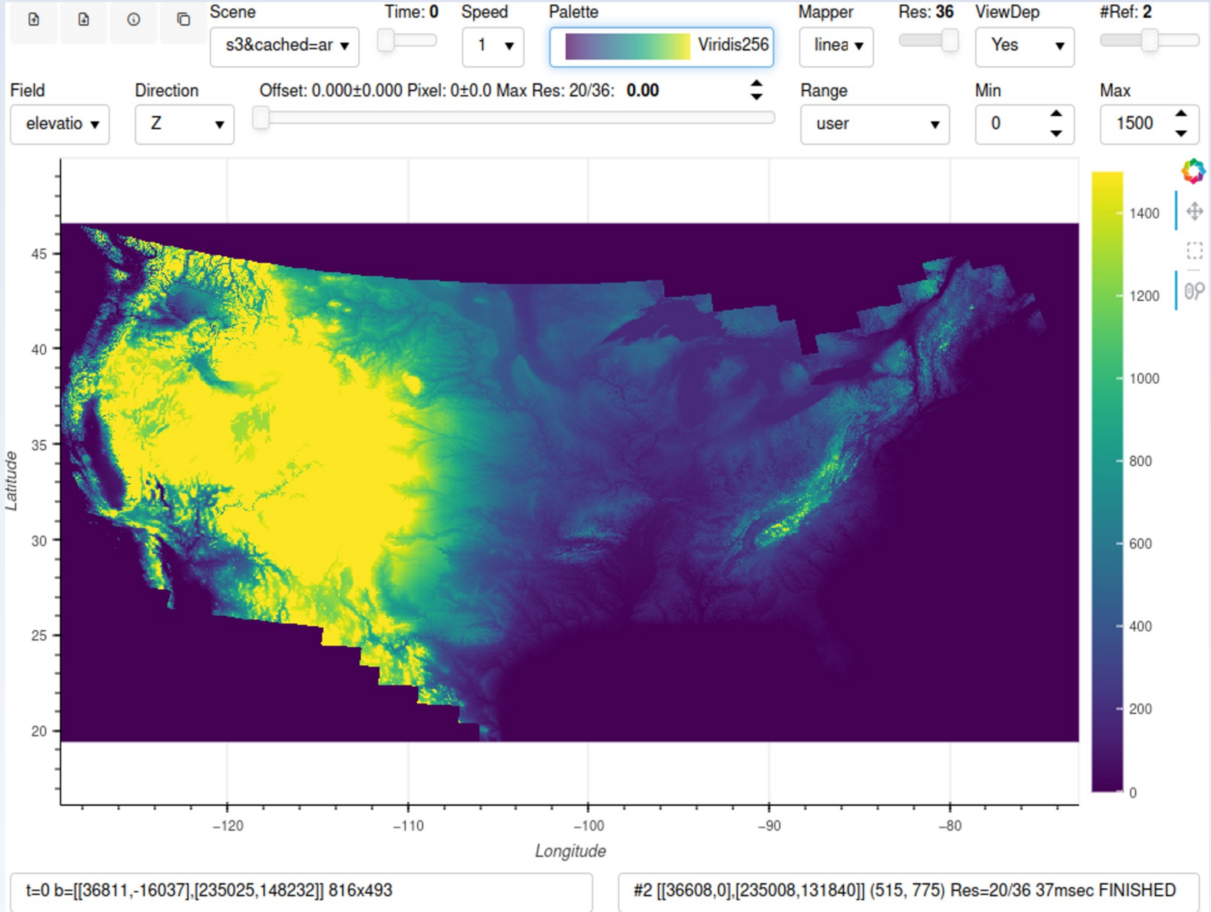
Step 4: Geographical Regions

Visualize and analyze two geographical regions 30 m resolution

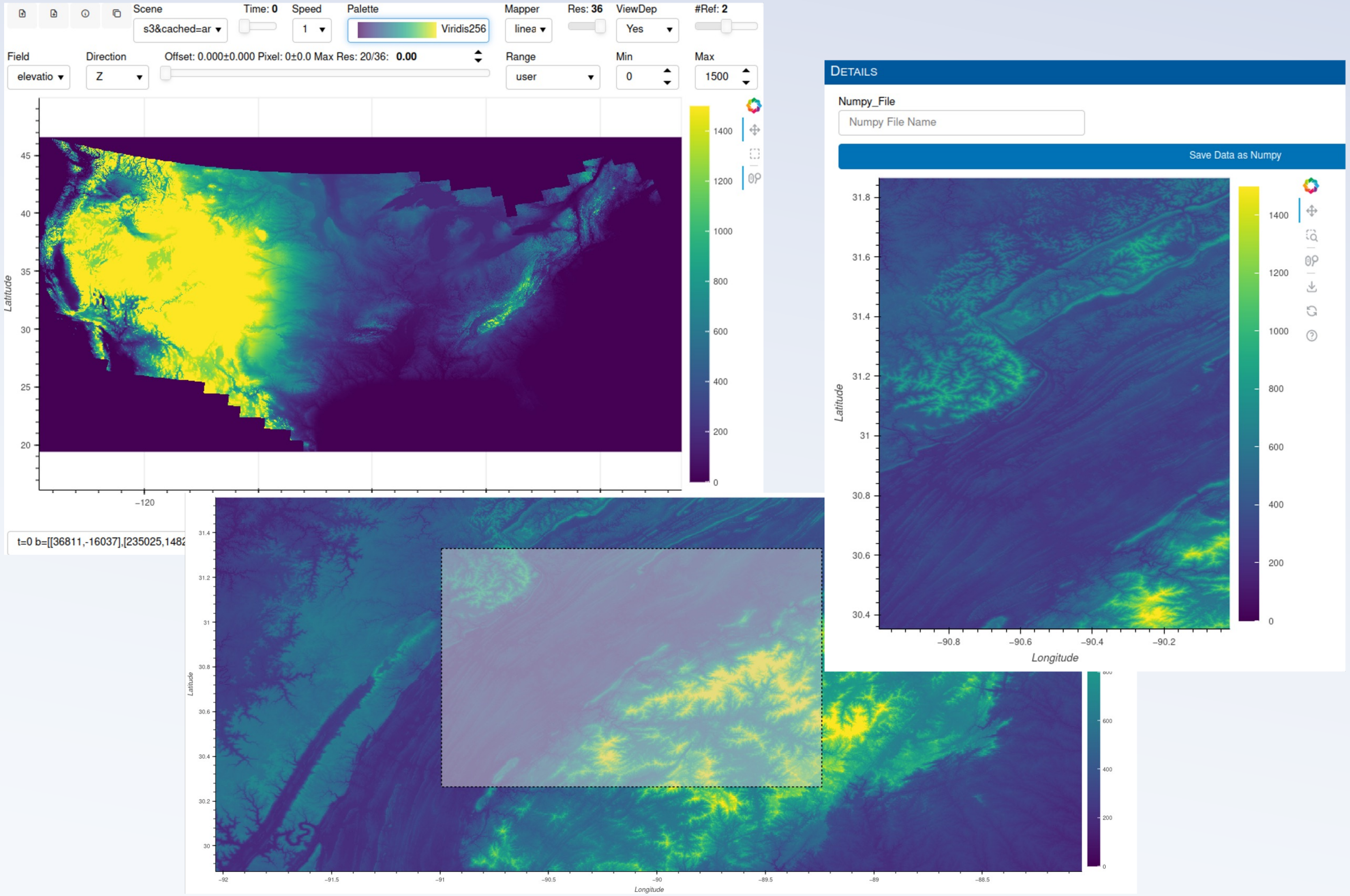
State of Tennessee



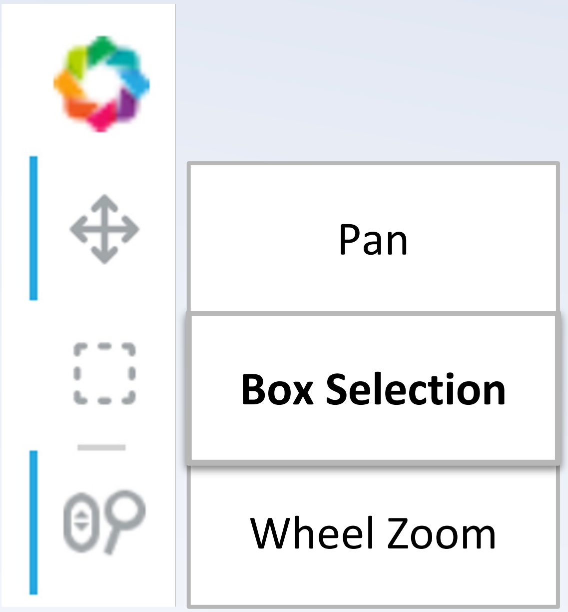
Contiguous United States (CONUS)



Step 4: Analysis of Large-Scale Subregions



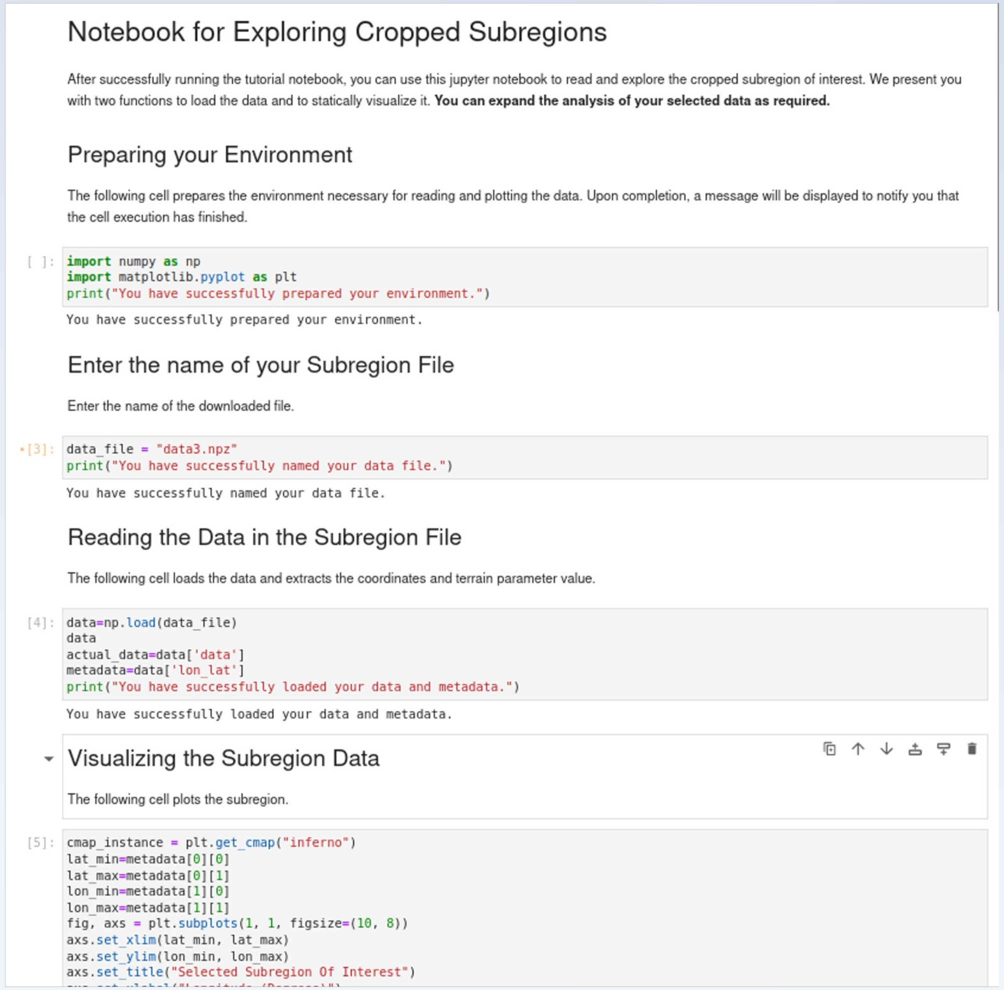
- Visualize large-scale data
- Select and explore subregions
- Work on only subregion of interest locally



Bonus Material: Exploring your Subregion Data

We provide an extra jupyter notebook **Explore_Data.ipynb** to:

- Load the downloaded subregion of interest in your local machine
- Extract geospatial information from the metadata
- Perform additional analysis on the subregion of interest



Notebook for Exploring Cropped Subregions

After successfully running the tutorial notebook, you can use this jupyter notebook to read and explore the cropped subregion of interest. We present you with two functions to load the data and to statically visualize it. **You can expand the analysis of your selected data as required.**

Preparing your Environment

The following cell prepares the environment necessary for reading and plotting the data. Upon completion, a message will be displayed to notify you that the cell execution has finished.

```
[ ]: import numpy as np
     import matplotlib.pyplot as plt
     print("You have successfully prepared your environment.")

     You have successfully prepared your environment.
```

Enter the name of your Subregion File

Enter the name of the downloaded file.

```
[3]: data_file = "data3.npz"
     print("You have successfully named your data file.")

     You have successfully named your data file.
```

Reading the Data in the Subregion File

The following cell loads the data and extracts the coordinates and terrain parameter value.

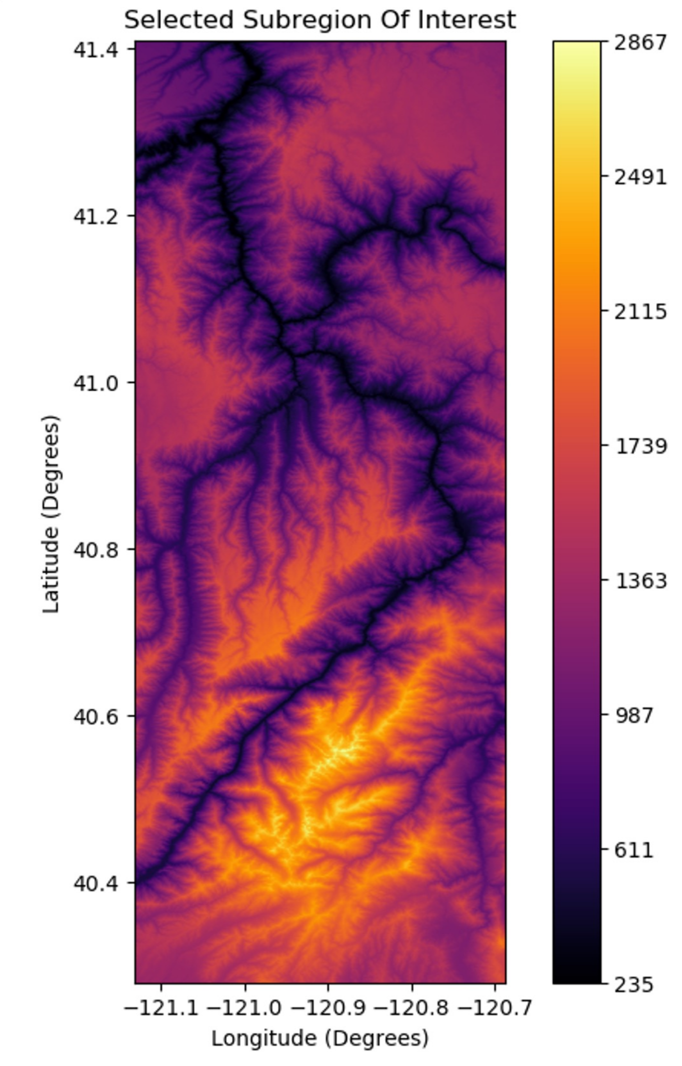
```
[4]: data=np.load(data_file)
     data
     actual_data=data['data']
     metadata=data['lon_lat']
     print("You have successfully loaded your data and metadata.")

     You have successfully loaded your data and metadata.
```

Visualizing the Subregion Data

The following cell plots the subregion.

```
[5]: cmap_instance = plt.get_cmap("inferno")
     lat_min=metadata[0][0]
     lat_max=metadata[0][1]
     lon_min=metadata[1][0]
     lon_max=metadata[1][1]
     fig, axs = plt.subplots(1, 1, figsize=(10, 8))
     axs.set_xlim(lat_min, lat_max)
     axs.set_ylim(lon_min, lon_max)
     axs.set_title("Selected Subregion Of Interest")
```



Discussion

Construct a modular workflow that combines your application components with NSDF services

Is your application modular? Can you leverage APIs?
Can your application take advantage of the NSDF services?

Upload, download, and stream data to and from **public and private storage** solutions

How large is your data? How do you access, share, and store your data?
Can your data take advantage of private and public storage?

Deploy the NSDF dashboard for large-scale **data access, visualization,** and **analysis**

What type of analysis do you perform on your data?
Can your research take advantage of an interactive dashboard?

Survey

Share with us your thoughts! (Up to 3 mins)

Relevant Links for Tutorial

NSDF-Tutorial

GEOtiled

SOMOSPIE

OpenVisus

Check Other OpenVisus Dashboards

QR1: NASA 200TB Ocean Dataset Use-case

QR2: CHESS sample Use-case

QR3: Material Science Use-case

QR4: Bellows Use-case