UNIVERSIDAD DE CANTABRIA
INSTITUTO DE FÍSICA DE CANTABRIA (IFCA), CSIC-UC

The Next Generation of Scalable and Interactive Data Analysis and Storage Infrastructures in Support of Climate Research

La Siguiente Generación de Infraestructuras de Almacenamiento y Análisis de Datos Escalables e Interactivas en Apoyo de la Investigación Climática

Por:

Ezequiel Cimadevilla Álvarez

Bajo la supervisión de:
Antonio S. Cofiño González
Maialen Iturbide Martínez de Albéniz

Una tesis presentada para optar al título de Doctor por la
Universidad de Cantabria en el programa de
Doctorado en Ciencia y Tecnología

Septiembre 2025

El estático movimiento de la vida.

Álvaro L.

A databank without effective modes of access is merely a data graveyard.

Arthur M. Lesk, Introduction to Bioinformatics

My most productive single act as an IBM manager had nothing to do with product development. It was sending a promising engineer to go as a full-time IBM employee in mid-career to the University of Michigan to get a PhD. This action . . . had a payoff for IBM beyond my wildest dreams.

Fred Brooks on E. F. Codd, father of relational databases

Olvidósele a Virgilio de declararnos quién fue el primero que tuvo catarro en el mundo, y el primero que tomó las unciones para curarse del morbo gálico, y yo lo declaro al pie de la letra, y lo autorizo con más de veinte y cinco autores, porque vea vuesa merced si he trabajado bien y si ha de ser útil el tal libro a todo el mundo.

El primo del licenciado, en Don Quijote de la Mancha de Cervantes

Acknowledgements

This thesis is dedicated to my family. I would like to thank my supervisors, Antonio and Maialen, as well as José Manuel, for the time, effort, and contributions they have made throughout this work. I also extend my gratitude to the other members of the Climate and Data Mining group, the Institute of Physics of Cantabria, and the Faculty of Sciences of University of Cantabria for the support provided during the development of this thesis.

Esta tesis está dedicada a mi familia. Agradezco a mis supervisores, Antonio y Maialen, también a José Manuel, por el tiempo, el esfuerzo y las aportaciones que han realizado a lo largo de este trabajo. Extiendo también mi agradecimiento al resto de miembros del grupo de Clima y Minería de Datos, del Instituto de Física de Cantabria y de la Facultad de Ciencias de la Universidad de Cantabria por el apoyo brindado durante el desarrollo de esta tesis.

Contents

Chapter 1

Resumen en Español

1.1 Contexto

Los datos climáticos son pilares fundamentales en las ciencias de la Tierra, ya que proporcionan la base necesaria para comprender y predecir las dinámicas atmosféricas y las tendencias del clima a largo plazo. A diferencia de los datos meteorológicos, que reflejan condiciones a corto plazo, los datos climáticos ofrecen una visión más amplia al capturar patrones sostenidos a lo largo de décadas o incluso siglos. Estos conjuntos de datos revelan la compleja interacción entre factores naturales y actividades humanas que influyen en el clima del planeta. Gracias a ellos, los científicos pueden identificar tendencias, evaluar cambios y diseñar estrategias de mitigación y adaptación frente a los desafíos ambientales globales actuales.

La era del Big Data ha abierto un abanico de oportunidades y desafíos en el campo de las ciencias de la Tierra. En este contexto, el Big Data plantea dos tipos principales de problemas: uno de carácter epistemológico y otro de índole tecnológica. El problema epistemológico se refiere a la medida en que la disponibilidad masiva de datos climáticos — provenientes de diversas fuentes, aunque predominantemente de modelos climáticos — permite obtener nuevo conocimiento sobre la dinámica del sistema climático y su estudio. Por otro lado, el problema tecnológico alude a las capacidades técnicas disponibles para enfrentar los retos que impone el Big Data. Esto incluye la eficiencia, escalabilidad y usabilidad de los sistemas e infraestructuras empleados en la ciencia de datos del clima. Esta tesis se centra en abordar el problema tecnológico dentro del marco de la ciencia de datos y el Big Data aplicado al estudio del clima.

El volumen, la velocidad y la variedad de datos generados a partir de modelos climáticos suponen un desafío tanto científico como técnico en el campo de la administración de datos meteorológicos y climáticos (*climate data management*). A pesar de esta abundancia de información, existe una ausencia de sistemas robustos de apoyo a la toma de decisiones capaces de almacenar, procesar, analizar y traducir eficazmente estos datos en conocimiento. La complejidad de los datos climáticos exige herramientas y técnicas avanzadas de análisis de datos. Sin estos sistemas, el vasto potencial inherente a los datos climáticos permanece en gran medida sin aprovechar, lo que dificulta nuestra capacidad de tomar decisiones informadas en áreas críticas como la mitigación del cambio climático, la preparación ante desastres y la asignación de recursos.

Las prácticas y sistemas actuales utilizados en la gestión de datos meteorológicos y climáticos, pertenecientes al campo de ciencias de la Tierra, son incapaces de manejar la complejidad de este tipo de datos en la era del Big Data. Como resultado, dicha complejidad recae sobre los usuarios de estos sistemas, imponiendo elevados costos y dificultades que representan una barrera para quienes desean analizar datos meteorológicos y climáticos con el fin de convertir esta información en conocimiento útil. Cerrar esta brecha mediante el desarrollo de sistemas innovadores de apoyo a la toma de decisiones, diseñados específicamente para afrontar los desafíos únicos que plantea el Big Data climático, es fundamental para aprovechar al máximo esta información y abordar los problemas de nuestra época relacionados con el clima.

1.2 Objetivos

Esta tesis se centra en el problema y la solución tecnológica de la ciencia de datos climáticos, reconociendo al mismo tiempo el problema epistemológico como el objetivo final, siendo la tecnología un medio para abordarlo de forma eficaz. Consideramos tres dimensiones de la solución tecnológica en la ciencia de datos climáticos: el almacenamiento de datos climáticos, el análisis de datos climáticos y las infraestructuras de datos climáticos.

El almacenamiento de datos climáticos hace referencia a los desafíos asociados al almacenamiento y la organización de estos datos en sistemas de información. La investigación sobre el almacenamiento de datos estudia tanto los fundamentos teóricos como las implementaciones prácticas de los sistemas que permiten organizar, estructurar y acceder a la información. Esto incluye el desarrollo de modelos que definen cómo se estructura y se relaciona la información. Además, se investigan temas como el almacenamiento distribuido, la replicación de datos, los modelos de consistencia y la tolerancia a fallos, todos ellos fundamentales para garantizar la disponibilidad y fiabilidad de la información en entornos complejos.

El análisis de datos climáticos es el proceso de estudio, interpretación y visualización de datos relacionados con el clima, con el objetivo de comprender patrones, tendencias y anomalías en el sistema climático terrestre. Este proceso implica realizar inferencias lógicas y estadísticas a partir de conjuntos de datos climáticos. La inferencia lógica y la inferencia estadística difieren fundamentalmente tanto en sus objetivos como en sus métodos para derivar conclusiones epistemológicas a partir de los datos, aunque suelen aparecer conjuntamente en el análisis de datos climáticos. Por ejemplo, combinar registros de temperaturas superficiales en rejilla que se han dividido en varios ficheros a lo largo de la coordenada temporal requiere la composición lógica de la serie temporal en una única unidad lógica, una operación de inferencia lógica. En cambio, calcular la media espacial de esa serie temporal lógica en las dimensiones espaciales implica inferencia estadística. Esta tesis mostrará una inclinación hacia las operaciones de inferencia lógica en el contexto de la ciencia de datos climáticos, más que hacia la inferencia estadística. Otro aspecto clave del análisis de datos es el procesamiento y la optimización de consultas, lo que implica diseñar algoritmos capaces de ejecutar operaciones de búsqueda y análisis de manera eficiente. En este campo se estudian técnicas para generar planes de ejecución óptimos y adaptarlos dinámicamente según las características de los datos y las condiciones del sistema.

Tanto el almacenamiento como el análisis de datos climáticos se lleva a cabo en complejas infraestructuras, tanto en términos de hardware como de software, que permiten la explotación de datos climáticos y la adquisición de conocimiento sobre el sistema climático. Estas infraestructuras pueden variar desde estaciones de trabajo personales hasta infraestructuras de computación de altas prestaciones (HPC) e infraestructuras cloud. La elección de una u otra infraestructura condiciona todo el proceso de almacenamiento y análisis de datos climáticos, dando lugar a diferentes soluciones tecnológicas cuyos beneficios y limitaciones deben ser apropiadamente entendidos para poder lograr un uso eficiente de dichas infraestructuras. Dichas infraestructuras permiten a investigadores y científicos analizar grandes volúmenes de datos climáticos con mayor o menor eficiencia, dependiendo de los servicios que ofrecen. La gestión de datos climáticos se refiere a los procesos de organización y mantenimiento de los datos dentro de las infraestructuras climáticas, con el fin de garantizar su precisión, accesibilidad y utilidad a largo plazo.

Considerando estas tres dimensiones de ciencia de datos climáticos, esta tesis se plantea los siguientes objetivos:

1. **Caracterizar los datos climáticos y derivar sus requisitos de almacenamiento y análisis.** Comprender la naturaleza y los requisitos de almacenamiento y análisis de los datos climáticos generados por los Modelos Climáticos Globales. Asimismo, examinar las características de los datos densos y dispersos y

sus implicaciones para el almacenamiento, análisis y gestión de datos climáticos. Además, este trabajo tiene como objetivo derivar requisitos de almacenamiento/distribución y metadatos que apoyen operaciones comunes de corte, agregación multidimensional y construcción de series temporales.

2. **Mejorar las infraestructuras de datos climáticos.** Este trabajo diferencia entre infraestructuras de datos climáticos que solo proporcionan almacenamiento de datos y aquellas que ofrecen capacidades de análisis de datos climáticos sobre los datos disponibles. Se introduce el concepto de **laboratorio de datos** (o **DataLab**), tomado de las ciencias experimentales, como el lugar donde se lleva a cabo la investigación con datos climáticos, abarcando no solo el almacenamiento de datos, sino también las capacidades de cómputo. Se presentan los elementos e instrumentos que conforman el laboratorio de datos y se discute cómo facilitan y favorecen las tareas de análisis de datos climáticos. Como componente indispensable del laboratorio de datos, los flujos de trabajo alineados con FAIR (basados en cuadernos de programación en R y Python) serán un requisito fundamental.

3. **Mejorar el análisis de datos climáticos.** Esta tesis describe y analiza los principales métodos de inferencia lógica y estadística presentes en el análisis de datos climáticos. Se propone el uso de **agregaciones virtuales** para generar unidades lógicas de análisis, con el fin de simplificar de manera significativa la experiencia de análisis de datos climáticos, eliminando de los usuarios y aplicaciones muchas tareas complejas de análisis lógico. También se presenta una taxonomía basada en criterios cualitativos que considera las diferentes formas en que actualmente se lleva a cabo el análisis de datos climáticos, según cómo se accede a los datos — ya sea descargando archivos, accediendo a fuentes de datos remotas o realizando cálculos en servidores remotos.

4. **Evaluar el rendimiento de las infraestructuras.** Las tecnologías propuestas en esta tesis, dedicadas al almacenamiento y análisis de datos climáticos junto con sus infraestructuras asociadas, se evalúan en términos de rendimiento. Se presta especial atención a la latencia y el rendimiento (throughput), que proporcionan medidas objetivas de eficiencia y permiten la comparación científica entre las soluciones propuestas. Se realizan evaluaciones de rendimiento para valorar la viabilidad de los laboratorios de datos presentados y de los métodos para generar agregaciones virtuales.

1.3 Principales Resultados y Conclusiones

Esta tesis ha examinado la convergencia entre la ciencia de datos climáticos y el desafío tecnológico que plantea el Big Data, con un énfasis particular en cómo las infraestructuras, herramientas y tecnologías de datos modernas pueden mejorar el almacenamiento, análisis, accesibilidad y reproducibilidad de los datos climáticos. La investigación resalta el potencial transformador de los enfoques centrados en los datos para abordar la creciente complejidad y escala de los conjuntos de datos climáticos. Esta tesis ofrece una visión integral del estado actual de la gestión de datos climáticos en tres dimensiones clave: almacenamiento de datos climáticos, análisis de datos climáticos e infraestructuras de datos climáticos. El panorama tecnológico existente se centra predominantemente en la descarga de archivos, un modelo que presenta varias limitaciones. Estas limitaciones dificultan el potencial de la ciencia de datos climáticos al introducir complejidad innecesaria en los flujos de trabajo analíticos. Como resultado, los analistas e investigadores de datos climáticos se ven obligados a gestionar estos desafíos, lo que a su vez restringe las capacidades que pueden ofrecer las infraestructuras de datos climáticos.

Para abordar estas cuestiones, este trabajo se ha centrado en la introducción de nuevos conceptos e identificación de limitaciones existentes. En el centro de este enfoque se encuentra el concepto de *laboratorio de datos climáticos*. Se definió un laboratorio de datos como un entorno equipado con las herramientas e infraestructuras necesarias para posibilitar una ciencia de datos eficiente. Inspirándose en las ciencias experimentales, donde los laboratorios cuentan con instrumentos especializados como microscopios, espectrofotómetros y centrífugas, el laboratorio de datos se ha conceptualizado como su equivalente digital. En este contexto, los instrumentos son más abstractos que sus contrapartes experimentales. Los laboratorios de ciencia de datos incluyen tanto infraestructura de hardware como herramientas de software que apoyan diversos aspectos de la gestión de datos climáticos, como formatos de almacenamiento, librerías y aplicaciones de análisis.

Esta tesis ha utilizado el concepto de *Analysis Ready Data* (ARD) como instrumento para ofrecer a los usuarios una representación de los datos a un nivel lógico superior, reduciendo la complejidad de los scripts que deben manejar manipulaciones lógicas de los datos climáticos, como la localización de archivos y la gestión de sistemas de ficheros. Al proporcionar agregaciones lógicas, los usuarios pueden trabajar con los datos sin preocuparse por detalles de implementación, realizando operaciones lógicas y estadísticas sobre las dimensiones de estos cubos de datos, en lugar de operar directamente sobre ficheros individuales. Además, se han identificado tres modos de análisis de datos climáticos — descarga y análisis, acceso remoto a datos y computación junto a los datos — cada uno con sus ventajas y desventajas según el tipo de flujo de trabajo de análisis climático.

En el marco de esta tesis, tanto el laboratorio de datos como los ARD se han integrado en un único entorno, el *laboratorio de datos ESGF-VA*. El laboratorio de datos se presenta como el siguiente paso evolutivo en las infraestructuras de datos climáticos. A diferencia de los sistemas tradicionales, que se limitan principalmente a ofrecer descargas de archivos, las infraestructuras modernas deberían proporcionar servicios mejorados, como acceso remoto a datos y computación del lado del servidor. Estas capacidades mejoran significativamente la experiencia del usuario al aliviar cargas estructurales y técnicas, permitiendo que los investigadores se concentren en los aspectos centrales del análisis de datos climáticos.

En conjunto, las contribuciones de esta tesis se han adherido a los principios FAIR, asegurando que los datos climáticos sean localizables, accesibles, interoperables y reutilizables, lo que maximiza su utilidad en diferentes dominios de investigación y facilita la reproducibilidad de los análisis. Los laboratorios de datos presentados desempeñan un papel crucial en el avance de la reproducibilidad y en el soporte de los principios FAIR. Al proporcionar un entorno controlado e interactivo donde los conjuntos de datos, los flujos de trabajo analíticos y los recursos computacionales están claramente documentados, versionados y gestionados de manera consistente, los laboratorios de datos permiten a los investigadores replicar análisis de manera confiable y contribuir nuevos desarrollos a trabajos previos. En particular, estas infraestructuras son especialmente adecuadas para evaluaciones a gran escala y federadas, como las realizadas por el IPCC, donde múltiples investigadores deben coordinar análisis, compartir resultados y producir hallazgos consistentes y verificables. Al integrar la reproducibilidad y las prácticas compatibles con FAIR en el núcleo de los laboratorios de datos, la investigación climática puede alcanzar mayores niveles de transparencia, fiabilidad y utilidad a largo plazo.

1.4 Líneas de Trabajo Futuro

Dado que una tesis doctoral representa un trabajo acotado en el tiempo, ciertos aspectos complementarios relacionados con los datos climáticos quedarán fuera del alcance de este estudio, a pesar de su relevancia potencial.

Uno de estos aspectos es la transición desde sistemas de almacenamiento basados en ficheros hacia sistemas basados en bases de datos. Durante décadas, el campo del almacenamiento de datos ha desarrollado principios sólidos sobre los que se construyen los sistemas modernos, entre ellos la noción de *modelo de datos*, entendido como una combinación de estructuras de almacenamiento, reglas de manipulación y restricciones de integridad. Sin embargo, la comunidad dedicada al estudio del clima aún no ha incorporado plenamente estos avances, lo que ha limitado el desarrollo de infraestructuras más eficientes y usables. Actualmente, el almacenamiento y análisis de datos climáticos sigue dependiendo en gran medida de sistemas basados en ficheros, lo que representa una oportunidad clara para la evolución hacia enfoques más integrados.

Otro eje de investigación que se plantea como trabajo futuro es la gestión de datos climáticos dispersos, como los generados por estaciones meteorológicas. Aunque este tipo de datos no será abordado en profundidad, se hará referencia a ellos como punto de contraste para comprender mejor la naturaleza de los datos densos producidos por los modelos climáticos. A lo largo del trabajo se destacarán las principales fuentes actuales de información climática y meteorológica, donde se evidenciará que los modelos numéricos del clima y los sistemas de teledetección - como los satélites - constituyen las fuentes más abundantes. En cambio, los datos provenientes de estaciones meteorológicas, aunque cuantitativamente menores, desempeñan un papel fundamental en la calibración y validación de los modelos, aportando una representación más fiel de la realidad observada.

1.5 Publicaciones y Contribuciones

La investigación realizada en esta tesis ha dado lugar a varias contribuciones académicas, entre ellas la publicación de tres artículos en revistas revisadas por pares y la presentación de los resultados en congresos internacionales. Las principales aportaciones de este trabajo son las siguientes:

1. Cimadevilla E, Iturbide M, Cofiño AS, Fernández J, Sitz LE, Palacio A, et al. (2025). *The IPCC Interactive Atlas DataLab: Online reusability for regional climate change assessment.* PLOS Clim 4(6): e0000644. `https://doi.org/10.1371/journal.pclm.0000644`

2. Cimadevilla, E., Lawrence, B.N., & Cofiño, A.S. (2025). *The Earth System Grid Federation (ESGF) Virtual Aggregation (CMIP6 v20240125).* Geoscientific Model Development, 18(8), 2461–2478. `https://doi.org/10.5194/gmd-18-2461-2025`

3. Cimadevilla, E. (2025). *Why the relational data model matters for climate data management.* Computers & Geosciences, 201, 105931. `https://doi.org/10.1016/j.cageo.2025.105931`

Además, el trabajo realizado aquí también ha contribuido a otras publicaciones:

1. Hoz, A.P., et al. (2025). *DataLab as a Service: Distributed Computing Framework for Multi-Interactive Analysis Environments.* IEEE Access, 13, 22566–22577. `https://doi.org/10.1109/ACCESS.2025.3536637`

2. Iturbide, M., et al. (2022). *Implementation of FAIR principles in the IPCC: the WGI AR6 Atlas repository.* Scientific Data, 9(1), 629. `https://doi.org/10.1038/s41597-022-01739-y`

Otras contribuciones incluyen presentaciones orales y en formato póster:

1. Cimadevilla, E. and Cofiño, A.S. (2022) *Storage growth mitigation through data analysis ready climate datasets using HDF5 Virtual Datasets*, 28 March. Available at: `https://doi.org/10.5194/egusphere-egu22-7151`.
2. Cimadevilla, E., Iturbide, M. and Cofiño, A.S. (2023) *Virtual aggregations to improve scientific ETL and data analysis for datasets from the Earth System Grid Federation*, 15 May. Available at: `https://doi.org/10.5194/egusphere-egu23-16117`.
3. Cimadevilla, E. (2024) *A Science Gateway for climate data analysis based on Virtual Analysis Ready Data.* Proceedings of the 16th International Workshop on Science Gateways, Toulouse: Zenodo, 30 September. Available at: `https://doi.org/10.5281/zenodo.13863563`.

Chapter 2

Context, Objectives and Structure

2.1 Context

Climate research is facing an unprecedented transformation, driven both by the rapid evolution of data-intensive science and by the growing urgency of understanding and assessing climate change. The advancement of climate research and, in particular, the assessment of climate change, increasingly depends on the ability to store, access, manage, analyze, and share vast and complex climate datasets in a transparent and collaborative manner. However, significant technological challenges remain that hinder progress and limit the full potential of available scientific knowledge.

The accessibility and availability of appropriate infrastructures—from scalable storage systems to high-performance and interactive analysis environments—are still unevenly distributed across institutions and regions. The abundance, heterogeneity, and particularity of climate data present unique obstacles. Climate data differ not only in format and volume but also in temporal and spatial resolution, observational vs. simulated origins, and the disciplinary traditions that shape how they are produced and documented. Without robust mechanisms for harmonization, integration, and efficient access, these characteristics complicate scientific workflows and can hinder the reproducibility and reuse of results.

In addition, the reproducibility and reusability of climate analyses are central to scientific credibility and to the ability of assessments to inform decision making. Yet, current practices often lack the infrastructure and methodological support required to ensure that results can be systematically validated, shared, and extended.

2.1.1 Data-Intensive Science

Science has evolved through multiple paradigms, each shaping how we explore and understand the world. Experimental science focuses on direct observation and measurement to acquire empirical knowledge about natural phenomena, while theoretical science emphasizes structured theories that organize and allow formal reasoning. With the rise of digital computing in the twentieth century, computational science emerged as a third paradigm, using numerical models and large-scale simulations to study complex systems that cannot be fully captured by theory or experiment alone. During the past two decades, **data-intensive science** has emerged as the *fourth paradigm* of scientific discovery [1], leveraging vast datasets, machine learning and computational power to reveal patterns beyond the reach of purely theoretical or experimental workflows. The epistemic implications of data-intensive science remain actively debated in the *philosophy of science* [2–6].

Modern data-intensive science has been propelled by the **Big Data** phenomenon. It is estimated that humanity accumulated 180 EB of data between the invention of writing and 2006. Between 2006 and 2011, the total grew tenfold and reached 1,600 EB [4]. Thus, the Big Data phenomenon can be defined as the *data deluge* or overwhelming flood of data generated by modern digital technologies, scientific research, and everyday human activities [7]. This phenomenon is driven by advancements in computing, widespread sensor networks, social media, and large-scale scientific experiments. Even if *Big Data* is a recent buzzword, data has long been——and remains—a valuable resource for inquiry and decision-making [4].

It is considered that the data-intensive science paradigm possesses two new challenges to science [4]. First, the **epistemological problem** with Big Data lies in the fundamental question of how knowledge is generated, validated, and interpreted from data. The abundance of data has prompted dramatic claims that the data deluge the traditional scientific method is no longer adequate [6]. This raises concerns about the reliability, objectivity, and generalization of data-driven insights. Correlation is often mistaken for causation, biases in data and algorithms can reinforce existing inequalities, and models trained on incomplete or skewed data may produce misleading conclusions.

Second, it presumes a **technological solution** to an epistemological problem: more and better techniques, technologies and data infrastructures that will bring Big Data to a manageable scale [4]. This involves areas of research that provide the tools to manipulate data but do not necessarily address the deeper question of what constitutes valid and meaningful knowledge. Thus, while technological solutions push the boundaries of data storage and data processing, the epistemological problem demands critical scrutiny of the assumptions and limitations of data-driven reasoning.

2.1.2 Climate Data Science

The study of the Earth and its climate system has not been oblivious to the data-intensive science paradigm and the technological challenges it poses. The amount of both observational and model data has opened a new paradigm of studying and understanding the Earth. Historically, climate science relied on theoretical models, physical simulations, and observational studies. However, climate science has transitioned into a data-intensive discipline, leveraging vast datasets and advanced algorithms to improve predictions and understanding of climate dynamics. This phenomenon has emerged as a result of a data deluge driven by the increased use of computational climate models and the widespread availability of remote sensing networks. This work refers to data that belong to this scientific field as **climate data**.

One of the main contributors to the data deluge in climate science are Global Climate Models (GCMs). GCMs simulate the global dynamics of climate processes, including general circulation, by solving the physical equations that govern individual components and their interconnected relationships [8]. These equations are continuous in both space and time, but their analytical solutions are intractable due to the complexity and non-linearity of the system. To make the problem computationally tractable, climate models apply discretization, i.e. the transformation of continuous equations into a finite set of algebraic expressions that can be solved numerically on a computer. These models produce multi-decadal projections that are crucial for studying climate evolution under various radiative forcing or emission scenarios. Figure 2.1 shows a schematic illustration of the discretized simulation performed by GCMs at the grid-box level.

These factors enable climate scientists to generate data at scales that were once unimaginable [7, 10, 11]. The significant growth of data poses a scientific scalability challenge for the climate research community [12]. Contributions to the increase in data volume include the systematic increase in model resolution and the complexity of experimental protocols and data requests [13]. For example, the horizontal resolution of GCMs participating in the Coupled Model Intercomparison Project Phase 6 (CMIP6, [14]) ranges from 1° to 2° (approximately 100-200 kilometers). Currently, CMIP6 has produced more than 10 PB of unique data, which expands to approximately 20 PB when replicated across the research centers 5.2.1.

Impact and adaptation studies at the regional level require higher-resolution climate change data. Regional Climate Models (RCMs) are used in dynamical downscaling to

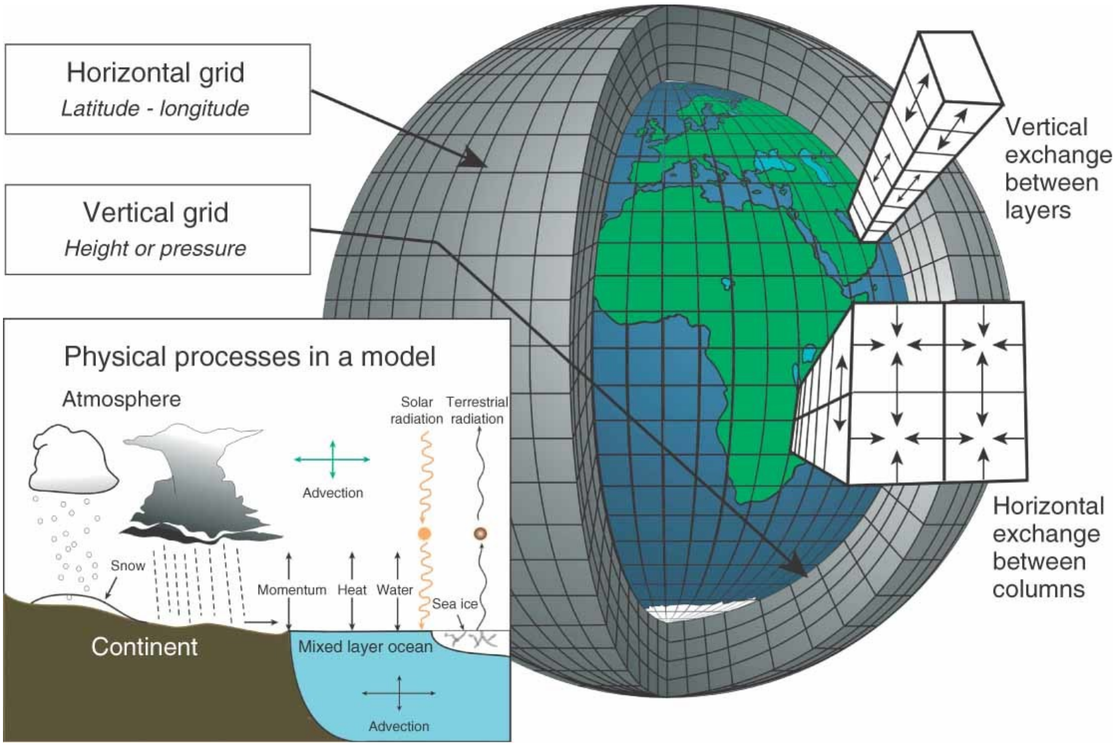


FIGURE 2.1: Schematic view of the discretization performed by a Global Climate Model (GCM) illustrating the physical processes solved and other elements such as the selection of the grid and coordinate system. **Source:** Adapted from Figure 2 in Edwards, 2011.

simulate regional climate dynamics at high resolution over a limited area (2.2), with boundary conditions driven by the output of GCMs [15, 16]. Additionally, statistical downscaling offers a cost-effective method to develop statistical models that establish an empirical link between coarse, large-scale atmospheric variables (predictors) and high-resolution regional surface variables (predictands) [17]. Together, these approaches generate vast amounts of climate data at multiple spatial and temporal scales, adding further heterogeneity and complexity to its management, and highlighting the pressing need for infrastructures and tools capable of supporting scalable storage, efficient access, and reproducible analysis.

In addition to GCMs, RCMs, and classical statistical approaches, machine learning has opened up a vast range of new opportunities for generating climate and meteorological data. Machine learning techniques, such as neural networks and ensemble methods, enable the analysis of complex, high-dimensional datasets, offering improved accuracy and efficiency in climate prediction [18, 19]. By learning from existing large datasets, machine learning models can capture intricate patterns in the data that may be missed by simple parametric models, leading to more precise and localized climate predictions. Furthermore, machine learning can help to integrate multi-source data from satellites, weather stations, ocean buoys, reanalyses, and model outputs. These capabilities can

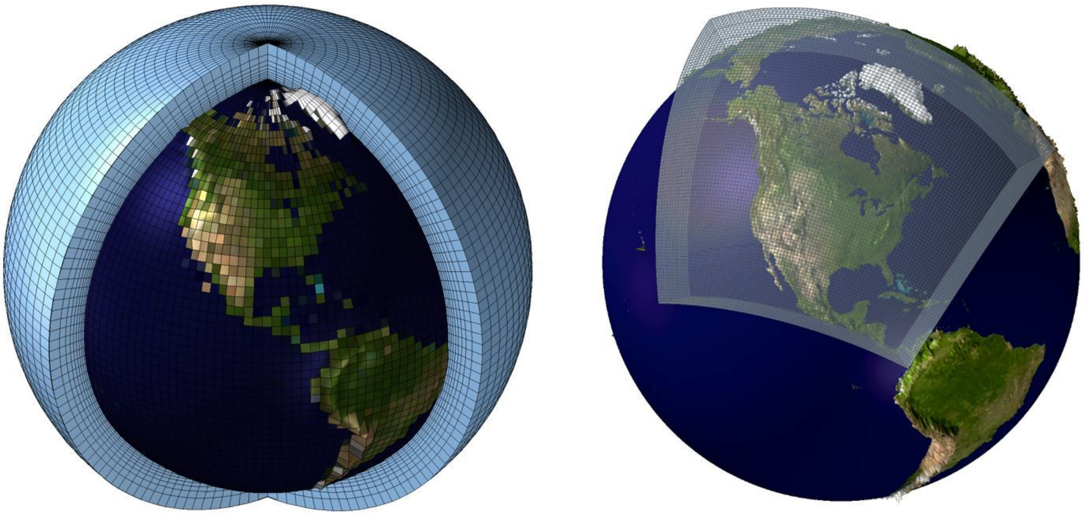


FIGURE 2.2: Conceptualization of a global climate model (left), and a regional climate model (right). **Source:** `https://www.ouranos.ca/en/science-du-climat-modelisation-climatique` (last accessed: 11 June 2025).

improve the reliability and resolution of climate models, and offer new tools for addressing challenges related to climate variability and change [20, 21].

Climate data science sits at the intersection of climate research and modern data technologies. It addresses the growing need to store, manage, and analyze vast and complex climate datasets generated from observations, simulations, and reanalyses. As climate data continues to increase in volume and complexity, driven largely by advances in global climate modeling and observational systems, so does the challenge of developing technology that allows the extraction of meaningful knowledge from it.

2.1.3 Climate Change Assessment

The vast amount of data generated by climate models is organized under initiatives such as the Coupled Model Intercomparison Project (CMIP). CMIP can trace its origins back to the *Charney Report* [22], which examined the links between CO_2 and climate and provided an authoritative summary of the state of the science at the time. The report can be considered one of the earliest uses of the *multi-model ensemble* approach, which allows to capture uncertainties and increase the robustness of projections, as no single model can represent the full complexity of the climate system. While these advancements enrich climate modeling and analysis, they also exacerbate the data management difficulties in accessing and processing the resulting large datasets [12].

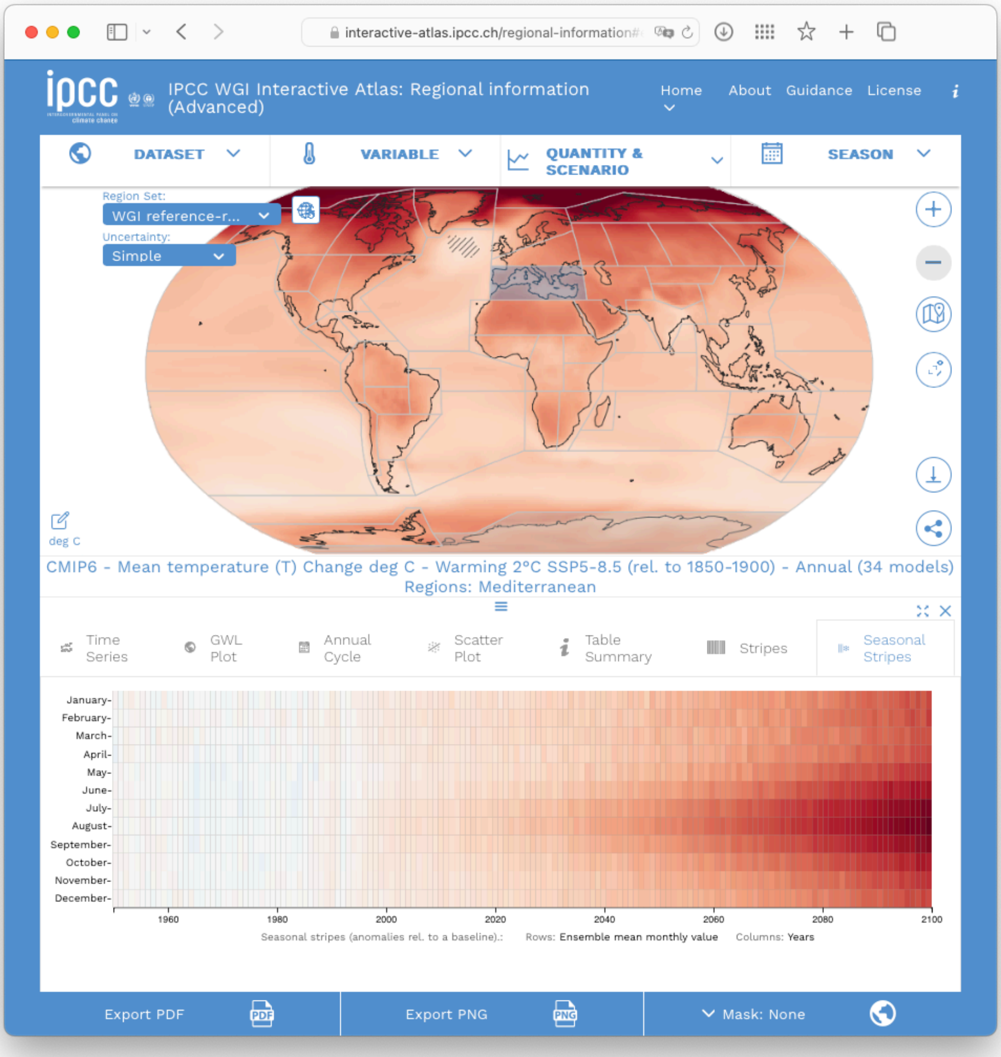
The Intergovernmental Panel on Climate Change (IPCC) is the leading international body responsible for assessing scientific knowledge related to climate change. Established in 1988 by the United Nations Environment Programme (UNEP) and the World Meteorological Organization (WMO), the IPCC assesses the state of knowledge on climate change. It publishes regular Assessment Reports—comprising a Summary for Policymakers as well as full technical chapters and methodology reports—serving policymakers, scientists, practitioners, and other stakeholders on climate impacts, future risks, and adaptation and mitigation options.

The Sixth Assessment Report (AR6) [23] provides the most comprehensive and up-to-date assessment of the current state of knowledge on climate change. It represents a robust scientific foundation for climate-related policy decisions, synthesizing scientific knowledge on climate change, its impacts, and mitigation strategies [24, 25]. The *Interactive Atlas*[1] [26] was introduced in AR6 as an innovation to expand the assessment, enabling flexible spatial and temporal analysis of most datasets and Climate Impact Drivers (CIDs) used in the report. It allows users to explore, interact with and download global maps, spatially aggregated time series, and other regional products that display recent trends and projected changes in emission scenarios for more than 20 CIDs. Figure 2.3 shows a snapshot of the Interactive Atlas.

The accessibility and reproducibility of scientific results in the context of the AR6 and the Interactive Atlas are major concerns, and central to addressing them is the effective management of climate data and the infrastructures that support it, which underpin the credibility and utility of scientific assessments [27]. Robust data management practices—including standardized data collection, quality control, long-term archiving, and open access policies—are essential for ensuring consistency, reproducibility, and comparability across observational records and model outputs. These practices not only facilitate the integration of heterogeneous datasets but also enhance transparency and enable informed decision-making across scientific, policy, and public domains.

Climate research increasingly requires collaborative and interactive science. Progress depends not only on individual researchers but also on large, distributed communities working together, often across disciplinary, institutional, and geographical boundaries. This calls for a next generation of climate data infrastructures and workflows that facilitate transparent collaboration, shared experimentation, and open communication of results.

[1] `https://interactive-atlas.ipcc.ch`



Region Set: WGI reference-r...

Uncertainty: Simple

CMIP6 - Mean temperature (T) Change deg C - Warming 2°C SSP5-8.5 (rel. to 1850-1900) - Annual (34 models)
Regions: Mediterranean

Seasonal stripes (anomalies rel. to a baseline).: Rows: Ensemble mean monthly value Columns: Years

FIGURE 2.3: Snapshot of the IPCC AR6 Interactive Atlas web application. The Interactive Atlas is a novel tool for flexible spatial and temporal analyses of much of the observed and projected climate change information underpinning the Working Group I contribution to the Sixth Assessment Report, including regional synthesis for Climatic Impact-Drivers. **Source:** `https://interactive-atlas.ipcc.ch/regional-information` (last accessed: 29 August 2025).

2.2 Objectives and Contributions

This thesis addresses the technological foundations of climate data science by focusing on next-generation scalable data analysis and storage infrastructure. Throughout, technology is treated as a means rather than an end, in service of the epistemic goal of producing reliable, meaningful knowledge about climate change. The objectives and contributions of this work are framed by three complementary technological dimensions: **climate data storage**, **climate data analysis**, and **climate data infrastructure**.

Climate data storage refers to the challenges associated with organizing and preserving climate-related information within information systems. Research in this area encompasses both the theoretical foundations and practical implementations of systems designed to structure, manage, and access data. This includes the development of models that define how information is organized and interrelated. Additionally, topics such as distributed storage, data replication, consistency models, and fault tolerance are investigated, all of which are essential for ensuring the availability and reliability of information in complex environments.

Climate data analysis involves the interpretation and visualization of climate-related data with the aim of understanding patterns, trends, and anomalies in the Earth's climate system. This process requires both logical and statistical inference from climate datasets. Logical and statistical inference differ fundamentally in their objectives and methodologies for deriving epistemological conclusions from data, although they often appear together in climate data analysis. For instance, combining gridded surface temperature records that are split across multiple files along the temporal axis requires the logical composition of a time series into a single logical unit—an operation of logical inference. In contrast, computing the spatial average of that logical time series across spatial dimensions involves statistical inference. This thesis places particular emphasis on logical inference operations within the context of climate data science, rather than on statistical inference.

Both climate data storage and analysis are carried out within complex **climate data infrastructures**, both hardware and software, that enable the exploitation of data and the acquisition of knowledge about the climate system. These infrastructures range from personal workstations to high-performance computing (HPC) systems and cloud-based platforms. The choice of infrastructure significantly influences the entire process of climate data storage and analysis, leading to different technological solutions whose benefits and limitations must be properly understood to ensure their efficient use. These infrastructures enable researchers and scientists to analyze large volumes of climate data with varying degrees of efficiency, depending on the services they provide. In this thesis, *climate data management* refers to the processes of organizing and maintaining data within these infrastructures to ensure its accuracy, accessibility, and long-term usability.

Considering these three dimensions of climate data science, this thesis aims to advance the state of the art in climate data storage, analysis, and infrastructure by designing, implementing, and evaluating systems that reduce time to insight and enhance reproducibility in climate research. To this end, it pursues the following specific objectives, which address key technological challenges in climate data science:

1. **Characterize climate data and derive its storage and analysis requirements.** Understand the nature and storage and analysis requirements of climate data generated by Global Climate Models. Also, examining the characteristics of dense and sparse data and their implications for climate data storage, analysis, and management. Moreover, this works aims to derive storage/layout and metadata requirements that support common slicing, multi-dimensional aggregation, and time-series construction.

   The first part of the thesis is dedicated to meeting this objective. Each of the three dimensions of the technological solution to climate data–intensive science — climate data storage, analysis, and infrastructures — is analyzed in detail. A comprehensive understanding of these dimensions constitutes a necessary theoretical prerequisite to effectively engage with the contributions presented in this thesis.

2. **Enhance climate data infrastructures.** This work differentiates between climate data infrastructures that provide only data storage and climate data infrastructures that provide climate data analysis capabilities on top of their available data. This thesis offers its own interpretation of the concept of a **data laboratory** (or **DataLab**), borrowing the term from the experimental sciences. In this work, it is introduced as the setting for climate data research, encompassing not only data storage but also computing capabilities. The elements and instruments that make the data laboratory are presented and it is discussed how they ease and favor climate data analysis tasks. As an indispensable component of the data laboratory, FAIR-aligned workflows (based on data recipes and notebooks) in R and Python will be a fundamental requirement. The justification for data laboratories is provided in Section 5.4.

   Chapter 6 introduces the Interactive Atlas DataLab, a data laboratory designed to extend the contributions of the Interactive Atlas of the AR6. The Interactive Atlas DataLab provides the necessary hardware and software infrastructure to ensure the reproducibility and reusability of the products supporting the Interactive Atlas. Accessible to anyone with an Internet connection, the DataLab strengthens and promotes the implementation of the FAIR principles within the IPCC and the AR6. Chapter 8 introduces a data laboratory that, unlike the Interactive Atlas DataLab, is based on remote data access and virtual analysis-ready data. That chapter will bring together and summarize all the components of the thesis in a single place.

3. **Enhance climate data analysis.** This thesis describes and analyzes the main methods of logical and statistical inference found in climate data analysis. A taxonomy is presented that considers the different ways in which climate data analysis is currently carried out, based on how data are accessed—whether by downloading files, accessing remote data sources, or performing computations on remote servers. The usage of **virtual aggregations** to generate logical units of analysis is proposed, to significantly simplify the climate data analysis experience, removing many complex logical analysis tasks from users and applications.

   This thesis introduces several technologies for improving climate data analysis in Chapter 4. Implementations of these technologies, along with their evaluation, are presented in Chapter 7 and Chapter 8. Chapter 7 addresses this objective by proposing a methodology for generating virtual aggregations within the climate data infrastructure of the Earth System Grid Federation (ESGF), the reference system for storing and distributing climate data in CMIP. Chapter 8 integrates this methodology into a data laboratory, demonstrating how climate data analysis can be significantly enhanced through the provision of appropriate tools and environments.

4. **Evaluate the performance of the infrastructures.** The technologies proposed in this thesis, dedicated to climate data storage and analysis, along with their supporting infrastructures, are evaluated in terms of performance. Special attention is given to latency and throughput, which provide objective measures of efficiency and allow for scientific comparison among the proposed solutions. This objective is achieved through performance evaluations conducted to assess the practicality of the presented data laboratories and the methods for generating virtual aggregations.

2.3 Structure

The thesis is organized in three main parts: Introduction (Part I), Main Results (Part II) and Concluding Remarks (Part III). Part I consists of three chapters that serve as an introduction to the fundamental components of this thesis, which are *climate data storage* 3, *climate data analysis* 4 and *climate data infrastructures* 5. Part II presents the main contributions of this thesis to the three topics covered in Part I. In accordance with the objectives of this thesis, this part introduces the data laboratory of the Interactive Atlas 7 as an enhancement to current climate data infrastructures and the ESGF Virtual

Aggregation 7 as a methodology to improve climate data analysis. Furthermore, the ESGF Virtual Aggregation is evaluated in terms of performance and its capability to reproduce results generated within the Interactive Atlas data laboratory 8. Part III of the thesis presents the final conclusions drawn from the research. It also outlines potential directions for future work, identifying areas that were beyond the scope of the current study but hold promise for further exploration.

Part I

Introduction

Chapter 3

Climate Data Storage

3.1 Multidimensional Arrays and Storage Patterns

Given the increasing data complexity discussed in Chapter 2, this section introduces multidimensional arrays as the core structure for scientific datasets. The scientific community often refers to scientific data as *multidimensional* [28, 29]. While this is correct, it is important to recognize that all data is inherently multidimensional, which makes the statement unnecessarily redundant [30]. For example, a simple dataset of product purchases already has three dimensions: time, person, and product - without prejudice to these entities having, in turn, other dimensions (such as name or age in the case of a person). Similarly, a dataset of surface temperatures can have time, latitude, and longitude as its dimensions. The key difference is that purchase data is typically sparse, whereas surface temperature data generated by a climate model is dense. That is, for each combination of time, latitude, and longitude, there is a recorded temperature value, whereas there is not necessarily a purchase record for every combination of person, product, and time.

Thus, rather than using *multidimensional* as a catch-all, it is clearer to describe datasets as *dense* or *sparse*, depending on how fully the dimensional space is populated. Common misinterpretations in climate data storage and manipulation stem from conflating storage-level representation with logical structure [30]. Accordingly, this thesis emphasizes the distinction between *sparse* and *dense* data (see Figure 3.1).

Because outside many scientific domains data tends to be sparse, database systems outside the scientific realm have different requirements for the storage of sparse data, in contrast to the requirements for the storage of dense data. By contrast, climate datasets are largely dense. Consequently, it relies on self-describing multidimensional array stores

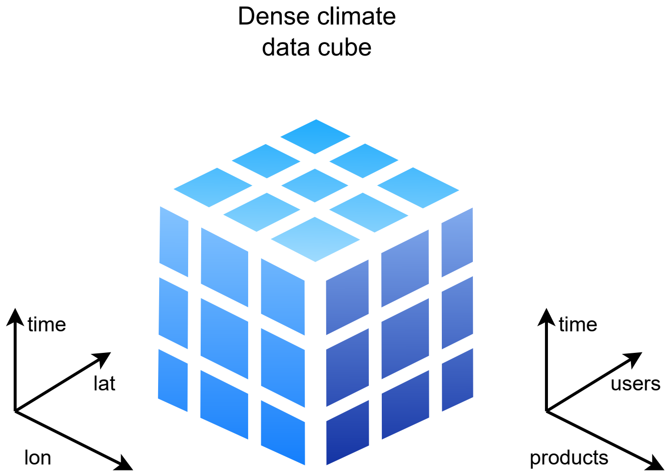


Figure 3.1: Characterization of two types of data: *dense* and *sparse*. On the left, a data cube represents a climate variable, such as surface temperature, which is considered *dense* because a climate model provides a temperature value for each time step and spatial position (latitude/longitude). On the right, a data cube represents a database of product purchases. Since not all users buy all products at every time step, the data cube is mostly empty, making it *sparse*. Note that climate data does not necessarily have to be dense; for example, climate data recorded from weather stations is sparse data. **Source:**Cimadevilla, 2025.

(e.g., NetCDF, HDF5, and, increasingly, cloud-native stores such as Zarr). These libraries support chunked, compressed multidimensional variables and rich metadata.

Several *scientific data types* have been proposed that allow for categorizing climate data. These data types include *Grid*, *Trajectory*, *Station*, *Radial* and *Swath* data types among others [31, 32]. Gridded datasets produced by climate models (GCMs or RCMs) belong to the category of dense data. The Grid scientific data type refers to information of physical or *field* variables that exist within a given spatio-temporal domain, the *support* variables that describe the spans of the spatio-temporal cells of the grid, and additional metadata such as the units in which the physical variables are measured. Thus, gridded fields are best interpreted as values representative of finite grid cells (area/volume elements) and time intervals, for example, cell means, rather than as point measurements. Figure 3.2 illustrates alternative interpretations of gridded data.

In contrast, station data or time-series data consists of discrete measurements collected at specific locations, such as weather stations, ocean buoys, or sensors. Each record represents the value of a climate variable—such as temperature, humidity, or precipitation—at a given time and place. Climate data produced by observational stations is sparse data that belongs to the *Station* scientific data type. The Station scientific data type represents data at scattered locations and times with no implied relationship among

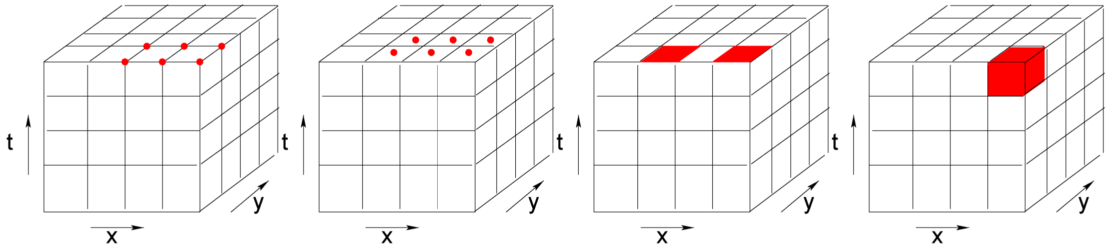


FIGURE 3.2: Different meaning and interpretations of cells in spacetime arrays. Support variables, such as spatio-temporal variables describing the domain of a model, can represent either point or cell values. Climate variables, such as temperature or precipitation, may represent information computed as an average, sum or any other function from all the values existing in the scope of the spatio-temporal cell. **Source:** Lu et al., 2018.

coordinate positions. Unlike gridded datasets, station datasets are sparse, occupying only a small subset of the potential spatio-temporal dimensional space.

The development of climate software libraries for data storage has promoted the appearance of models of data that rely on the utilization of multidimensional arrays complemented with key-value attributes, though minor variations may exist among different data models. Climate data storage libraries usually allow hierarchical structuring of the data. Due to the capabilities of these libraries to attach attributes to the multidimensional arrays used for storage, climate data storage libraries are usually referred to as *self-describing*. For example, a 3-d multidimensional array (time, latitude, longitude) of surface temperatures can have attached the attribute *units* with value *Kelvin* or *Celsius*. Since different individuals or organizations can use different attributes to describe the same information, climate metadata standards — such as the CF conventions — provide a standard way to represent variables, coordinates, units, and grids across datasets.

Suitable data formats for the storage of climate data are NetCDF [33], HDF5 [34] and Zarr [35]. Both HDF5 and Zarr can act as the storage format for NetCDF-4 datasets, providing chunked, compressed storage for multidimensional arrays with attributes. In practice, NetCDF-4 uses HDF5 as its on-disk container, while NCZarr enables representing the NetCDF data model in Zarr. This chapter describes the rationale and interaction between these different storage technologies.

3.1.1 Multidimensional Arrays

Gridded climate data generated by climate models encompasses spatio-temporal data often represented as 3-D or 4-D dense arrays, commonly referred to as rasters or grids. The fundamental abstraction currently in use to deal with this kind of data is multidimensional arrays [29] with attached attributes. Multidimensional arrays are widely used and have been extensively studied in various fields of computer science, including programming

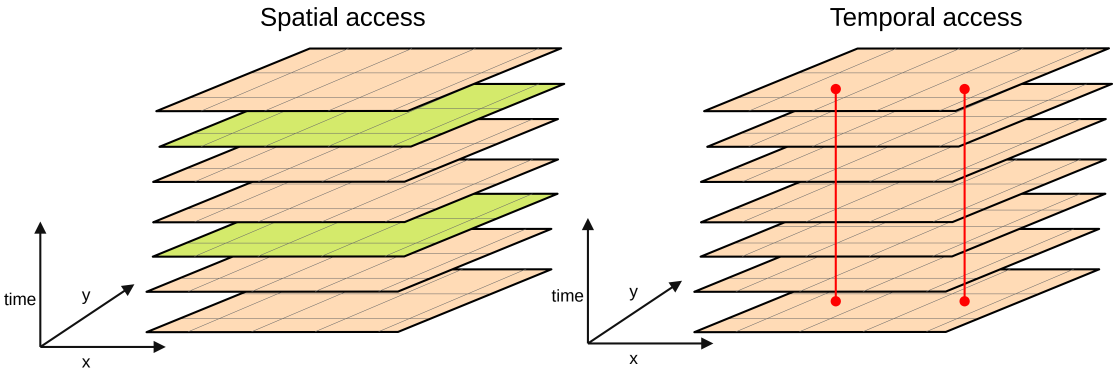


Figure 3.3: Different types of data access in a three-dimensional array consisting of two spatial dimensions and one temporal dimension. Depending on how the multidimensional array is stored on disk, data access will be more or less efficient.

languages, scientific and high-performance computing, graphics, and databases [36]. Using multidimensional arrays, various storage layouts can be employed to optimize different data access patterns. Figure 3.3 illustrates different access patterns to a climate field variable of three dimensions.

It is important to note here the close relationship between multidimensional arrays and gridded data produced by GCMs. Climate data represents physical phenomena that vary continuously across space and time (*field variables*) and the information supporting the spatio-temporal cells associated with the physical variables (*support variables*) [28]. The data of a climate variable, such as surface temperature, are discretized in a spatio-temporal grid of contiguous locations in both spatial and temporal axes. Multidimensional arrays offer a mechanism to store the values of field and support variables in such a way that they are close to each other when stored on disk or in memory. In addition, information such as the units of measurement for both field and support variables needs to be recorded.

Multidimensional arrays align with the inherent structure of dense data, such as spatial grids (2D), volumetric data (3D), or spatiotemporal data (4D). Additional axes can encode other continuous geophysical quantities (e.g., height, pressure) or discrete categories (e.g. ensemble model number, scenario). Multidimensional arrays enable direct mapping between real-world dimensions (e.g., latitude, longitude, altitude, time) and array dimensions, making them an intuitive choice.

Multidimensional arrays are an efficient choice for storage of dense data. Unlike sparse data, which often requires intricate data structures, arrays store all elements directly and straightforwardly. Moreover, the contiguous memory allocation of arrays optimizes memory usage and reduces overhead compared to alternative storage methods such as tables of coordinates. Figure 3.4 illustrates the storage of a multidimensional array in a contiguous manner.

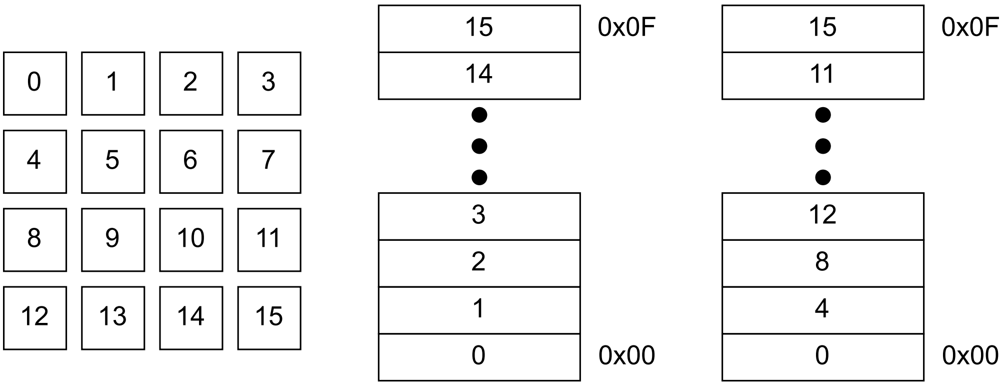


Figure 3.4: Illustration of how a multidimensional array is stored in the linear space of a storage device, such as memory or disk, using row-major and column-major order. Due to the process of loading data from storage, contiguous locations are fetched faster than distant ones. Thus, different storage methods for the multidimensional array provide more efficient data access depending on the storage pattern.

3.1.2 Chunking and Tiling

Chunking and tiling involve dividing a large array into smaller, more manageable subarrays (chunks or tiles) to enhance the performance and efficiency of storing, processing, and analyzing large datasets. One key advantage of chunking is that the chunks can be stored independently, often in compressed formats, which helps reduce disk space requirements. Unlike a full-dimensional array, which must be entirely decompressed to access even a single element, chunking allows selective decompression of individual chunks, making data access more efficient. Figure 3.5 illustrates chunking in a multidimensional array.

Using smaller chunks also improves memory efficiency by enabling computations to fit within memory caches, eliminating the need to load the entire dataset at once. Additionally, chunking minimizes input/output (I/O) overhead by focusing on smaller, localized chunks of data. This approach increases flexibility, allowing for selective loading and processing of specific chunks, such as data corresponding to a particular region or time frame.

Regular, directional, and sliced chunking are strategies for dividing multidimensional arrays into smaller chunks, each tailored to specific computational or storage needs [36]. Regular chunking divides the array into equal-sized chunks across all dimensions. Each chunk has the same shape and size (except possibly at the edges, where the array might not divide evenly). The algorithm to find chunks is easy to implement, but this chunking is not flexible enough to represent many data access patterns. Directional chunking creates irregular chunks by slicing the array into arbitrary non-uniform segments, often based on predefined ranges or domain-specific criteria. It offers flexibility for datasets with varying densities or importance across dimensions and enables customized processing

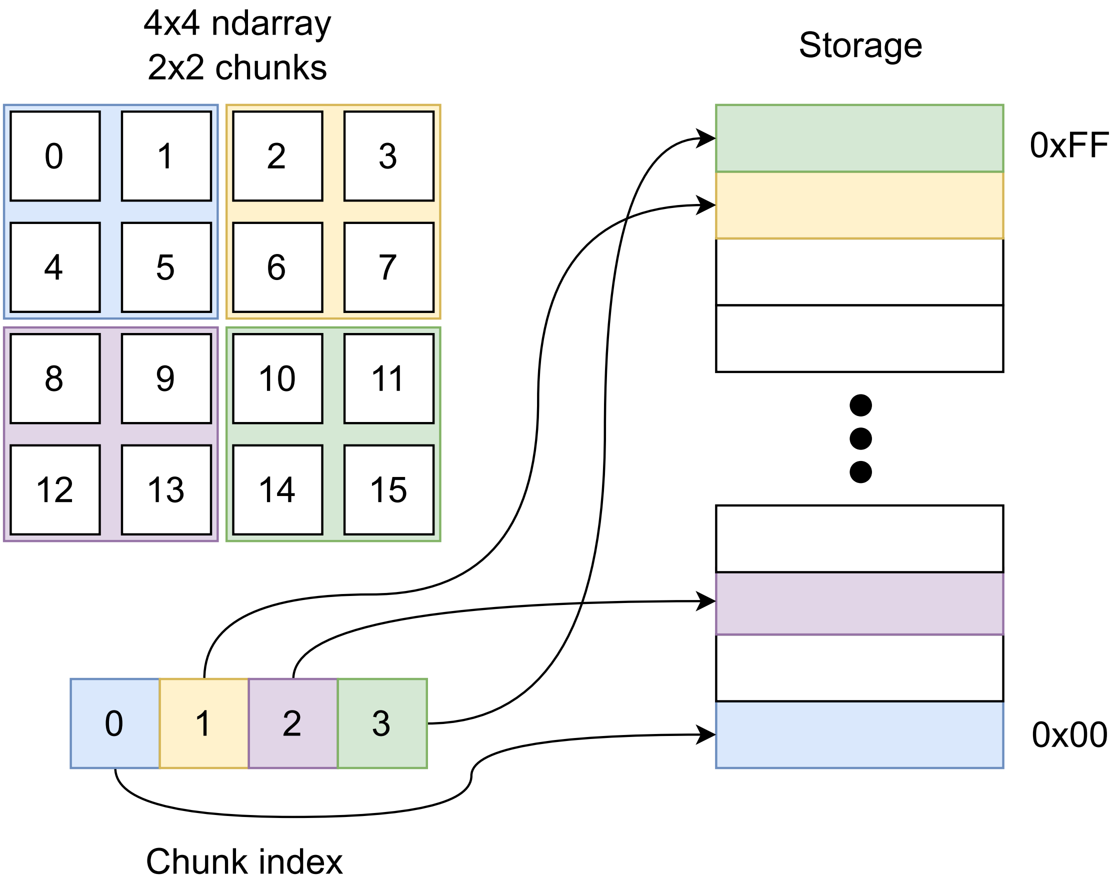


FIGURE 3.5: Illustration of chunk indexing. A chunk index is a data structure that maps each chunk's identifier to its position in linear storage.

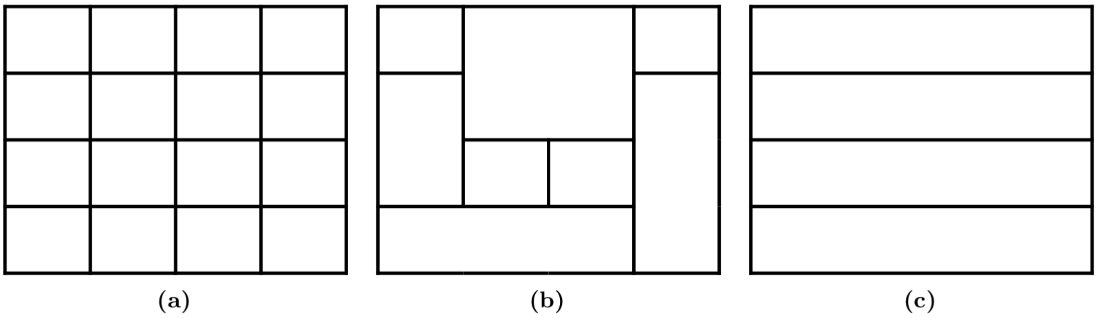


FIGURE 3.6: Arbitrary chunking: (a) regular, (b) directional, and (c) sliced. **Source:** Rusu, 2023.

or storage, such as focusing on high-activity regions. Sliced chunking is a special case of arbitrary chunking that corresponds to slicing a particular dimension with hyperplanes at every position in its domain. Figure 3.6 illustrates different chunking types.

Chunking involves critical decisions about how to store and locate data of a chunked dataset. Some formats opt for using data structures such as B-trees for indexing the chunks of multidimensional arrays in the context of a linear address space or file. Other formats opt for using a key-value store in which the format of the keys is easily obtained from client applications by looking only at the shape of the multidimensional array and the chosen chunking. These decisions may have performance implications that need to be accounted for in order to provide proper usage of a given technology. These issues are further discussed in the context of specific storage applications in the following sections.

3.1.3 Compression

One of the key advantages of chunking is its ability to exploit localized redundancy. In many types of data, such as images or text, patterns tend to repeat within a confined region. For example, neighboring pixels in an image are often similar in color or brightness, and repeated words or phrases occur within sections of text. Chunking ensures that compression algorithms can capture these localized patterns effectively, reducing the storage required for the same data.

Processing data in chunks significantly reduces memory and computational overhead. Instead of requiring the entire dataset to be loaded into memory, only one chunk at a time needs to be processed. This feature is crucial for applications and systems dealing with Big Data. Furthermore, this incremental approach allows data to be compressed and decompressed in parallel. By dividing data into independent chunks, compression tasks can be distributed across multiple processors or systems, improving efficiency and scalability.

Examples of lossless compression algorithms commonly used within climate data are GZIP and LZF. GZIP is a widely used compression algorithm based on the DEFLATE algorithm. It allows users to enable compression by specifying a compression level ranging from 0 (no compression) to 9 (maximum compression). The higher the compression level, the more CPU-intensive the process becomes, but the resulting files are smaller. This trade-off makes GZIP suitable for scenarios where storage space is a critical concern, and the computational cost of compression and decompression is acceptable. LZF is a lightweight compression algorithm designed for speed and simplicity. Unlike GZIP, which focuses on achieving high compression ratios, LZF prioritizes fast compression and decompression times. While LZF typically produces larger files compared to GZIP, its lower computational overhead makes it a popular choice for performance-sensitive tasks, making it ideal for applications that involve frequent read and write operations or require near-instantaneous access to compressed data.

Recent trends in HPC indicate a rapid rise in core counts accompanied by a proportional decline in memory bandwidth per core. The efficiency of future computations will heavily depend on the extent of data movement. For scientific applications that primarily process large arrays of floating-point numbers, lossless compression typically provides only minimal data reduction. Lossy compression has been widely adopted in computer graphics, particularly for reducing texture storage, with dedicated hardware for texture decompression now standard in GPUs. While lossy compression in graphics is driven by the goal of maintaining visual fidelity, quantitative analysis and numerical simulations impose much stricter requirements on acceptable error levels [37, 38].

Weather and climate forecast centers worldwide generate climate data at a scale of petabytes annually [39]. Compression is crucial for reducing storage demands and enabling efficient data sharing. Existing methods fail to distinguish meaningful information from noise in the data, leaving the true level of significant precision unaccounted for. By preserving only the bits that represent meaningful information, compression ratios as high as 60x can be achieved [38].

3.1.4 Layout Filters (Shuffle)

Shuffling is a preprocessing step applied before compression to improve its efficiency. It works by reorganizing the data at the byte level, ensuring that all first bytes of each data element are grouped together, followed by all second bytes, and so on. This reordering enhances the ability of compression algorithms to detect patterns and redundancies, leading to better compression ratios.

The primary advantage of shuffling is that it significantly improves compression efficiency. Many compression algorithms, such as GZIP and LZF, perform better when data exhibit similar byte patterns. By exposing these patterns, shuffling optimizes the storage of datasets, reducing their overall size and improving read and write performance. Since compressed data takes up less disk space, I/O operations become faster, making shuffling particularly useful for large datasets in scientific computing and big data applications.

Shuffling is most effective when applied to structured numerical data, such as multidimensional arrays of floating-point or integer values. It is especially beneficial for datasets where neighboring values do not share similar byte patterns in their natural order. However, for data that are already highly compressed or inherently redundant at the byte level, shuffling may offer little to no additional benefit.

3.1.5 Data Integrity (Checksums)

An additional benefit of chunking is improved error resilience. Errors in a dataset are confined to the specific chunk in which they occur, preventing them from propagating to other parts of the data. This localization ensures that even if corruption occurs, the impact is limited, making chunking particularly valuable in scenarios involving data transmission over unreliable networks. Error correction codes can also be applied at the chunk level, further enhancing reliability.

The Fletcher-32 checksum is an example of an error detection algorithm that computes a checksum value to verify the integrity of data. This checksum filter is implemented by the various climate storage libraries. Fletcher-32 is computationally inexpensive compared to more complex algorithms such as CRC (Cyclic Redundancy Check). It requires only simple addition and modulo operations, making it suitable for systems with limited processing power. The 32-bit checksum provides a balance between error detection capability and data size overhead. While it is not as robust as some modern error-checking methods, its efficiency and effectiveness for typical use cases ensure its continued relevance in various domains, such as climate data management.

In distributed workflows, checksums play a key role in ensuring reliable collaboration across nodes and systems. Each worker can independently validate the integrity of the chunks it processes, avoiding the propagation of corrupted data throughout the pipeline. This guarantees that large-scale computations, often executed on heterogeneous and geographically dispersed resources, can proceed with confidence in the correctness of intermediate and final results. As a result, checksums are not just a safeguard but a foundational mechanism for building trustworthy distributed data workflows.

3.2 HDF5

HDF5 (Hierarchical Data Format version 5) is a library and file format designed to store and organize large amounts of dense data efficiently. Developed by the HDF Group, HDF5 is widely used in scientific computing, engineering, and other domains that require high-performance data storage and retrieval. Its design enables the management of complex datasets, making it an essential tool for researchers, developers, and data scientists. It is designed to perform equally well in small, single-user environments and large, distributed computing systems. With support for parallel I/O operations, HDF5 is particularly suited to HPC applications, such as climate models, enabling efficient data access and processing in supercomputing environments.

HDF5 plays a critical role in managing and organizing climate data, particularly model outputs, which consist of multidimensional array data produced by HPC applications. Climate models produce datasets that include variables like temperature, precipitation, wind speed, and atmospheric pressure across time and space. These datasets are typically represented as multidimensional arrays with associated metadata, requiring a robust

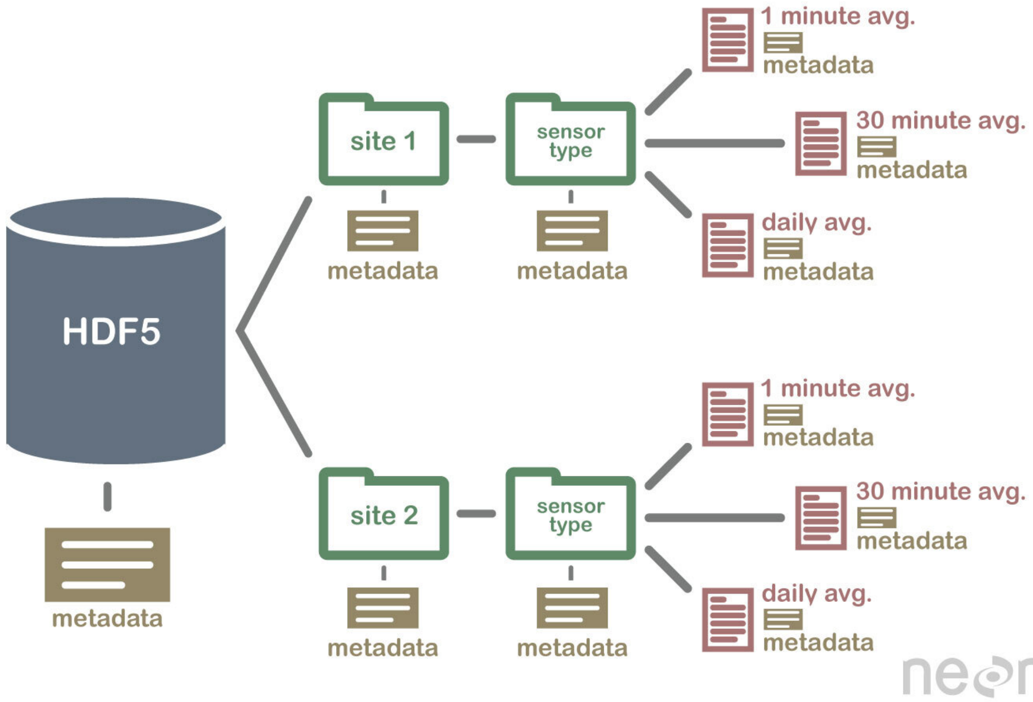


FIGURE 3.7: Hierarchical structure of an HDF5 file, containing several datasets of different temporal resolution. Both groups and datasets can be attached with attributes. **Source:** `https://www.neonscience.org/resources/learning-hub/tutorials/about-hdf5` (last accessed: 11 June 2025).

format capable of handling both large data volumes and intricate structures. HDF5 provides an ideal foundation for this purpose.

HDF5 allows organizing data hierarchically, much like a file system, with datasets functioning as files and groups acting as directories. Figure 3.7 illustrates the hierarchical structure of an HDF5 file. The format is optimized for efficient data storage and access of dense multidimensional arrays, incorporating features such as chunking and compression to minimize storage space while maximizing retrieval speed. Its ability to access specific subsets of data without loading entire files is especially valuable when working with large datasets.

An HDF5 file appears to the user as a directed graph. The nodes of this graph are the higher-level HDF5 objects that are exposed by the HDF5 APIs (groups, datasets, and attributes). At the lowest level, as information is written to the disk, an HDF5 file is made up of a superblock, B-tree nodes, heap blocks, object headers, object data, and free space. The HDF5 library utilizes these lower-level objects to construct higher-level objects, which are then exposed to users and applications through the APIs. For example, a group consists of an object header containing a message that references a local heap (for storing links to objects within the group) and a B-tree (which indexes these links). Similarly, a dataset is represented by an object header that includes messages defining

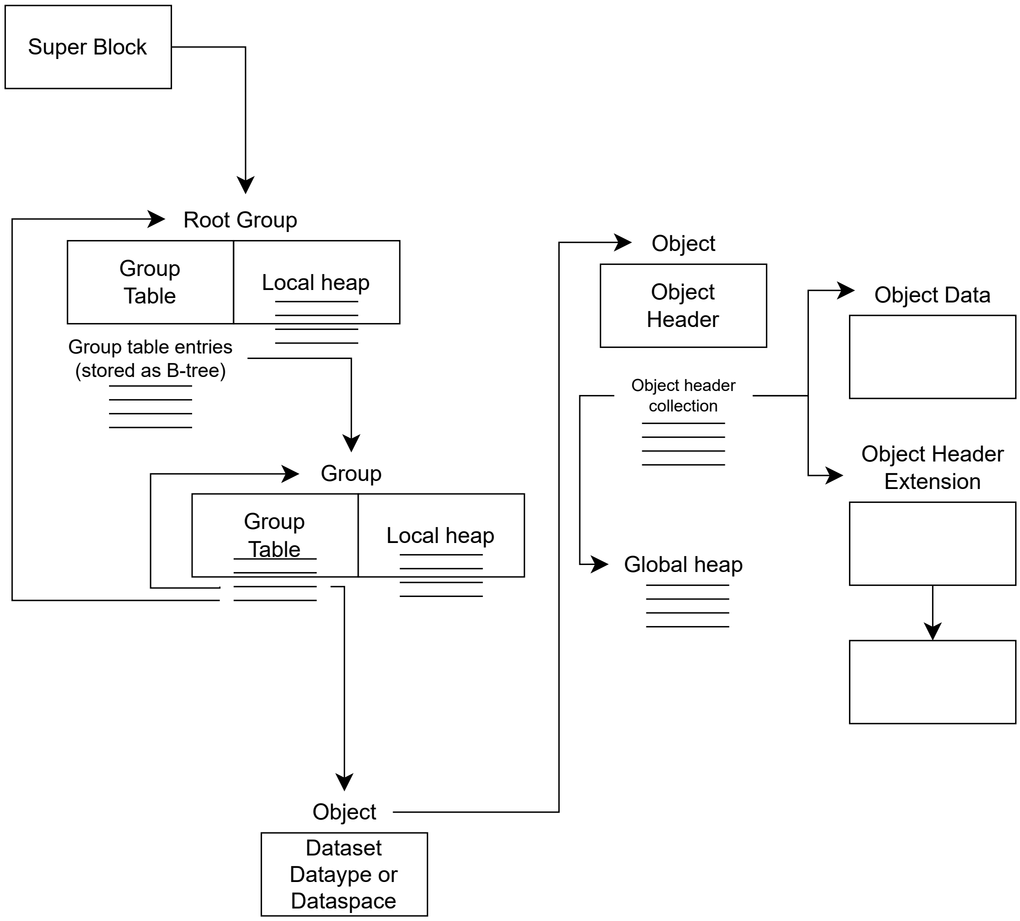


Figure 3.8: Illustration of HDF5 metadata structures for storage of high-level objects (groups, datasets and attributes). For instance, a group is an object header that contains a message pointing to a local heap (for storing the links to objects in the group) and a B-tree (which indexes the links). A dataset is an object header that contains messages describing the datatype, dataspace, layout, filters, external files, fill value, and other elements, with the layout message pointing either to a raw data chunk or to a B-tree that references raw data chunks.

its datatype, dataspace, layout, filters, external files, fill value, and other attributes. The layout message, in turn, directs either to a raw data chunk or to a B-tree that indexes raw data chunks. Figure 3.8 illustrates the low-level objects of an HDF5 file.

3.2.1 Virtual File Drivers (VFDs)

HDF5 Virtual File Drivers (VFDs) provide an abstraction layer between the HDF5 lower-level objects and the underlying storage systems. Thus, low-level objects may be stored in different places that can enhance the storage performance of different storage systems. In the context of climate datasets, SEC2 is the most used VFD because it is the default driver for UNIX-like systems where climate model data is analyzed. This

VFD uses standard POSIX system calls (read, write, lseek) to manage files on a local disk. Climate data storage currently relies mainly of the SEC2 VFD.

The Message Passing Interface (MPI) VFD allows parallel I/O operations in HPC environments. It enables multiple processes in an MPI application to efficiently read from and write to the same HDF5 file concurrently. By leveraging MPI-IO and parallel file systems, the MPI VFD provides scalability, optimized data distribution, and improved I/O performance for large-scale data processing tasks found in climate model simulations.

Another relevant VFD is the *Multi* VFD. It divides an HDF5 file into multiple physical files based on different data types (metadata, raw data, etc.), which can improve performance in certain workflows. The Multi VFD allows splitting the contents of an HDF5 file into six different categories:

1. The superblock data - The superblock may begin at certain predefined offsets within the HDF5 file, allowing a block of unspecified content for users to place additional information at the beginning (and end) of the HDF5 file without limiting the HDF5 library's ability to manage the objects within the file itself.

2. The B-tree data - B-trees allow flexible storage for objects that tend to grow in ways that cause the object to be stored discontinuously. The B-trees are used in several places in the HDF5 file format when an index is needed for another data structure.

3. The raw data - Multidimensional array data stored in HDF5 datasets.

4. The global heap data - Each HDF5 file has a global heap, which stores various types of information that is typically shared between datasets.

5. The local heap data - A local heap is a collection of small pieces of data that are particular to a single object in the HDF5 file. Objects can be inserted and removed from the heap at any time. The address of a heap does not change once the heap is created. For example, a group stores addresses of objects in symbol table nodes with the names of links stored in the group's local heap.

6. The object header data - Data objects contain the *real* user-visible information in the file. These objects compose the scientific data and other information that are generally thought of as "data" by the end-user. All the other information in the file is provided as a framework for storing and accessing these data objects.

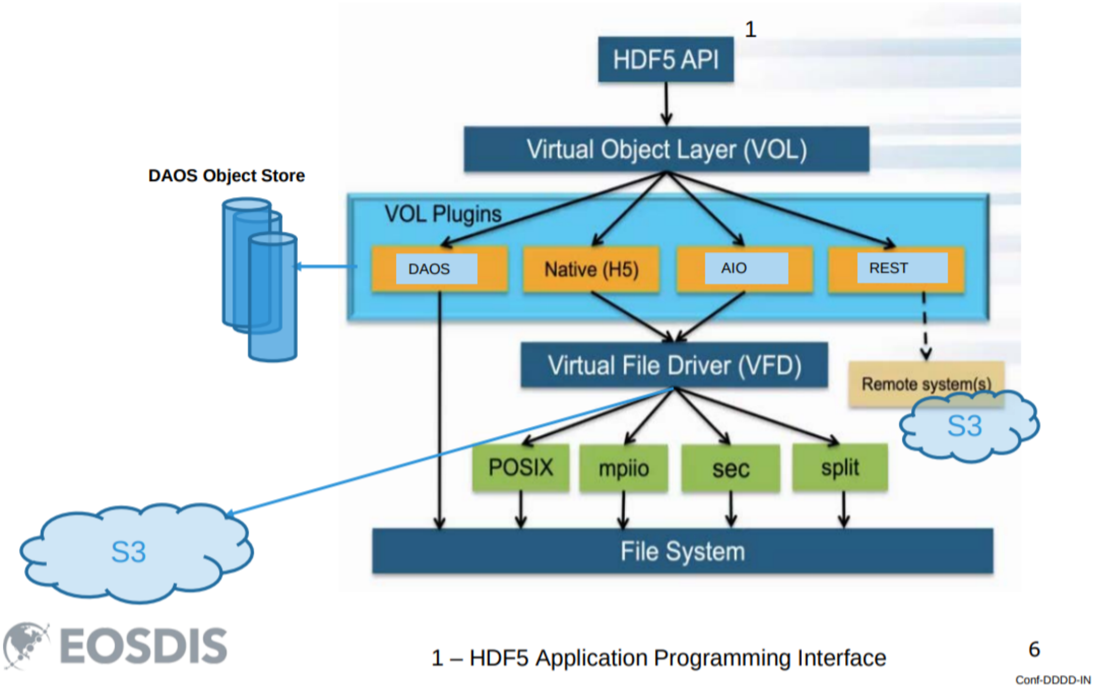


FIGURE 3.9: Architecture of the HDF5 library. **Source:** `https://support.hdfgroup.org/clinic/` (last accessed: 12 June 2025).

3.2.2 Virtual Object Layer (VOL)

The HDF5 Virtual Object Layer (VOL) is an abstraction layer within the HDF5 library that allows users to define custom storage backends for HDF5 objects such as datasets, groups, and attributes. It separates the HDF5 specification from the actual data storage mechanism, enabling users to work with HDF5 data structures while storing data in non-traditional storage solutions, such as cloud storage and object stores.

The traditional HDF5 format relies on the storage of HDF5 objects in the form of a directed graph as defined by the HDF5 specifications. The introduction of the VOL allows for overriding this behavior and defining custom implementations of the storage of HDF5 objects beyond those explicitly defined in the specifications. Figure 3.9 shows the architecture of HDF5 and its VOL and VFD layers.

3.2.3 Chunk Indexing

HDF5 supports dynamic extension of any or all axes of chunked datasets (multidimensional arrays). To accommodate the discontiguous storage of chunks within the linear address space of an HDF5 file, a mechanism is required to index the chunk locations. B-trees provide a flexible storage solution for objects that grow in a way that leads to discontiguous storage [40, 41]. B-trees are self-balancing search trees that maintain sorted

Operation	Time Complexity
Search	$O(\log N)$
Insertion	$O(\log N)$
Deletion	$O(\log N)$

TABLE 3.1: Time complexities of B-Tree operations.

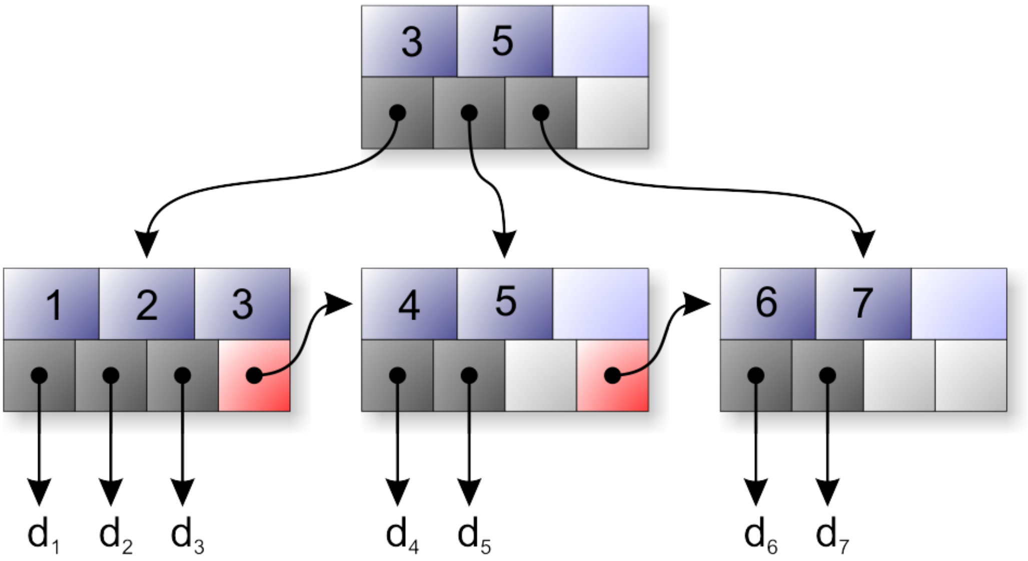


FIGURE 3.10: Illustration of a B-tree whose leaf nodes are connected using a single linked list. **Source:** `https://commons.wikimedia.org/wiki/File:Btree.png` (last accessed: 11 June 2025).

data and allow searches, sequential access, insertions, and deletions in logarithmic time. They are widely used in databases and file systems due to their efficiency in handling large amounts of data. Figure 3.10 illustrates a B-Tree data structure. The complexities of the main operations in a B-Tree are summarized in Table 3.1.

In HDF5, the version 1 B-tree structure is the original indexing method. It is gradually being replaced by the version 2 B-tree structure. However, both versions may coexist within the same file, depending on the application settings used during file creation. Version 1 B-trees in HDF5 files are based on the B-link tree structure [42]. At each level, sibling nodes are organized in a doubly linked list. The B-trees used in the file format store one more key than the number of children, meaning each child pointer in a B-tree node has both a left key and a right key. Pointers from internal nodes lead to subtrees, while pointers from leaf nodes direct to symbol nodes and raw data chunks. Apart from this distinction, internal and leaf nodes share the same structure.

Version 2 B-trees function like standard B-trees but with one key difference. Instead of using a simple pointer (or file address) to reference a child node, the pointer also includes two additional pieces of information: the number of records in the child node and the total number of records in the child node and all its descendants. This extra data enables efficient array-like indexing, making it easier to locate the nth record in the B-tree.

3.3 Zarr

Zarr is a cloud-native data format designed for storing and accessing large and dense multidimensional arrays, making it particularly well-suited for climate data applications [43]. Zarr enables concurrent, chunk-wise access to datasets, allowing researchers to analyze high-resolution climate simulations and observational data at scale without requiring extensive local storage or complex data extraction processes. Its native compatibility with cloud object storage enhances the accessibility and performance of climate data workflows, supporting real-time analysis and machine learning applications. Using Zarr, the climate science community has facilitated more efficient data sharing, interoperability, and collaboration between institutions [44].

Zarr defines a specification in which the elements of a Zarr store, such as groups, datasets (multidimensional arrays), and attributes of either groups or datasets, are stored in well-known endpoints. For example, the global entry point of a Zarr store might be the *.zmetadata* JSON file of a consolidated Zarr store or the *.zgroup* JSON file of a Zarr group. Endpoints of datasets are identified by the *.zarray* JSON file. Both groups and datasets store their attributes in the *.zattrs* JSON file.

The development and advancement of cloud computing technologies and tools have created new opportunities for climate data science. Various aspects of cloud computing make it especially effective in supporting scientific workflows, making it an attractive choice for researchers [44, 45]. Chapter 5 is dedicated to exploring the characteristics of cloud infrastructures. The storage approach of Zarr is different from HDF5, which maps the directed graph of low-level storage objects to byte locations in a linear address space. Due to this difference in storage mechanism, Zarr has found its potential as the storage format in object storage systems, mainly provided by cloud providers.

3.3.1 Chunk Indexing

In contrast to HDF5, which locates chunks by iterating over a B-tree to determine their byte position and length within the linear address space of an HDF5 file, Zarr uses a simple computation to derive a chunk's resource key based on the shape of a multidimensional array and its chunk shape. Zarr stores chunks as independent entities, often compressed, in a key-value store such as a file system directory or object storage

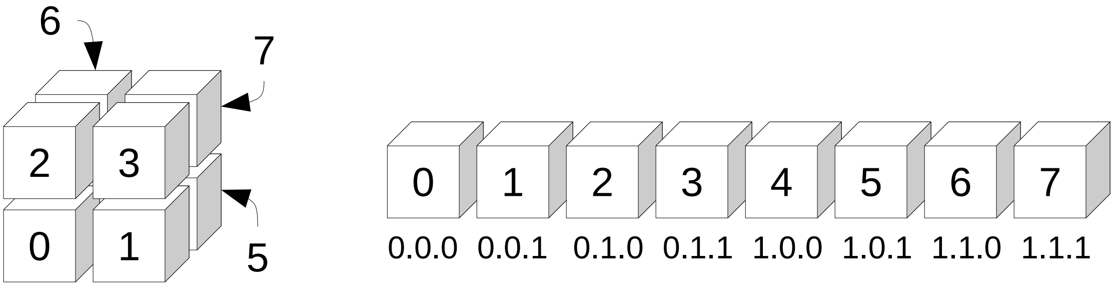


FIGURE 3.11: Illustration of chunk indexing based on chunk coordinates. Each cube is numbered with the numerical index of the chunk, and each one represents a chunk of arbitrary size. On the right, the coordinates of each chunk in the three-dimensional space of the multidimensional array are shown.

system. The chunk indices or chunk coordinates serve as keys in this structure. Figure 3.11 illustrates chunk indexing based on chunk coordinates. When a user requests a specific array region, Zarr computes the corresponding chunk indices, retrieves the relevant chunks, decompresses them, and extracts the required elements.

A Zarr array is stored as a collection of chunks, where each chunk is a contiguous sub-array of the full dataset. Given a Zarr array of shape $S = (s_1, s_2, \ldots, s_d)$ and a regular chunk shape $C = (c_1, c_2, \ldots, c_d)$, the chunk keys of the multidimensional array are computed as follows. Let $x = (x_1, x_2, \ldots, x_d)$ be the coordinates of an element in the array. The corresponding chunk index $k = (k_1, k_2, \ldots, k_d)$ is computed as:

$$k_i = \left\lfloor \frac{x_i}{c_i} \right\rfloor, \quad \forall i \in \{1, \ldots, d\}. \tag{3.1}$$

This formula determines which chunk contains the element at position x by integer division. The local offset $o = (o_1, o_2, \ldots, o_d)$ within the chunk is computed as:

$$o_i = x_i \mod c_i, \quad \forall i \in \{1, \ldots, d\}. \tag{3.2}$$

This determines the position of the element relative to the chunk's origin. In Zarr, each chunk is stored separately, and its storage key is often derived from the chunk index k. A typical representation is a string encoding:

$$\text{key} = \text{format}(k_1, k_2, \ldots, k_d). \tag{3.3}$$

For example, a chunk at index $(3, 2, 5)$ may have a key represented as "3.2.5". In summary, the following steps describe the process of locating a chunk in a Zarr array:

1. Compute the chunk index k using Eq. (1).

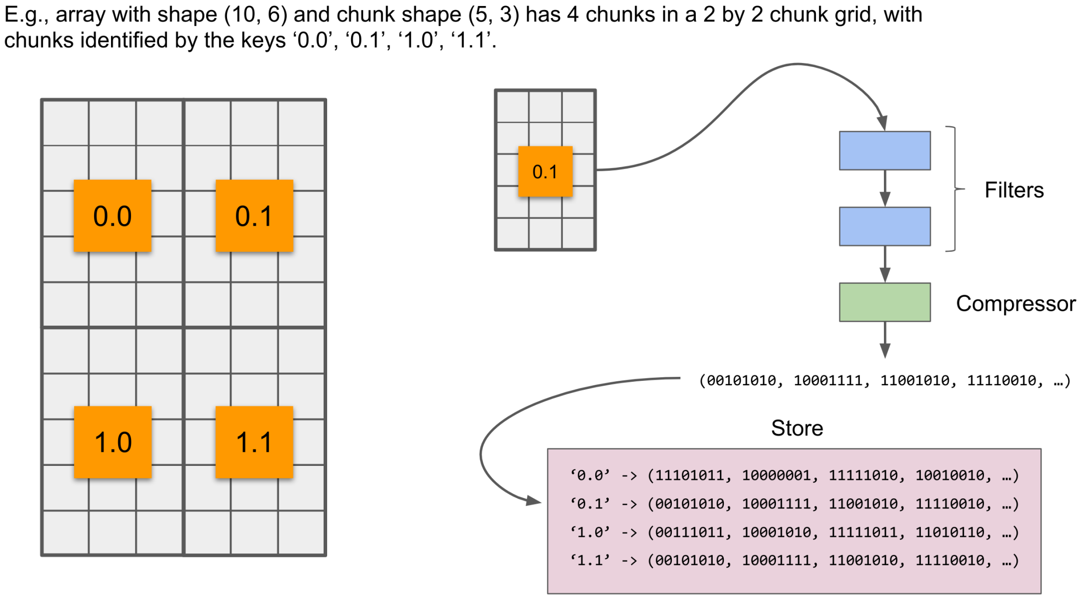


FIGURE 3.12: Illustration of chunked storage based on chunk coordinates in a Zarr store. **Source:** `https://commons.wikimedia.org/wiki/File:Zarr-scipy2019-storage.png` (last accessed: 11 June 2025).

2. Compute the local offset o within the chunk using Eq. (2).

3. Retrieve or generate the storage key for chunk lookup.

For example, in a two-dimensional array with a shape of (1000, 1000) and chunk sizes of (100, 100), the dataset is divided into a 10 x 10 grid of chunks. Each chunk has a unique coordinate in this grid, indexed by dividing the requested array indices by the chunk size. For instance, an access request for element (350, 420) would be mapped to chunk (3, 4), as integer division (350 / 100, 420 / 100) yields (3, 4). Thus, the resource for the chunk of a Zarr array named *arr* stored in a Zarr store *st* in server *https://example.com* would be *https://example.com/st/arr/3.4*. Figure 3.12 illustrates the chunk indexing followed by Zarr.

This chunk indexing method avoids traversal of B-trees and retrieval of metadata from remote sources if remote data access is used (see Section 4.2.2). This is why Zarr has been widely adopted for climate data storage in the context of object storage provided by cloud providers. HDF5 introduces significant overhead when traversing the B-tree to remotely access chunks of a multidimensional array. This overhead has traditionally been overlooked because, when accessing data locally, the associated latencies are negligible or nonexistent due to different levels of caching either by the HDF5 library, the operating system, or hardware devices.

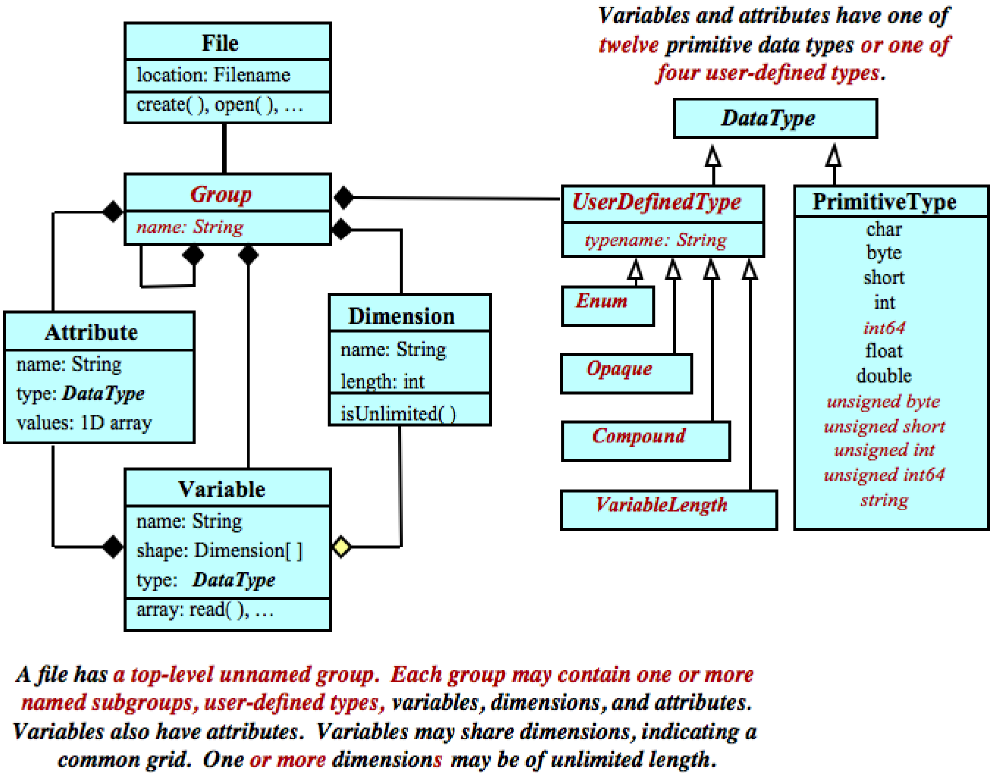


Figure 3.13: The NetCDF-4 data model. In addition to multidimensional arrays (variables), attributes, and groups, NetCDF also includes shared dimensions as part of its data model. **Source:** `https://docs.unidata.ucar.edu/netcdf-c/current/netcdf_data_model.html` (last accessed: 28 august 2025).

3.4 NetCDF-4 and the Common Data Model

The NetCDF (Network Common Data Format) library is a set of software interfaces and data formats specifically designed for managing and storing large-scale scientific data [33, 46]. It provides a self-describing, portable, and efficient way to organize dense data, making it an essential tool in scientific fields such as meteorology, oceanography, and climate research. Figure 3.13 shows the data model of NetCDF-4. NetCDF remains the most widely adopted format for climate data due to its compatibility, tooling, and support in major data distribution infrastructures.

Although NetCDF originated as its own storage format for multidimensional dense arrays, NetCDF-4 now relies on HDF5 for storage while preserving the NetCDF data model and API. This was the result of collaboration between the working groups of NetCDF and HDF5, in which the NetCDF library incorporated the capability of using HDF5 as the storage format by integrating features like chunking, compression, and parallel I/O. Furthermore, the NetCDF library has also incorporated the functionality to store climate datasets as Zarr stores for appropriate storage in cloud storage systems.

HDF5 and Zarr provide groups, datasets, and attributes as structure types, while NetCDF provides groups, variables, attributes, and shared dimensions. Despite the differing terminology, HDF5 *datasets*, Zarr *arrays*, and NetCDF *variables* all represent multidimensional arrays. The key distinction is that NetCDF supports named, shared dimensions between variables, whereas in HDF5 and Zarr each dataset/array has its own independent dataspace [47–50]. In general, the data and information of physical and climate variables, such as surface temperature, are stored in multidimensional arrays (either NetCDF variables, HDF5 datasets, or Zarr arrays), as well as data and information of supporting variables, such as cell bounds of the grid. Extra information, like units of measurement, is stored in the attributes attached to the multidimensional arrays.

3.4.1 Shared Dimensions and Coordinate Variables

The justification for shared dimensions in NetCDF, in contrast to private data spaces from HDF5 and Zarr, relies on the addition of new ways to represent relationships among multidimensional arrays in a NetCDF dataset. From the HDF5 perspective, the data space of a dataset has no intrinsic meaning except to define the layout in computer storage [47]. HDF5 uses the data space of the dataset as an implementation detail at the physical level of abstraction. The aim is to represent that both physical and support variables might share a common spatio-temporal domain. A key question is how to express that the domain is shared between the variables. NetCDF proposed the use of shared dimensions as a mechanism to capture this relationship. When two variables are defined over the same dimensions, they are understood to share the same domain [47].

```
netcdf example_file {
dimensions:
    time = 2 ;
    lat = 2 ;
    lon = 2 ;
    bnds = 2 ;
variables:
    float time(time) ;
        time:units = "days since 2024-01-01" ;
        time:calendar = "gregorian" ;
        time:axis = "T" ;
        time:standard_name = "time" ;
        time:bounds = "time_bnds" ;

    float lat(lat) ;
        lat:units = "degrees_north" ;
        lat:axis = "Y" ;
        lat:standard_name = "latitude" ;
        lat:bounds = "lat_bnds" ;

    float lon(lon) ;
        lon:units = "degrees_east" ;
        lon:axis = "X" ;
        lon:standard_name = "longitude" ;
        lon:bounds = "lon_bnds" ;

    float time_bnds(time, bnds) ;
    float lat_bnds(lat, bnds) ;
    float lon_bnds(lon, bnds) ;

    float height ;
        height:units = "m" ;
        height:axis = "Z" ;
        height:positive = "up" ;
        height:standard_name = "height" ;

    double tas(time, lat, lon) ;
        tas:units = "K" ;
        tas:coordinates = "height" ;
        tas:standard_name = "air_temperature" ;
        tas:cell_methods = "area: time: mean" ;
        tas:comment = "near-surface (usually, 2 meter) air temperature" ;

    double pr(time, lat, lon) ;
        pr:units = "kg m-2 s-1" ;
        pr:standard_name = "precipitation_flux" ;
        pr:cell_methods = "time: mean" ;

// global attributes:
    :title = "Example Climate Data" ;
    :gcm_id = "Naive GCM" ;
    :contact = "Ezequiel Cimadevilla" ;
}
```

Listing 3.1: Text representation of a NetCDF dataset which ignores the values of the variables for legibility. The dataset contains two field variables (tas and pr) and several support variables (time, lat, lon, time_bnds, lat_bnds, lon_bnds, and height). Relationships between field variables and support variables are represented using both NetCDF shared dimensions and the coordinates attribute.

HDF5 introduced the dimension scale API in order to offer a mechanism for the implementation of NetCDF shared dimensions (2005). The dimension scale API is based on HDF5 attributes, rather than on adding additional structure types to the data model. HDF5 user attributes (*NAME*, *CLASS*, *REFERENCE_LIST*, and *DIMENSION_LIST*) are reserved for the special purpose of describing the relationships between the HDF5 variables that describe NetCDF shared dimensions and the ones that describe physical and support variables.

Although the introduction of the dimension scale API into the HDF5 library provided a mechanism for NetCDF to adopt the advantages of HDF5, this implementation contains deficiencies regarding data integrity. For example, the HDF5 library does not enforce a consistent state of a file that makes use of dimension scales. An application that modifies the dimension scales in an incorrect manner will be allowed to do so without HDF5 providing any error information, leaving the file and the propositions stored within in an inconsistent state. Moreover, it should be noted that attributes are no longer dedicated solely to describing information about the values of a multidimensional array. They are now overloaded with the purpose of describing relationships between variables too. This overloading of attributes is common in NetCDF metadata standards.

3.4.2 Climate and Forecast Metadata Conventions

Metadata is essential in climate data storage, enhancing usability, interpretation, and reliability by providing contextual information about variables, units, spatial references, and methodologies. Without it, datasets would lack clarity, making accurate analysis difficult. Because the names and values of attributes may be chosen differently and arbitrarily by different users or communities, several conventions and standards have emerged to define how data in NetCDF files should be structured and interpreted. Climate metadata standards ensure consistency and interoperability across scientific disciplines by standardizing variable names, units, and coordinate systems. These standards enable seamless data integration, accurate comparisons, and long-term preservation while supporting reproducibility and accessibility for researchers and advanced analytical tools in climate data science.

The *Climate and Forecast (CF) conventions* (Eaton et al.) define metadata that clearly describe the meaning of each variable and its spatial and temporal properties. Different metadata schemas allow mapping from the different scientific data types to NetCDF structure types, enabling discovery and interoperability between applications. This ensures that data from different sources can be accurately compared and enables the development of applications with advanced extraction, regridding, and visualization capabilities. HDF5, Zarr, and NetCDF attributes play the critical role of storing metadata. Two types of attribute usage can be distinguished: the first is to provide information about the values of the multidimensional array to which the attribute is attached, such as the CF *units* attribute, and the second is to establish relationships between different multidimensional arrays contained in a dataset, such as the CF *coordinates* and *bounds* attributes. Figure 3.14 provides a conceptual illustration of the cell boundaries (*bounds*), used to delimit the boundaries of grid points in gridded data.

3.4.3 Relating Field and Support Variables

The Grid scientific data type refers to both field variables and support variables that are stored in a NetCDF dataset as multidimensional arrays with attributes. Field variables

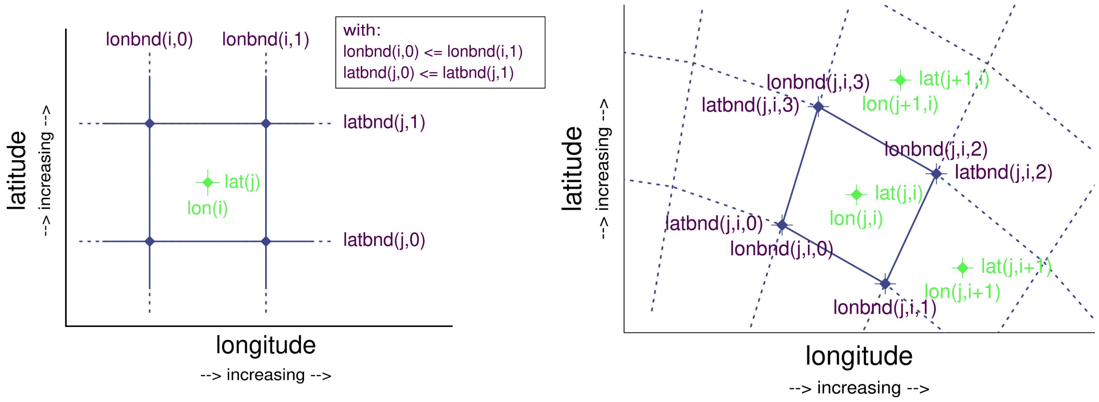


Figure 3.14: On the left: Arrangement of longitude and latitude bounds, for a one-dimensional horizontal coordinate system. The tuples (lon(i), lat(j)) represent the centers of grid cells, while the four vertices of each grid cell are defined by (lonbnd(i,0), latbnd(j,0)), (lonbnd(i,1), latbnd(j,0)), (lonbnd(i,1), latbnd(j,1)), and (lonbnd(i,0), latbnd(j,1)). On the right: Arrangement of longitude and latitude bounds for a two-dimensional horizontal coordinate system. The tuples (lon(j, i), lat(j, i)) represent the centers of grid cells, while the vertices of each grid cell are given by (lonbnd(j, i,n), latbnd(j, i,n)). **Source:** Figure adapted from `https://doi.org/10.5281/zenodo.14275599` (last accessed: 11 June 2025).

include information on physical variables such as surface temperature, precipitation, or wind speed. Support variables include information about the spatio-temporal domain in which the field variables live. Examples of support variables are spatio-temporal coordinates such as time, latitude, or longitude.

The modeling of relationships between field and support variables in a NetCDF dataset is one of the main sources of complexity in climate data management. For example, consider the *coordinates* attribute. It is common to find this attribute used in field variables, such as surface temperature or precipitation, for describing which NetCDF variables act as support variables and allow for locating the field variable in both space and time, although discrete information can also be provided, such as an ensemble model member.

In the NetCDF example shown in Listing 3.1, the *tas* variable represents surface temperature data. It uses the *coordinates* attribute to indicate that the NetCDF variable *height* specifies the vertical position of the temperature data. However, there is no need to include *lat*, *lon*, or *time* in the value of the attribute, since their relationship is already inferred through shared dimensions and *coordinate variables* (variables whose names match the dimensions they describe). This dual mechanism of inferring relationships between NetCDF variables is not only unnecessarily complex and unintuitive; it also exemplifies how the use of shared dimensions increases complexity by conflating physical and logical levels of abstraction.

Why not simply create a four-dimensional variable to store temperatures and let the *coordinate variables* define all coordinates? Because this would require storing a four-dimensional array on disk, where the vertical axis would contain only one value. Such a representation would unnecessarily complicate the underlying HDF5 structure. Moreover, it can be appreciated how the purpose of shared dimensions is undermined. Since attributes are still required to represent relationships among NetCDF variables, shared dimensions are effectively redundant and could be removed, thereby following the desired principle of reducing complexity while preserving exactly the same functionality.

It is worth noting that three different methods have already been identified to represent a simple relationship between climate variables and their supporting variables, each pathological in its own way. First, HDF5 dimension scales rely on physical pointers (HDF5 object references), which attributes then use to reference other variables within the file. Second, NetCDF employs shared dimensions, a problematic construct that conflates logical and physical abstraction levels while still requiring attributes to define the same relationships. Finally, the CF conventions rely on the *coordinates* attribute, which depends on *string formatting* and lacks integrity constraints to enforce the validity of its values.

The complexities of climate data storage discussed in this chapter, in particular those dedicated to establishing relationships between field and support variables, are far from trivial. It is common to encounter NetCDF files that cannot be correctly interpreted by applications due to missing metadata or attributes necessary to accurately understand the scientific data stored in a NetCDF dataset. In this context, *quality checkers* have emerged as tools designed to identify and report improperly structured NetCDF files before they are published. In the current state of the art, climate data analysis users and applications are responsible for handling data manipulation and integrity concerns that may arise when working with climate data.

3.4.4 NetCDF Java and the Common Data Model (CDM)

The Unidata Common Data Model (CDM) is an abstract framework that provides a unified way to represent, access, and interpret scientific datasets across multiple storage formats. Its main utility lies in offering a consistent interface for heterogeneous data sources, enabling applications to seamlessly handle NetCDF, OPeNDAP, HDF5, and other data formats.

The NetCDF Java library serves as an implementation of the CDM, enabling the reading of multiple file formats beyond NetCDF. These files, referred to as CDM files, are defined as any data files that can be accessed via the CDM data model through the NetCDF

Java library. The CDM also contributes logical manipulation rules for NetCDF datasets. These features will be detailed in section 4.4.1.

The CDM is the underlying data abstraction that NetCDF Java uses to provide a consistent interface across various data formats. It organizes data into a structured model that includes dimensions, variables, attributes, and coordinate systems. Unlike the traditional NetCDF model, which primarily handles array-based data storage, the CDM introduces richer semantics by explicitly defining coordinate systems, feature types, and transformations. This allows scientific data to be interpreted correctly in terms of geospatial and temporal relationships. The CDM defines three hierarchical layers:

1. Data Access Layer (also known as the *syntactic layer*) - Responsible for data reading and writing.
2. Coordinate System Layer - Specialized georeferencing coordinate systems are identified, which are particularly significant for the climate community.
3. Scientific Feature Types Layer - Categorizes data into specific types, such as Grid, Radial, and Point data, while introducing specialized methods for each type.

These three layers together establish a robust abstraction for climate data access. In practice, the combination of NetCDF-4 and the CDM has become a stable foundation in climate data-intensive science, ensuring interoperability and long-term usability of datasets across diverse platforms and tools. NetCDF Java plays a central role in this ecosystem being the main implementation of the CDM: it not only enables applications to work transparently with heterogeneous formats but also enforces consistent semantics for geospatial and temporal data. Several technologies of this work rely on NetCDF Java to access and offer climate data. They will be discussed in Section 4.2.2.1 and Section 4.3.2.

Chapter 4

Climate Data Analysis

4.1 Overview

Climate data stored using the different storage technologies presented in Chapter 3 requires analysis tools to extract knowledge from it. Moreover, it is desirable that these analyses be available to the scientific community to meet the objective of reproducibility, which also requires that the data meet the FAIR principles. Analyzing climate data involves examining historical, contemporary, and simulated future records to identify trends, patterns, and anomalies in temperature, precipitation, humidity, and other climate variables. This process involves the application of various statistical and computational methods to analyze large datasets collected from climate models and observational systems.

A fundamental aspect of climate data analysis is detecting trends in global and regional temperature changes. For example, time series reveal trends, variability, and abrupt shifts in those variables at global and regional scales. Figure 4.1 shows time series of historical and future simulated changes in four global climate indicators from CMIP6 [Coupled Model Intercomparison Project Phase 6; 13, 14], including global surface air temperature (GSAT), global land precipitation, September Arctic sea-ice area, and global mean sea level (GMSL).

These trends are further assessed through comparative studies in different geographic locations to understand spatial variations and contributing factors. The IPCC AR6 WGI *Interactive Atlas* offers interactive map and time-series exploration tools that allow users to compare observed and modeled changes by region, variable/index, time window, and emissions scenario.

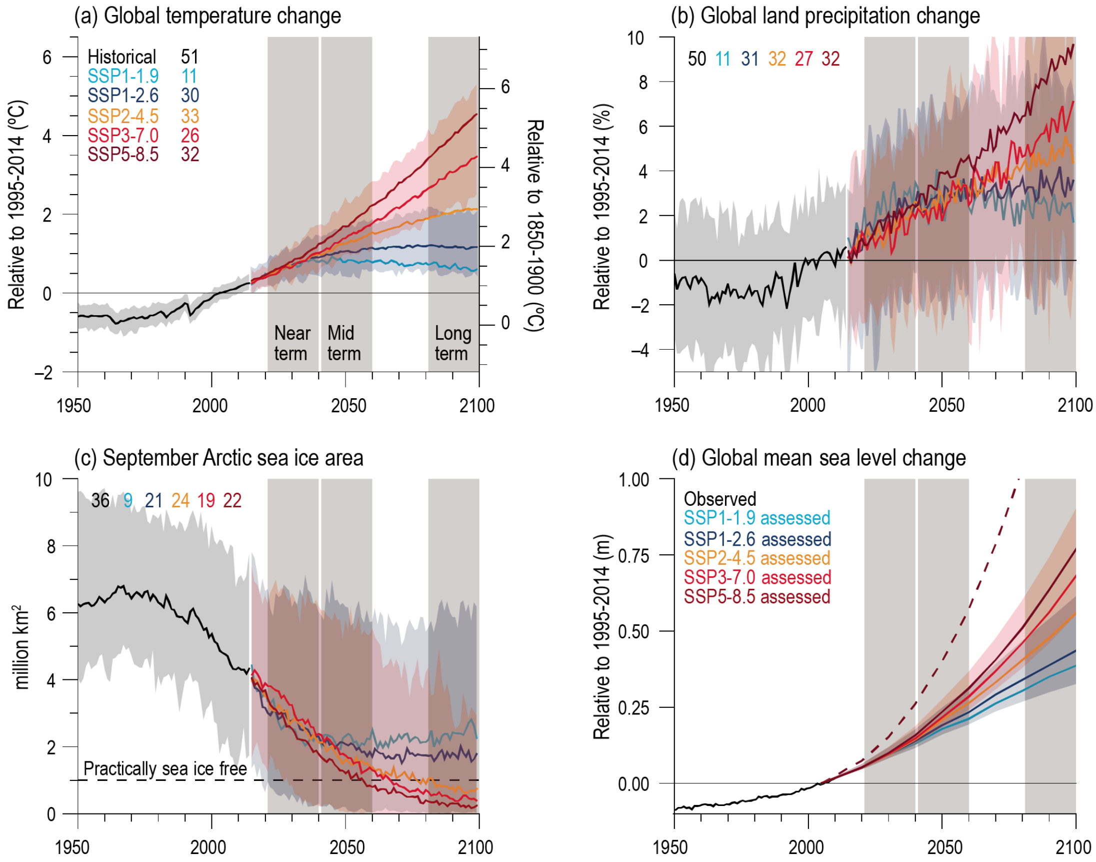


FIGURE 4.1: Time series of historical and future simulated changes in key global climate change indicators from CMIP6 historical and scenario simulations: (a) Global surface air temperature changes relative to the 1995–2014 average (left axis) and relative to the 1850–1900 average (right axis; offset by 0.82°C). (b) Global land precipitation changes relative to the 1995–2014 average. (c) September Arctic sea ice area. (d) Global mean sea level (GMSL) change relative to the 1995–2014 average. **Source:** Figure 4.2 in Lee et al., 2021.

These trends are further assessed through comparative studies in different geographic locations to understand spatial variations and contributing factors. The IPCC AR6 WGI *Interactive Atlas* [26] offers interactive map and time-series exploration tools that allow users to compare observed and modeled changes by region, variable/index, time window, and emissions scenario. Climate data analysis workflows, which result in outputs like those shown in the Interactive Atlas and Figure 4.1, currently involve complex programming tasks that are typically performed using common programming languages in climate data science, such as Python and R [52]. Due to current practices of climate data distribution, based on the distribution of binary file formats, programming tasks for climate data analysis require the location and manipulation of either NetCDF files or Zarr stores, both subject to the same programming complexities.

Two types of climate data manipulation or inference can be distinguished: logical inference and statistical inference. Both are commonly encountered in any climate data analysis task but differ fundamentally in their intentions and methods in deriving conclusions from data.

Logical inference relies on formal logic and deterministic rules, meaning that if the premises are true, the conclusion must also be true. This approach is commonly used in data management for tasks like database queries and constraint validation. For instance, if a rule states that every spatio-temporal cell of a gridded dataset must have a unique value of temperature, and two values of temperature are provided for the same spatio-temporal cell, logical inference can determine that a data integrity violation has occurred.

In contrast, statistical inference operates on descriptive or probabilistic principles, drawing conclusions based on patterns and uncertainty within data. It is used to generalize insights from a sample to a larger population, derive descriptive statistics about a database, or optimize machine learning models. Both logical and statistical inferences, as well as data infrastructures where data analysis is carried out, play crucial roles in the technological solution to data-intensive science, often appearing together in various data-intensive science tasks.

For example, merging gridded surface temperature records that are split into multiple files along the time coordinate requires logically combining the time series into a single unit, which represents an operation of logical inference. In contrast, computing the spatial mean of that time series across spatial dimensions involves statistical inference. Additionally, climate data analysis applications are responsible for providing the capability to generate plots, aiding in the visualization of results of both logical and statistical inferences.

This chapter covers the various ways in which climate data analysis workflows can be carried out, including different protocols for data access, how applications currently handle the complexities of both logical and statistical inferences, and how applications can be relieved of these complexities through the use of analysis-ready data (ARD).

4.2 Analysis Workflows

Climate data analysis has traditionally involved downloading binary files to local workstations or infrastructure for subsequent analysis. However, this approach poses significant challenges for analysts, requiring considerable time to manage data access tasks that often divert attention from core research objectives. Climate data infrastructures, especially those dedicated to climate model intercomparison projects, currently hold NetCDF files in the order of the millions. In the era of data-intensive science [7], datasets have become increasingly complex [53], rendering traditional methods of analysis inefficient. This inefficiency is evident not only in the analysis of Earth System Model Output (ESMO) but also in the field of Earth Observation [54, 55]. The prevailing challenge lies in transforming vast quantities of existing data into valuable information to ensure proper and efficient utilization. Three methods by which climate data analysis can be performed are identified:

- *Download-and-analyze* - Involves transferring files and storing from a server to a local workstation, previous to any analysis task.
- *Remote data access* - In comparison to file downloads, the strength of remote data access lies in its built-in subsetting capabilities, allowing users to analyze data without downloading entire files.
- *Next-to-data computing* - This method involves server-side computing adjacent to the datasets, so only results or small subsets are transferred to the client.

4.2.1 Download and Analyze

File downloads are the simplest method upon which climate data analysis has traditionally been based. They offer several advantages, particularly in terms of accessibility and flexibility. Researchers can work offline, apply customized processing techniques, and use their preferred software without relying on an internet connection. This method is especially useful when dealing with small datasets or specific subsets of large datasets that fit within local storage and computing capacities. It also allows researchers to perform in-depth, iterative analyses without concerns about server restrictions or remote access limitations.

Challenges in the download-and-analyze workflow arise as the number of files and their size begin to increase, which can occur due to high-resolution climate models or multiple model runs, for example. In these cases, the amount of data can grow to the order of terabytes or even petabytes, making downloads impractical due to bandwidth limitations and storage constraints. Additionally, managing and organizing large collections of downloaded files can be cumbersome, requiring efficient data management practices to ensure consistency, version control, and reproducibility of research.

Although it is easy to set up for climate data providers, it places a significant burden on the users, who must be able to locate and transfer potentially large numbers of files. This task may or may not be trivial, depending on the number of files and the care that the data provider has taken in organizing and presenting the files. This method becomes time-consuming for large datasets and often inefficient for interactive or exploratory analyses.

Despite these challenges, file downloads remain a fundamental method for climate data analysis. Many research institutions and data providers continue to offer bulk download options, often complemented by metadata, documentation, and preprocessing tools to assist researchers in handling the data efficiently. However, as climate data continues to grow in volume and complexity, alternative methods, such as remote data access and server-side computing, are becoming increasingly important for scalable and efficient analysis.

4.2.2 Remote Data Access

Remote data access has become an increasingly popular method for climate data analysis as datasets grow larger and more complex. Unlike file downloads, where data is transferred to a local machine, remote data access allows researchers to query and analyze data directly on remote servers or cloud platforms without the need to download the entire dataset to their local system. Remote data access is assumed to provide implicit subsetting capabilities, which is the mechanism that avoids the download of entire files. Thus, spatial sections of a larger domain or specific time periods of a larger time series can be analyzed without the need to download the files containing the full spatio-temporal domain.

By eliminating the need to download files, remote data access eliminates the need for local storage management, which can become a significant challenge when dealing with large climate data. Researchers can access datasets on-demand, enabling them to work with the most up-to-date versions of the data without worrying about managing multiple local copies. Remote data access also reduces the risk of overloading local storage and computing resources, as data remains hosted on remote servers. This approach is particularly useful for interactive or exploratory analysis, where users may need to quickly retrieve specific data subsets or perform complex queries without waiting for large-scale downloads. Moreover, when combined with Analysis Ready Data (ARD, [56]), remote data access greatly improves the data analysis experience 7. Remote data access takes different forms: via middleware that implements client–server protocols (e.g., **OPeNDAP**), or via storage technologies with native remote access (e.g., **Zarr**).

4.2.2.1 OPeNDAP

An example of remote data access based on client-server middleware is OPeNDAP (Open-source Project for a Network Data Access Protocol) [57], an open-source protocol designed to facilitate the efficient transfer and manipulation of scientific data, making it particularly valuable for researchers working with large-scale, high-resolution climate datasets. OPeNDAP works by allowing remote clients (such as data analysis software, programming languages, or web-based platforms) to query and retrieve specific subsets of data stored on remote servers. The data remains hosted on the server, while only the necessary portions of the dataset are transmitted to the client for processing. It supports a wide variety of data formats commonly used in climate data science, including NetCDF and HDF5, among others. This broad compatibility allows researchers to access data from different sources and work with it in a consistent and standardized way.

OPeNDAP operates on a client/server model. Clients send data requests over the network to servers, which respond with the requested data. This architecture is analogous to the World Wide Web, where browsers request web pages from servers. Basic to the operation of OPeNDAP is its data model and the set of messages that define the communication between client and server. At its core, DAP employs an intermediate data representation, which serves as a transport mechanism for moving data from the remote source to the client. This representation forms the basis of the OPeNDAP data model, encompassing various data types that enable interoperability between different scientific datasets.

To ensure effective data translation, DAP incorporates a standardized format for ancillary data, which is crucial for converting datasets into the intermediate representation and subsequently mapping them onto the target data model. The ancillary data consists of two primary elements: the Data Descriptor Structure (DDS) and the Data Attribute Structure (DAS). The DDS provides a detailed description of the shape and size of the data types stored within a given dataset, ensuring that the structural aspects of the data are well-defined. Meanwhile, the DAS offers capsule descriptions of key properties of the data, capturing metadata that informs users about dataset attributes, conventions, and meanings.

To facilitate seamless interaction with OPeNDAP-enabled data sources, DAP also provides an API. This API consists of OPeNDAP classes and specialized data access calls designed to implement the protocol, allowing users to programmatically access, manipulate, and integrate data within their computational workflows. The API plays a critical role in enabling cross-platform compatibility and expanding the accessibility of scientific datasets across diverse research communities.

A particular provider of an OPeNDAP implementation is the THREDDS Data Server (TDS), which builds upon NetCDF-Java and the CDM 3.4.4 to provide a web-based data distribution system [58]. TDS enables users to access scientific datasets remotely via standard protocols such as OPeNDAP, HTTP, and WMS (Web Map Service). By leveraging the CDM, TDS can serve data in multiple formats while preserving the semantic structure of the original dataset. TDS also integrates metadata services, providing users with detailed information about datasets through catalogs and search interfaces. This makes it an essential tool for sharing large-scale scientific data with the research community while maintaining efficient and scalable access methods.

4.2.2.2 Zarr

In contrast to OPeNDAP, remote data access with Zarr can be provided without the need for server-side middleware, simplifying the architecture of systems that offer remote data access capabilities. As discussed in Section 3.3, Zarr owes part of its success to its remote data access features, which are highly compatible with the object storage systems commonly used in cloud infrastructures. The Zarr format and specification define clear locations for metadata required to extract information from multidimensional arrays. These locations include the `.zmetadata` endpoint for consolidated Zarr stores, and the `.zgroup`, `.zarray`, and `.zattrs` endpoints for regular Zarr stores.

By standardizing the locations of metadata resources, Zarr enables remote data access without requiring any server-side middleware to handle these operations. Any Zarr client can retrieve the necessary metadata and compute the locations of data chunks using the algorithm described in Section 3.3.

The Zarr approach to storing and retrieving data from multidimensional arrays offers several performance benefits. When accessing data remotely via OPeNDAP, some of the advantages of the underlying storage format can be lost, as OPeNDAP treats storage formats as transparent components. For instance, if an HDF5 file is stored with compression, OPeNDAP requires HDF5 to decompress the data chunks before transmission, resulting in uncompressed data being sent to the client. This unnecessarily increases the volume of data transferred over the network. In contrast, Zarr allows clients to request compressed chunks directly from the server, with decompression performed locally on the client side. In the context of Big Data, particularly in climate data science, this represents a significant improvement in both performance and efficiency.

4.2.3 Next-to-Data Computing

Remote data access improves the data analysis experience by eliminating the need for file downloads; however, computation is still performed on the user's local workstation, and the data must be transferred to the host carrying out the analysis task. The main advantage of next-to-data or server-side computing is the significant reduction in network traffic, as computations are performed directly on the server where the data is locally stored.

However, this approach imposes a considerable burden on the data provider, who must maintain the necessary computational infrastructure. Additionally, the service provider must ensure security concerns and may need to implement user-requested data-analysis functions, while also managing access controls for the underlying infrastructure. Server-side processing can be implemented through HTTP middleware that handles data-processing requests, active storage solutions, or multi-user platforms (JupyterHub). Some computations may still need to be executed on the user's workstation, such as visualization tasks like plotting.

JupyterHub is a widely used server-side computing environment that enables researchers to run Jupyter Notebooks on remote servers. Unlike traditional Jupyter Notebook setups that run on local machines, JupyterHub provides a multi-user interface where multiple researchers can access shared computational resources. This is particularly beneficial for climate data science tasks that deal with large datasets and complex models, as it allows them to use HPC resources or cloud-based infrastructure without the need for local installations.

One of the key advantages of JupyterHub is its ability to integrate with cloud storage and remote data services. Researchers can access climate datasets stored on remote servers, perform computations, visualize results, and share their workflows with collaborators—all within a single interface. Additionally, since the execution happens on powerful remote machines, users are not constrained by the limitations of their local hardware. JupyterHub environments preconfigured with climate data libraries enable seamless and efficient analysis with a low learning curve.

Other solutions that follow the Next-to-Data computing approach exist and are, in fact, adopted to varying degrees by climate data infrastructures. However, they do not reach the same level of community standardization as Jupyter-based solutions, and they tend to be more ad-hoc implementations rather than generic contributions to Next-to-Data computing. An example in the context of climate data-intensive science is Web Processing Services (WPS) [59], which allow remote operations on climate simulations: a client sends a request to a server, the computation is performed on the server, and the result is then returned to the client. In this thesis, the focus is placed on Next-to-Data solutions based on JupyterHub.

4.3 Libraries and Toolchains

A wide range of specialized tools and libraries has been developed to facilitate climate data analysis, leveraging modern computational techniques to efficiently handle large datasets.

These applications provide capabilities for data ingestion, transformation, visualization, and statistical analysis. As explained in Section 2.1.1, this thesis distinguishes between logical inferences and statistical inferences. Since file downloads (see Section 4.2.1) remain the most common method for climate data analysis, and NetCDF is the dominant file format, tools for climate data processing and analysis must handle significant complexity arising from logical and statistical inferences. In Section 4.4, a discussion is provided on how analysis-ready data can reduce the complexities of performing logical inferences for climate data analysis applications. Two libraries are now presented that illustrate current practices in climate data analysis. Xarray, a Python library for the analysis of labeled multidimensional arrays and Climate4R, a framework that offers an R-based, domain-oriented environment specifically tailored to climate data access and analysis.

4.3.1 Xarray

Xarray is a Python library designed for working with multi-dimensional labeled datasets, making it a widely used tool for climate data analysis [60]. Based on the NetCDF data model, it seamlessly integrates with climate data storage libraries commonly used in climate data science. Xarray supports lazy loading of data and integrates with Dask, a parallel computing library, allowing users to process large datasets without exceeding local memory limits. It also includes built-in methods for *group by* operations and resampling, facilitating climatological calculations such as monthly or seasonal means. Additionally, Xarray easily integrates with libraries such as Matplotlib, Cartopy, Pandas, and Scikit-learn, enabling the generation of climate maps, time-series plots, and statistical models.

Xarray includes several built-in backends that support many common data formats. Additional backends are available through external libraries, and users can also develop their own. Xarray provides backends for NetCDF, HDF5, and Zarr, each based on different Python libraries. For example, NetCDF-4 datasets are opened using `NetCDF4-python`, HDF5 datasets with `h5netCDF`, and Zarr datasets with the `zarr` Python library.

Xarray also supports the user by providing functions that manage many common logical inferences that are commonly required in climate data analysis tasks. These functions allow users to combine multiple datasets or variables that span different dimensions, such as time, spatial coordinates, or model runs. An example of this includes xarray's [60] `open_mfdataset` function and software applications for climate data analysis. Listing 4.1 provides an example of the usage of the `open_mfdataset` function.

```
ds=xr.open_mfdataset(
  sorted(glob.glob(
    "/storage/ESGF/CMIP6/.../tas_3hr_BCC-CSM2-MR_historical_r1i1p1f1_gn_*.nc")),
  combine="nested",
  concat_dim=["time"])
```

Listing 4.1: Usage of xarray's *open_mfdataset* to generate an ARD dataset at the application layer from several NetCDF files.

The `concat` function in Xarray is used to concatenate multiple datasets along an existing dimension, such as time, latitude, or longitude. This function is often used after loading datasets or when individual datasets are already in memory but need to be combined along a specific axis. It provides flexibility when merging data arrays and ensures that the resulting dataset aligns correctly along the concatenating dimension.

The `merge` function in Xarray is used to merge two or more datasets (or data variables) with different coordinate dimensions or variables. Unlike concat, which merges datasets along an axis (e.g., stacking them), merge is used for combining datasets with overlapping coordinates or variables but potentially different dimensions. This is particularly useful when datasets contain different variables that share a common set of coordinates.

4.3.2 Climate4R

Climate4R is an open-source suite of R packages designed for climate data retrieval, analysis, and visualization [61]. Developed by the Santander Meteorology Group, Climate4R provides an end-to-end workflow for processing climate datasets. These operations include common transformation, calibration, and post-processing steps that are typically applied to raw model data before their use in sectoral applications. These steps include data collocation (e.g., regridding, temporal aggregation, or subsetting) and bias adjustment or downscaling (e.g., local scaling, quantile mapping, analogs, or regression).

Climate4R has been developed as a set of seamlessly integrated packages designed to ease climate data access (`loadeR`), collocation and transformation (`transformeR`), bias correction and downscaling (`downscaleR`), and visualization (`visualizeR`). It also includes full documentation via wikis and guided examples. Figure 4.2 illustrates the components of Climate4R.

Climate4R benefits from the use of the NetCDF-Java library and the CDM it implements 3.4.4. Thus, it is able to operate with all CDM-compatible datasets. Moreover, it can take advantage of virtually aggregated datasets 4.4.1 and remote data access protocols 4.2.2 such as OPeNDAP, which are directly supported by NetCDF-Java.

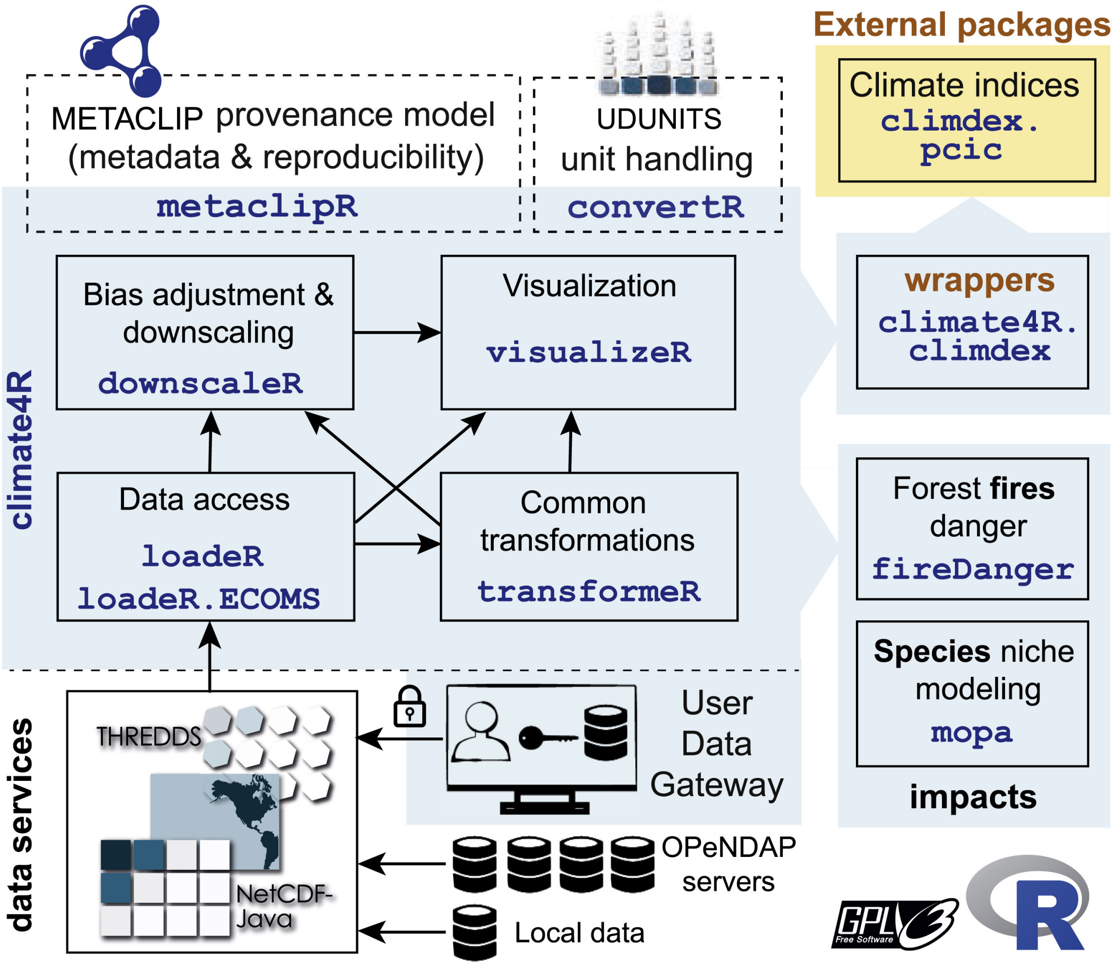


FIGURE 4.2: Description of the core R packages of the Climate4R framework. **Source:** Iturbide et al., 2019.

4.4 Analysis Ready Data (ARD)

Analysis Ready Data (ARD) refers to datasets that have been pre-processed and structured to facilitate immediate use in analysis, eliminating the need for significant preparatory work. This need arises mainly due to the complexities of dealing with the current state of the art of analyzing NetCDF files or Zarr stores, in particular, the manipulation of data that involves logical inferences. To facilitate climate data analysis, various methodologies are currently under consideration, based on either aggregations of the original datasets and/or transition to new infrastructures such as cloud providers. Aggregation-based approaches focus on creating either physical or *virtual* aggregations of data, optimized for efficient analysis, thereby relieving users from the intricacies of directly manipulating NetCDF files and Zarr stores. ARD may be provided as a physical copy of the original data [44] or relying on virtual aggregation techniques that save the cost of duplicating data 7.

ARD based on aggregations can be performed at different layers of abstraction and may involve varying levels of complexity depending on the desired outcome. Many approaches are based on data analysis applications offering functionality for abstracting the underlying files and hierarchical file system organization from the data user. Example of software applications that follow this pattern include CF-Python [62], xMIP, [63], intake-esm [64] and intake-esgf [65]. Xarray and its logical manipulation functionalities have already been discussed 4.3.1. In general, these approaches hold an in-memory representation of the virtual dataset or aggregation, which is manipulated by the data analysis package behind the scenes.

Software packages may provide a way to persist their aggregated logical view of the underlying files. However, these persistence formats are typically not interoperable between packages and often do not offer an interchangeable logical view of the aggregation. When generating aggregated views, data may either be duplicated or managed through virtual aggregations to avoid duplication. The advantage of relying on virtual dataset capabilities is that data duplication is avoided, and the existing infrastructure may be reused to obtain ARD capabilities without huge costs associated. Examples of virtual aggregations that follow this approach include (but are not limited to) NcML [66], Kerchunk [67], CFA [68], and HDF5 Virtual Datasets [34].

4.4.1 NetCDF Markup Language (NcML)

The NetCDF Markup Language (NcML) is an XML-based dialect designed for defining and modifying Common Data Model (CDM) datasets. NcML is implemented by the NetCDF-java library [66], which has already been mentioned 3.4. An NcML document, structured in XML, specifies the characteristics of a CDM dataset. Typically, an NcML references an existing dataset, known as the referenced CDM dataset, allowing for metadata modifications, virtual aggregations, and other enhancements without altering the original data. The purpose of NcML is to allow:

- **Metadata augmentation/deletion/correction** — add or fix global/variable attributes (names, units, CF tags) without rewriting the data.
- **Virtual aggregation** — define logical datasets by aggregating files (e.g., concatenation along time/ensemble), subsetting, or unioning variables.

Multiple CDM datasets can be aggregated into a single logical dataset using the aggregation NcML element. Several types of aggregation are supported:

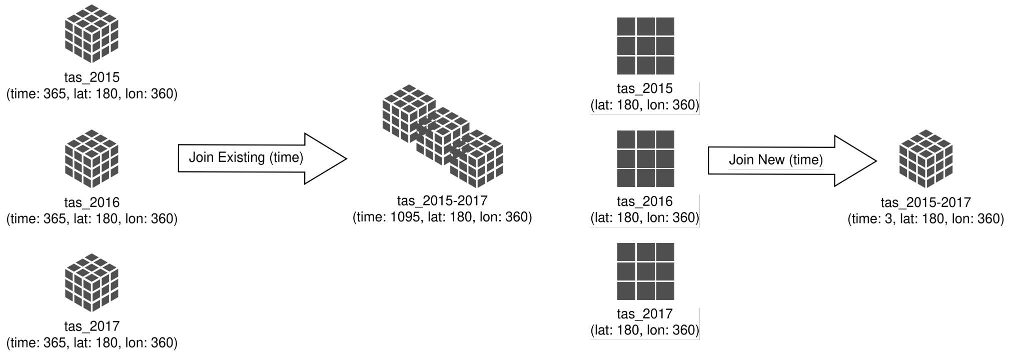


FIGURE 4.3: Illustration of both *join existing* and *join new* aggregations along the time dimension. In the case of the join exiting, the result of the aggregation is a multidimensional array with the same dimensions (*time*, *lat*, and *lon*), in which the size of the time dimension has increased. In the case of the join new aggregation, the time dimension is a new dimension created to aggregate the existing two-dimensional arrays into a new three-dimensional array.

- **Union**: The union of all dimensions, attributes, and variables across multiple NetCDF files.
- **Join Existing**: Variables with the same name across different files are concatenated along their existing outer dimension, referred to as the aggregation dimension. A coordinate variable must be present for this dimension. A join existing operation is illustrated in Figure 4.3.
- **Join New**: Variables with the same name in different files are concatenated along a newly created outer dimension. Each file represents a distinct entry in this dimension, and a new coordinate variable is generated. A join new operation is illustrated in Figure 4.3.
- **Forecast Model Run Collection (FMRC)**: A specialized aggregation method for collections of forecast model runs, utilizing two time coordinates: run time and forecast time.

NcML enhances the capabilities of climate data analysis applications (see Section 4.3) by providing logical views of NetCDF datasets that appear to users as a single data source. Moreover, it is compatible with remote data access protocols such as OPeNDAP [58]. As a result, users do not need to handle logical aggregations (logical inferences) at the application level and can instead focus on statistical inferences and the visualization of results. Listing 4.2 shows an NcML example.

```
<?xml version="1.0" encoding="UTF-8"?>
<netCDF xmlns="http://www.unidata.ucar.edu/namespaces/netCDF/NcML-2.2">
    <aggregation type="union">
        <netcdf>
        <aggregation dimName="time" type="joinExisting">
        <netCDF
         location="tas_day_BCC-CSM2-MR_rcp85_r1i1p1_gn_19500101-19521231.nc"/>
        <netCDF
         location="tas_day_BCC-CSM2-MR_rcp85_r1i1p1_gn_19530101-19551231.nc"/>
        <netCDF
         location="tas_day_BCC-CSM2-MR_rcp85_r1i1p1_gn_19560101-19581231.nc"/>
        </aggregation>
        </netcdf>
        <netcdf>
        <aggregation dimName="time" type="joinExisting">
        <netCDF
         location="pr_day_BCC-CSM2-MR_rcp85_r1i1p1_gn_19500101-19521231.nc"/>
        <netCDF
         location="pr_day_BCC-CSM2-MR_rcp85_r1i1p1_gn_19530101-19551231.nc"/>
        <netCDF
         location="pr_day_BCC-CSM2-MR_rcp85_r1i1p1_gn_19560101-19581231.nc"/>
        </aggregation>
        </netcdf>
    </aggregation>
</netCDF>
```

Listing 4.2: NcML file that showcases a logical aggregation by performing both a *union* and a *join existing* aggregation over several local NetCDF files. Special attention must be given to the compatibility of coordinate variables, ensuring they match across source NetCDF datasets to prevent inconsistencies in the aggregation process.

4.4.2 Kerchunk

Kerchunk is a library that provides a unified way to represent various chunked and compressed data formats, such as HDF5 and Zarr. It serves two different purposes that should be considered separately. First, it enables efficient data access from HDF5 datasets stored in cloud object storage. Second, it offers a flexible method for creating virtual aggregations from multiple files, regardless of the underlying format (HDF5 or Zarr).

Kerchunk enables efficient data access to HDF5 datasets stored in object storage from cloud providers by providing a Zarr interface to the underlying HDF5 dataset. It achieves this by preprocessing the source HDF5 dataset and extracting metadata - including chunk byte positions, chunk byte lengths, and other relevant details such as compression and dataset filter information. All this information is stored in a *sidecar* file, which Zarr-compatible applications, such as Xarray, can use to efficiently access HDF5 datasets stored in the cloud. This approach helps avoid latencies associated with fetching of HDF5 metadata, such as B-tree traversals needed to query chunk positions. Listing 4.3 shows a Kerchunk file that provides a Zarr interface to an underlying NetCDF/HDF5 dataset.

```
{
    ".zgroup": "{\n  \"zarr_format\": 2\n}",
    ".zattrs": "{}",

    "tas/.zarray": "{\n  \"chunks\": [1, 128, 256],\n  \"compressor\": {\n    \"
    level\": 4,\n    \"id\": \"zlib\"\n  },\n  \"dtype\": \"<f4\",\n  \"
    fill_value\": 1e+20,\n  \"filters\": [\n    {\n      \"elementsize\": 4,\n
       \"id\": \"shuffle\"\n    }\n  ],\n  \"order\": \"C\",\n  \"shape\": [2,
    128, 256],\n  \"zarr_format\": 2\n}",

    "tas/.zattrs": "{\n  \"_ARRAY_DIMENSIONS\": [\"time\", \"lat\", \"lon\"],\n
    \"units\": \"K\"\n}",

    "tas/0.0.0": [
        "https://example.com/example.nc",
        32822,
        73967
    ],

    "tas/1.0.0": [
        "https://example.com/example.nc",
        106789,
        73947
    ]
}
```

Listing 4.3: Illustrative Kerchunk JSON file providing a Zarr interface to an underlying NetCDF/HDF5 dataset. The dataset contains a single variable, "tas", with two chunks. Kerchunk records the location of each HDF5 chunk and maps them to the corresponding Zarr chunks. Chunk information includes the server hosting the chunk, the byte position where the chunk begins, and the length of the chunk in bytes.

Kerchunk can also be used to generate virtual aggregations similar to those generated by NcML. Listing 4.4 shows a Kerchunk file that performs a join existing aggregation of two NetCDF/HDF5 datasets hosted in the same server. Note that Zarr stores can also be aggregated using Kerchunk, where the chunk positions and length fields simply hold a value of zero. This approach allows the creation of virtual aggregate datasets spanning numerous source files, facilitating efficient, parallel, and cloud-friendly in-situ access without the need to copy or convert the original files [69].

```
{
    ".zgroup": "{\n  \"zarr_format\": 2\n}",
    ".zattrs": "{}",

    "tas/.zarray": "{\n  \"chunks\": [1, 128, 256],\n  \"compressor\": {\n    \"
    level\": 4,\n    \"id\": \"zlib\"\n  },\n  \"dtype\": \"<f4\",\n  \"
    fill_value\": 1e+20,\n  \"filters\": [\n    {\n      \"elementsize\": 4,\n
       \"id\": \"shuffle\"\n    }\n  ],\n  \"order\": \"C\",\n  \"shape\": [2,
    128, 256],\n  \"zarr_format\": 2\n}",

    "tas/.zattrs": "{\n  \"_ARRAY_DIMENSIONS\": [\"time\", \"lat\", \"lon\"],\n
    \"units\": \"K\"\n}",

    "tas/0.0.0": [
        "https://example.com/example1.nc",
        32822,
        73967
    ],

    "tas/1.0.0": [
        "https://example.com/example2.nc",
        106789,
        73947
    ]
}
```

Listing 4.4: Illustrative Kerchunk JSON file that generates a virtual aggregation by performing a *join existing* aggregation from two source NetCDF/HDF5 datasets. It is important to note that dataset filters such as compression are required to be identical across datasets for Kerchunk to generate the correct virtual aggregation.

4.5 Reproducibility and FAIR Principles

An increasingly important requirement in climate data analysis is reproducibility [70]. Reproducibility enhances scientific quality by enabling the replication of climate products and results generated through the analysis tools described in this section. Achieving full reproducibility in climate data analysis requires the careful integration of standardized computational environments, executable notebooks, and open access to data. The adherence to FAIR data principles — Findable, Accessible, Interoperable, and Reusable — ensures that climate data and climate data analysis products remain useful and widely available for scientific research and policy-making [27]. However, given the vast amounts of available data, ensuring reproducibility can be prohibitively expensive, which necessitates a compromise between methodological rigor and operational cost.

Several technologies are available to enhance reproducibility. By encapsulating software dependencies within containerized or virtualized environments, researchers can ensure that workflows remain stable over time and across platforms. Executable notebooks provide a transparent medium for combining narrative, code, and results, thereby supporting both methodological clarity and reproducibility. The adoption of ARD further reduces heterogeneity in the pre-processing steps, facilitating comparability between datasets. It should be noted that, despite the advantages of Jupyter Notebooks in enhancing reproducibility through interactive literate programming, by providing documents that combine live code, equations, visualizations, and narrative text, they do not guarantee reproducibility [71].

The limitation in reproducibility arises because Jupyter Notebooks depend not only on the surrounding climate data infrastructure—for example, for data availability and access—but also on the software environment in which they are executed. Even when stored in Docker containers, notebooks may become non-reproducible over time due to deprecated repositories or changes in software versions (e.g., Python packages and system libraries). Ensuring long-term reproducibility therefore requires active maintenance of the execution environment and clear documentation of software dependencies.

The degree to which reproducibility can be realized is inherently constrained by the resources allocated to a climate data infrastructure. For climate data infrastructures that function primarily as repositories and provide storage without advanced computational or analytical services, the scope for reproducibility is necessarily limited. Such infrastructures ensure the persistence, accessibility, and integrity of datasets, which are essential prerequisites for reproducibility, but they do not independently enable the re-execution of workflows or the replication of results. In these contexts, reproducibility relies heavily on the quality of metadata, adherence to community standards for data formatting and documentation, and the extent to which provenance information is preserved alongside the stored datasets. Consequently, while storage-only infrastructures play a foundational role by safeguarding the raw materials required for reproducibility, they must be complemented by additional platforms or tools to support the full spectrum of reproducible research practices.

In contrast to storage-only infrastructures, tools such as the Interactive Atlas represent web-based interfaces that provide users with access to a processed subset of climate data. These platforms are not designed to serve as comprehensive repositories but rather as gateways to curated, aggregated, or visualized information derived from underlying climate data infrastructures. While such interfaces greatly enhance accessibility and interpretability for a broad audience, they present important limitations with respect to reproducibility. In Chapter 5, the principal categories of climate data infrastructures are delineated, along with an examination of their reproducibility capacities.

Chapter 5

Climate Data Infrastructures

5.1 Overview

Data infrastructures are systems, technologies, and frameworks that support the collection, storage, processing, and distribution of data. They support climate data storage and analysis by including hardware components, such as cloud storage and high-performance computing clusters, and software tools, including data management platforms, storage and analysis libraries, APIs, and machine learning frameworks. The primary goal of data infrastructures is to ensure that data are efficiently managed, accessible, and usable for research. Data management refers to the processes of organizing and maintaining these infrastructures to guarantee the accuracy, accessibility, and long-term usability of their data.

Climate data infrastructures are specialized systems for collecting, storing, processing, and distributing climate-related information. Such data are valuable only when they are accessible to researchers and the public. These infrastructures provide standardized access to climate information, bridging gaps between institutions and geographic regions. By offering centralized or federated repositories, they ensure that data can be shared widely and efficiently. Additionally, climate data infrastructures supply the computational power, tools, and frameworks necessary for advanced analyses, driving progress in climate data science.

Managing the vast volumes of climate data is nothing but trivial, as traditional infrastructures lack the necessary capabilities for storage and analysis, particularly given the ever-increasing amounts of data being generated over time. In the context of climate data science, the volume of available data has grown at an unprecedented rate, a trend that is expected to continue. Figure 5.1 shows a projection from 2011 of the expected amount of

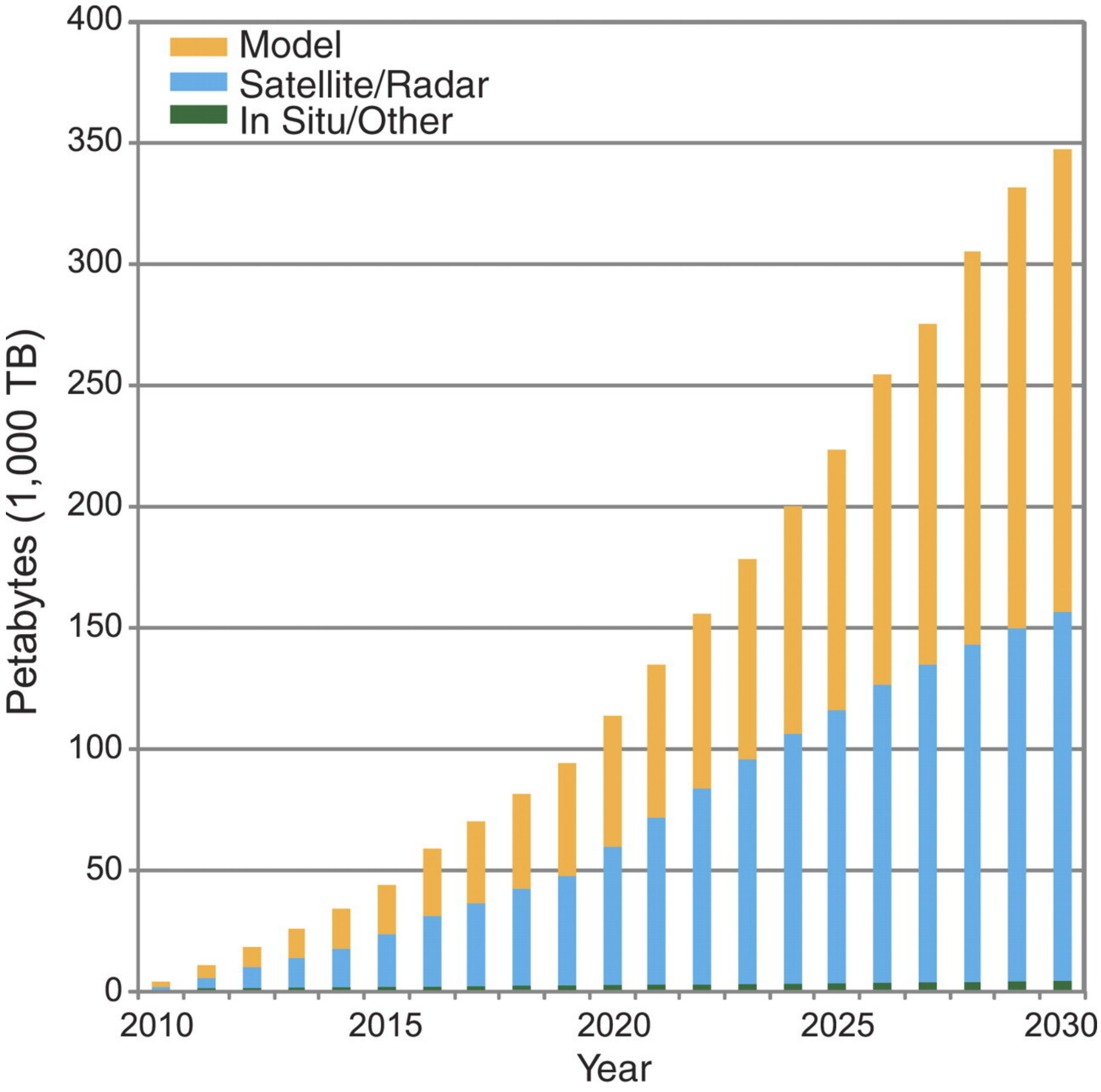


FIGURE 5.1: Projected increase in global climate data holdings for climate models, remotely sensed data, and in situ instrumental/proxy data provided by [10]. **Source:** Overpeck et al., 2011.

meteorological and climate data to be collected in the coming years. Unfortunately, the capacity to store, process, and analyze such amounts of available data has not increased at the same rate.

Scientific research is increasingly relying on advanced data infrastructures to support computational tasks, data management, and interdisciplinary collaboration. Scientific infrastructures enable researchers to process large datasets, simulate complex phenomena, and drive innovation on an unprecedented scale. To manage the data deluge in the field of climate data science, these infrastructures must support both petabytes of data storage and petaflops of processing power. This thesis distinguishes between two types of infrastructure: those oriented towards High-Performance Computing (HPC) and those designed for High-Throughput Computing (HTC). The distinction lies in where the hardware bottleneck occurs: HPC systems to be dedicated to CPU-bound applications, whereas HTC systems are dedicated to I/O-bound applications.

HPC infrastructures are usuallly used for simulating Global Climate Models (GCMs). These simulations demand intensive CPU usage due to the high computational cost of solving the differential equations governing the climate system. In contrast, HTC infrastructures in the same field are typically employed for analyzing data generated by GCMs, such as performing time series analysis. Since these statistical operations are relatively lightweight computationally, the primary bottleneck occurs at the I/O level, which must be fast enough to supply data efficiently to the CPU.

Research infrastructures within the scope of climate data science usually fall into the category of HPC systems. These infrastructures place significant emphasis on the CPU/GPU of the system due to the computational demands of running GCMs and/or RCMs and training machine learning models. At the storage level, these infrastructures typically utilize POSIX-based parallel file systems that provide strong consistency semantics, which are often unnecessary and even undesirable for many modern HPC applications. In contrast, cloud infrastructures employ object storage systems that bypass POSIX consistency requirements [72]. These systems usually fall into the category of HTC systems and are designed for extreme scalability, making them well-suited for cloud computing and similar commercial environments. The current state-of-the-art centers on research to adapt scientific workflows to the infrastructures of cloud providers [44, 73–75].

5.2 Data Storage Infrastructures

5.2.1 Earth System Grid Federation

The Earth System Grid Federation (ESGF) is a global infrastructure and network consisting of internationally distributed research centers and HPC infrastructures designed to support climate modeling and data dissemination [76, 77]. The ESGF enables modeling groups to upload model output to federation nodes for archiving and community access at any time. To facilitate multi-model analyses, the ESGF ensures standardization of model output the NetCDF format. It also facilitates the collection, archival, and access of model output through the ESGF data replication centers. Figure 5.2 illustrates the worldwide reach of the ESGF.

This design enhances data redundancy, scalability, and accessibility while distributing computational workloads. The infrastructure consists of data nodes that store and serve climate model outputs and index nodes that provide metadata services for efficient dataset

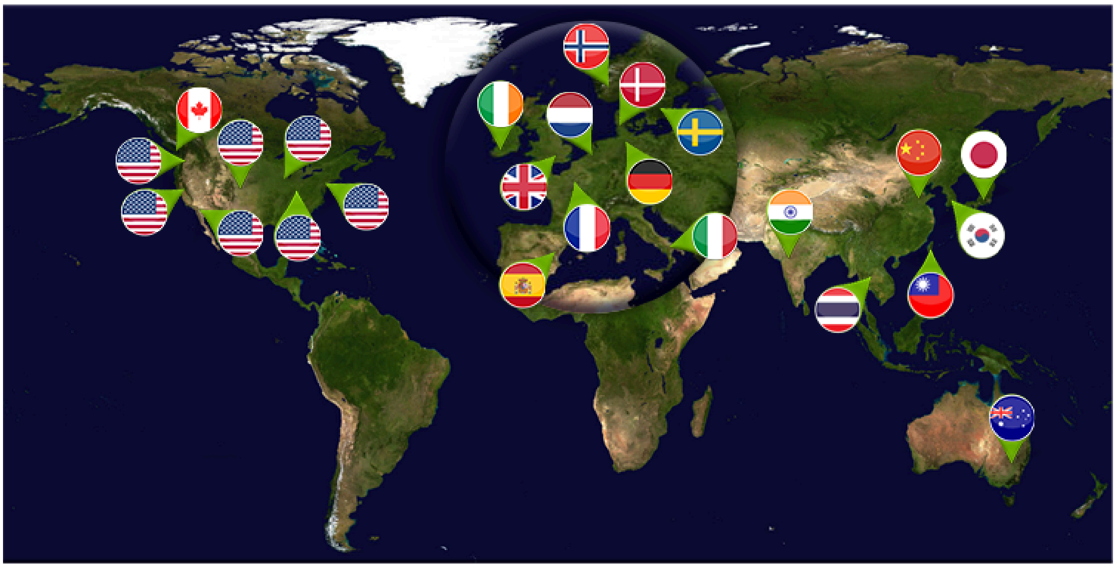


FIGURE 5.2: The ESGF is formed by tier 1 and tier 2 nodes that provide a global and federated storage infrastructure for climate model output. **Source:** ESGF, `https://esgf-node.ipsl.upmc.fr/projects/esgf-ipsl/` (last accessed, 11 June 2025).

discovery. Figure 5.3 illustrates the different types of ESGF nodes. Figure 5.4 illustrates the software architecture and different services existing in the federation. The ESGF employs standardized metadata and controlled vocabularies to ensure interoperability across different datasets, facilitating seamless data integration. It also imposes additional requirements and providing detailed specifications for the management and dissemination of model output [12, 78].

As a result, the ESGF has emerged as the primary distributed data archive for climate data, hosting data for international projects such as CMIP [14] and CORDEX [79]. It catalogs and stores tens of millions of files, with more than 30 petabytes of data, distributed between research institutes around the world [80], and serves as a reference archive for Assessment Reports (AR, Asadnabizadeh 81) on Climate Change produced by the Intergovernmental Panel on Climate Change (IPCC, Venturini et al. 82). The ESGF supports data replication strategies that enhance reliability and prevent data loss while integrating persistent identifiers (PIDs) to track dataset usage and maintain scientific reproducibility [83]. This approach ensures that the data remains verifiable and traceable, reinforcing the integrity of climate research.

The sheer size and complexity of ESGF emerged as a matter of great concern at the end of CMIP5, when the growth in data volume relative to CMIP3 (from 40 TB to 2 PB, a 50-fold increase in 6 years) suggested the community was on an unsustainable path [12]. The ESGF infrastructure is designed as a file distribution system, but scientific research often requires multidimensional data analysis on datasets encompassing multiple variables, spanning the entire time period, multiple model ensembles, and different climate

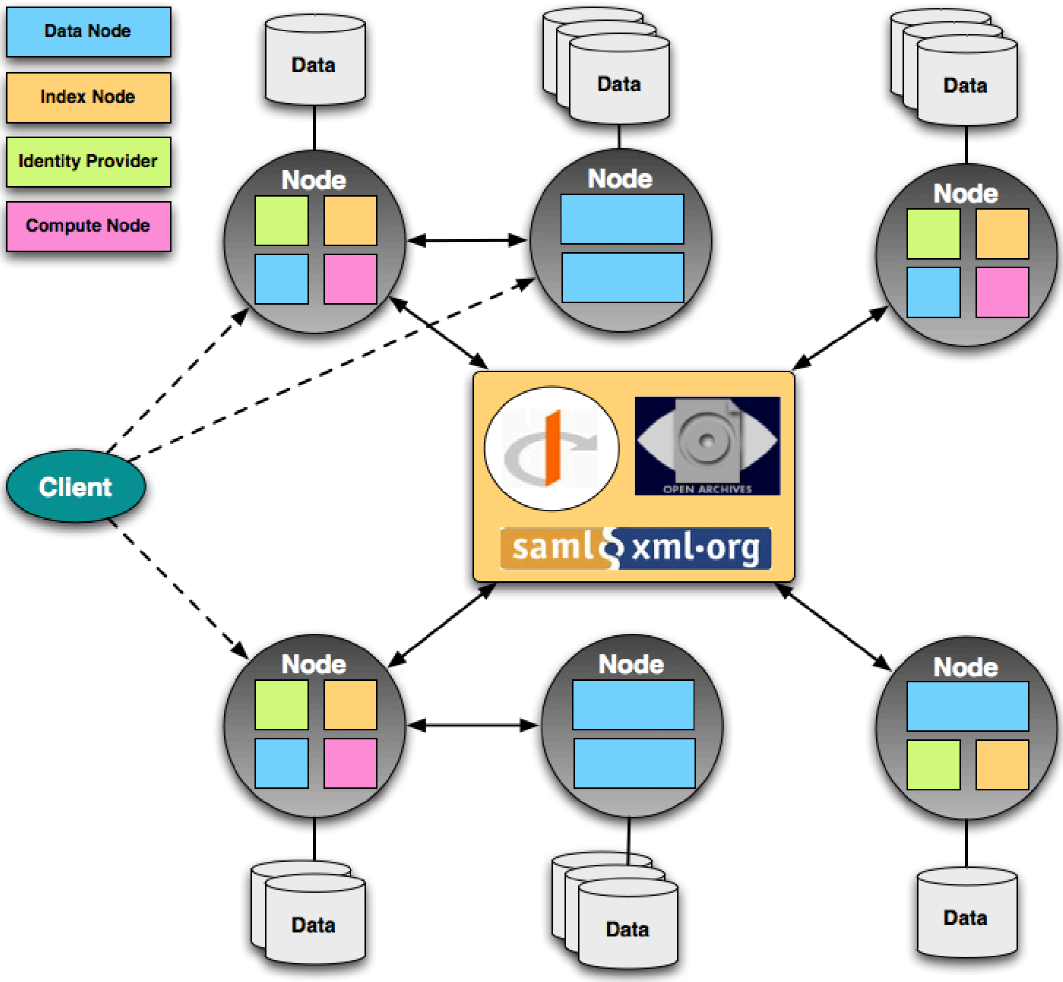


FIGURE 5.3: Illustration of the different software stack that make different ESGF node types. Index nodes are Tier 1 nodes that offer both data node and search, in addition to custom services such as computation and identity management. **Source:** ESGF, `https://esgf.llnl.gov/federation-design.html` (last accessed, 11 June 2025).

model runs. Several ongoing developments in scientific data research try to address the issues of growing data volume and variety and provide new approaches to data analysis [44]. While the ESGF provides a critical platform for data sharing, its current architecture lacks integrated tools for advanced data analysis. Thus, researchers must handle data access and analysis independently. This thesis will address this limitation of the ESGF by proposing a methodology based on remote data access and virtual ARD. The methodology is presented in Chapter 7 and it will be evaluated in terms of performance in Chapter 8.

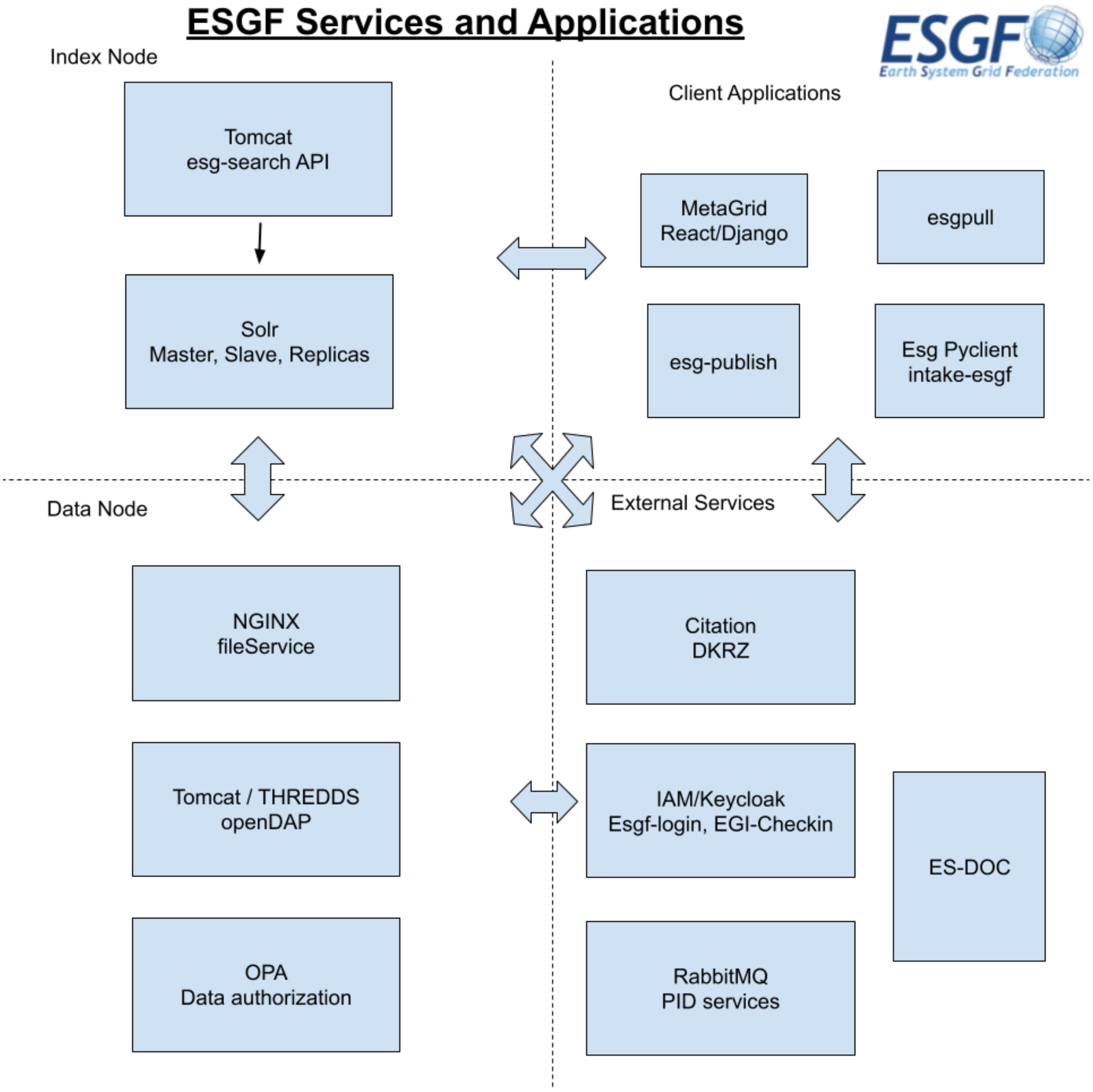


FIGURE 5.4: Software components of the different types of ESGF nodes. **Source:** `https://esgf.llnl.gov/esgf-technical-overview.html` (last accessed, 11 June 2025).

5.2.2 Pangeo

Pangeo is an open-source initiative designed to enable scalable, cloud-native analysis of large scientific datasets, with a particular focus on geoscience and climate modeling. It provides a flexible and modular framework for analyzing vast amounts of data by leveraging cloud computing and HTC-oriented infrastructures, in contrast to the HPC infrastructures of the ESGF. Thus, Pangeo enhances the accessibility and usability of climate model outputs, allowing researchers to efficiently process and analyze data using remote data access 4.2.2 without the constraints of traditional download-based workflows [44].

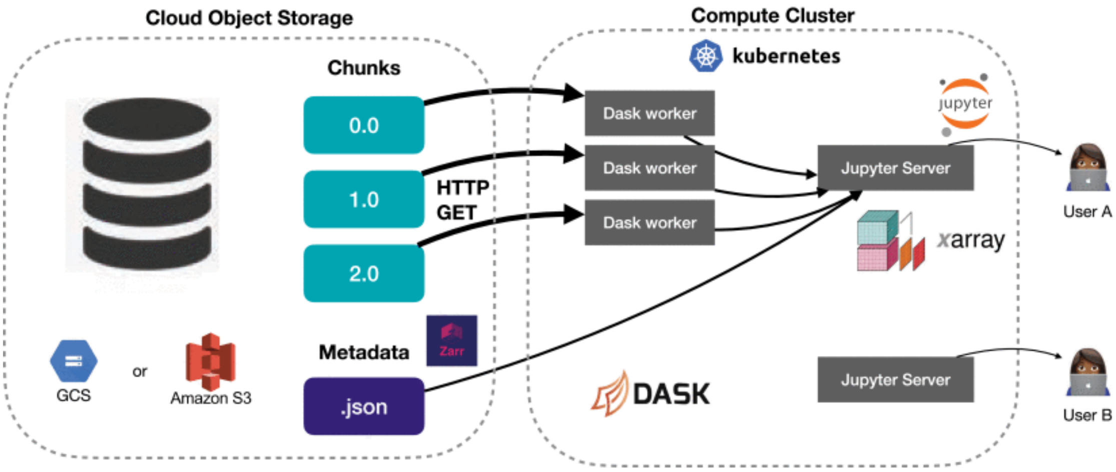


FIGURE 5.5: Pangeo overall architecture diagram. Cloud climate data repositories are hosted in cloud object stores (left), in the Zarr format. Compute nodes (right) fetch data and metadata from the object store. Users connect to the system via Jupyter and run interactive data analysis workflows with Xarray, which optionally makes use of Dask clusters. **Source:** Abernathey et al., 2021.

Pangeo proposes the use of open-source Python-based tools, including Dask for parallel computing, Xarray for multidimensional array processing 4.3.1, and Zarr for cloud-optimized data storage 3.3. These technologies allow users to perform interactive real-time analysis of climate data, reducing the time and computational resources needed for complex model evaluations. Collaborations between ESGF and Pangeo have resulted in the free availability of climate data sets hosted by cloud providers that offer additional functionality to the file downloads provided by ESGF [1]. The availability of cloud resources to users worldwide enables more efficient access to climate datasets, allowing them to leverage cloud-native tools to extract insights at unprecedented scales.

Pangeo Forge has arisen from Pangeo as a community-driven open source platform designed to facilitate the production of analysis-ready, cloud-optimized (ARCO) data [84]. It simplifies the process of extracting, transforming, and cataloging environmental data, making it accessible to a wider range of scientists. The platform addresses the challenges of traditional data workflows, which require extensive technical expertise and computational resources, by providing a structured framework for data preparation. Inspired by Conda Forge, Pangeo Forge adopts a crowdsourced model where contributors create and share *recipes* - automated workflows that convert raw datasets into optimized formats such as Zarr. By democratizing ARCO data production, Pangeo Forge reduces redundancy in data processing and enhances accessibility, fostering a more inclusive research environment.

[1] https://cloud.google.com/blog/

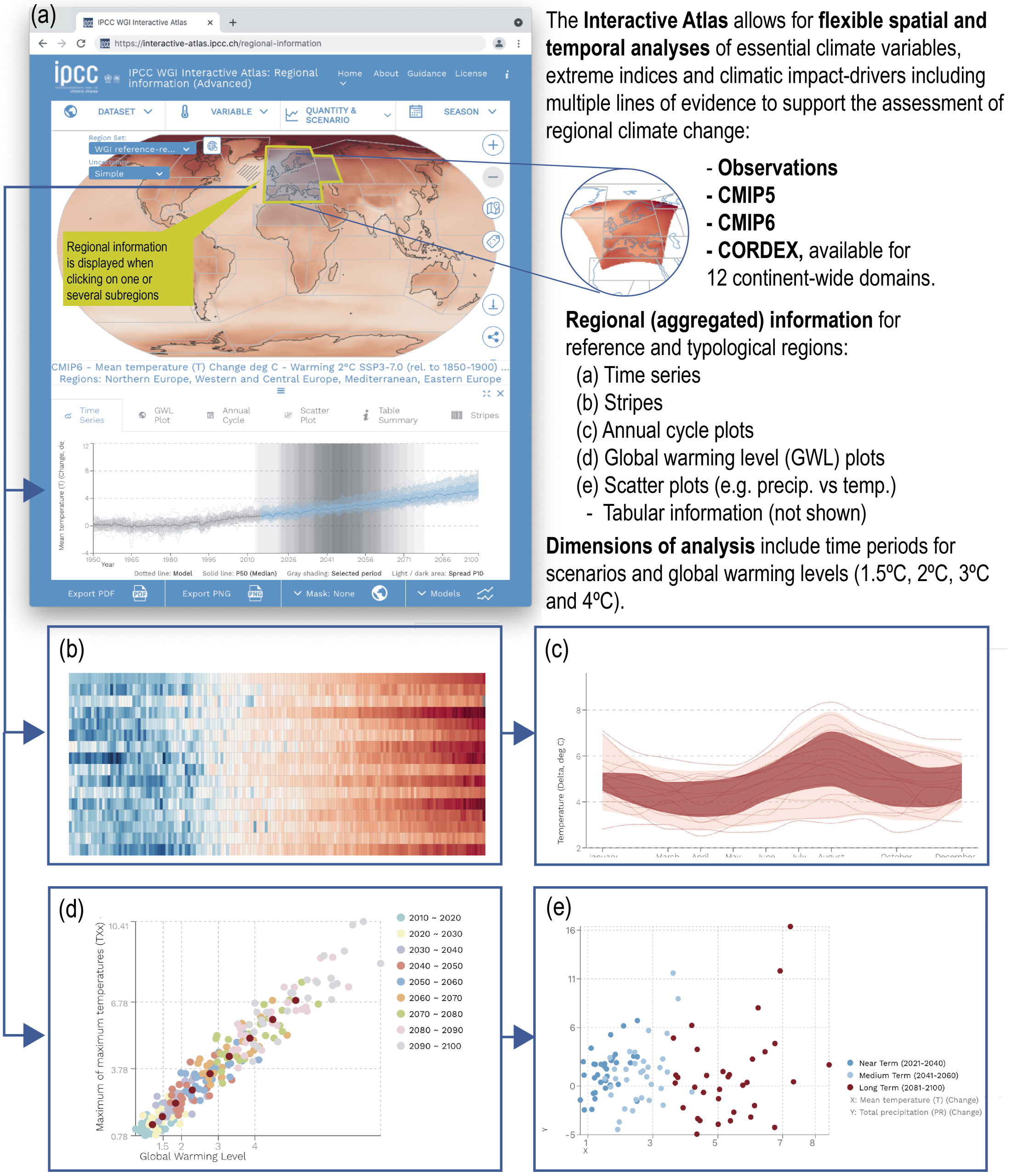
5.3 Climate Information Products

In contrast to climate storage infrastructures, Climate Information Products aim to improve climate-sensitive decision-making by providing climate information that is useful, usable, and used [85]. These tools are knowledge-oriented, rather than data-oriented, offering insights into the climate system in a clear and accessible manner. They are specifically designed for users without a comprehensive background in data science or advanced programming skills. As a result, Climate Information Products are often delivered through easy-to-use graphical interfaces, such as web applications, which typically do not require any programming expertise.

The Interactive Atlas (`https://interactive-atlas.ipcc.ch` was an innovation introduced in AR6 [23] to expand the assessment, allowing flexible spatial and temporal analysis for most of the datasets and Climate Impact Drivers (CIDs) used in the report. Users can explore, interact with, and download global maps, spatially aggregated time series, and other regional products displaying recent trends and future changes across emission scenarios for over twenty CIDs. The Atlas allows comparing different lines of evidence based on different emission/forcing scenario families, such as Representative Concentration Pathways (RCPs) used in CMIP5 [86] and CORDEX [87] and the Shared Socio-Economic Pathways (SSPs) used in CMIP6 [14]. Figure 5.6 illustrates various climate products available in the Interactive Atlas.

The AR6 provides authoritative assessments that serve as key references for aligning regional analyses with global evaluations of climate change. Complementarily, the Interactive Atlas offers an accessible platform to visualize and explore the underlying climate data. Facilitating the accessibility and reusability of this information was a key objective of AR6, guided by the FAIR principles, with an initial focus on the datasets and code underlying the figures in the report and the Atlas [27]. This work was overseen by the IPCC Task Group on Data Support for Climate Change Assessments (TG-Data), the IPCC Technical Support Unit (TSU) of Working Group I (WGI), and the IPCC Data Distribution Centers (IPCC-DDC) [88–91]. The Interactive Atlas served as a comprehensive test case for implementing the FAIR principles, with full publication of its underlying software and datasets.

The code recipes (regridding, index calculation, bias adjustment, etc.) and auxiliary information (common grids, masks, shapefiles for regions, etc.) are available for reproducibility and reusability through the IPCC-WGI/Atlas repository (`https://github.com/IPCC-WG1/Atlas`; [27]). Additionally, the gridded monthly dataset that underpins the Atlas is accessible via the IPCC Data Distribution Centre (IPCC-DDC,


(a)
IPCC WGI Interactive Atlas
https://interactive-atlas.ipcc.ch/regional-information
IPCC WGI Interactive Atlas: Regional information (Advanced)
Home About Guidance License
DATASET
VARIABLE
QUANTITY & SCENARIO
SEASON
Region Set:
WGI reference-re...
Uncertainty
Simple
Regional information is displayed when clicking on one or several subregions
CMIP6 - Mean temperature (T) Change deg C - Warming 2°C SSP3-7.0 (rel. to 1850-1900) ...
Regions: Northern Europe, Western and Central Europe, Mediterranean, Eastern Europe
Time Series
GWL Plot
Annual Cycle
Scatter Plot
Table Summary
Stripes
Mean temperature (T) (Change, de
Year
Dotted line: Model
Solid line: P50 (Median)
Gray shading: Selected period
Light / dark area: Spread P10
Export PDF
Export PNG
Mask: None
Models
(b)
(c)
Temperature (Delta, deg C)
(d)
Maximum of maximum temperatures (TXx)
Global Warming Level
2010 ~ 2020
2020 ~ 2030
2030 ~ 2040
2040 ~ 2050
2050 ~ 2060
2060 ~ 2070
2070 ~ 2080
2080 ~ 2090
2090 ~ 2100
(e)
Near Term (2021-2040)
Medium Term (2041-2060)
Long Term (2081-2100)
X: Mean temperature (T) (Change)
Y: Total precipitation (PR) (Change)

The **Interactive Atlas** allows for **flexible spatial and temporal analyses** of essential climate variables, extreme indices and climatic impact-drivers including multiple lines of evidence to support the assessment of regional climate change:

- **Observations**
- **CMIP5**
- **CMIP6**
- **CORDEX**, available for 12 continent-wide domains.

Regional (aggregated) information for reference and typological regions:

(a) Time series
(b) Stripes
(c) Annual cycle plots
(d) Global warming level (GWL) plots
(e) Scatter plots (e.g. precip. vs temp.)
- Tabular information (not shown)

Dimensions of analysis include time periods for scenarios and global warming levels (1.5ºC, 2ºC, 3ºC and 4ºC).

FIGURE 5.6: Screenshots from the Interactive Atlas showing regional information. (a) The main interface displays a global map with controls for selecting the dataset, variable, reference and baseline periods, and season (here: annual temperature change from CMIP6 at a global warming level of 2 °C under SSP3-7.0, relative to 1850–1900). (b–e) Examples of visualizations presenting regionally averaged information for the chosen reference regions. **Source:** Figure 8 in Gutiérrez et al., 2021.

`https://www.ipcc-data.org/`). These resources support the traditional “download-and-analyze” model for data processing [44]. However, the scale of the datasets presents significant challenges for users and computational infrastructures, leading to an increasing adoption of “next-to-data” computing approaches, including cloud-based solutions (see e.g. [44, 92, 93]).

5.4 Data Laboratories

This thesis introduces the concept of a Data Laboratory, or *DataLab*, to bridge the technological gap between climate data storage infrastructures, such as ESGF and Pangeo, and high-level, end-user–oriented tools, such as the Interactive Atlas. A *Data Laboratory* (DataLab) can be defined as a digital research environment that integrates computational resources, software tools, and access to large climate datasets to enable advanced analysis, experimentation, and reproducibility. Data laboratories, in contrast to Climate Information Products, target advances users, such as researchers, and data scientists who require advanced tools and computational resources to explore and manipulate climate data. This thesis provides its own definition of the concept of data laboratory as an infrastructure equipped with hardware and software tools that support, to varying degrees, the tasks of climate data storage and analysis. As an additional requirement for climate data laboratories, this work considers the extent to which a data laboratory enables others in the community to reproduce climate results.

Data laboratories could be designed as B2B (business-to-business) services, providing a specialized environment for scientific research, where users can access large datasets, experiment with different methodologies, and replicate or build upon existing research findings. Unlike Climate Information Products, which aim to present simplified and accessible insights for non-research purposes, data laboratories emphasize flexibility and depth, offering advanced functionalities for data analysis, model simulations, and complex visualizations.

In a traditional experimental laboratory, researchers typically work with physical materials, conduct experiments, and observe the results to test hypotheses or explore new phenomena. The process involves a great deal of hands-on work, such as setting up experiments, manipulating variables, measuring results, and drawing conclusions based on empirical data. These laboratories are designed to provide precise control over the conditions of the experiment and to ensure that results can be reliably reproduced and validated. The equipment in a traditional laboratory, such as microscopes, test tubes, and centrifuges, allows scientists to work directly with the physical world and manipulate it to gain deeper insights into natural processes.

In contrast, a data laboratory in the context of data science operates primarily in the digital realm, where the "materials" being manipulated are data - often large, complex, and high-dimensional datasets. Rather than physical tools, a data laboratory provides software, algorithms, computational power, and data infrastructures that enable researchers to process, analyze, and model data. Just as a traditional lab provides controlled conditions for physical experiments, a data laboratory offers an environment where the conditions for data analysis, including computational resources and access to various datasets, are carefully managed. Researchers in this context rely on advanced programming, statistical techniques, and data visualization to experiment with data and test hypotheses, which may involve simulating climate models, analyzing trends in climate datasets, or creating predictive algorithms.

There is an increasing demand for data laboratories in the realm of climate data science, as the complexity and scale of climate-related data continue to expand. Traditional methods of analyzing climate data, while valuable, are no longer sufficient to handle the vast datasets that are essential for understanding the climate system. Several approaches have already arisen, such as Climate Analytics-as-a-Service (CAaaS, [93]), Collaboratories [94] and Digital Spaces [95]. The data consolidation process involved in building these new systems may lead to the duplication of vast volumes of data, resulting in significant operational and storage costs. However, the cost of data duplication is expected to be offset by the efficiency gained in information synthesis.

Moreover, data laboratories hold significant potential to enhance reproducibility and adherence to FAIR principles. By providing a controlled and interactive environment where datasets, analytical workflows, and computational resources are clearly documented and consistently managed, DataLabs allow researchers to replicate analyses and build upon each other's work more reliably. Such environments can facilitate the generation of structured, transparent, and traceable outputs, including the production of climate assessment reports, like those of the IPCC, where multiple researchers must collaboratively analyze large datasets and present consistent findings.

The interactive features of DataLabs — such as dynamic visualizations, configurable workflows, and immediate feedback on model outputs — further support reproducibility by enabling users to explore alternative scenarios, test assumptions, and validate results in real time. The remainder of this thesis is dedicated to the assessment and evaluation of climate data laboratories and the advanced instruments available to them, such as virtual aggregations, which aim to enhance climate data storage, analysis, and infrastructure. Chapters 6 and 7 demonstrate that, despite recent advances in the provision of data laboratories, further improvements are still needed to transform current data infrastructures and systems into fully effective data laboratories.

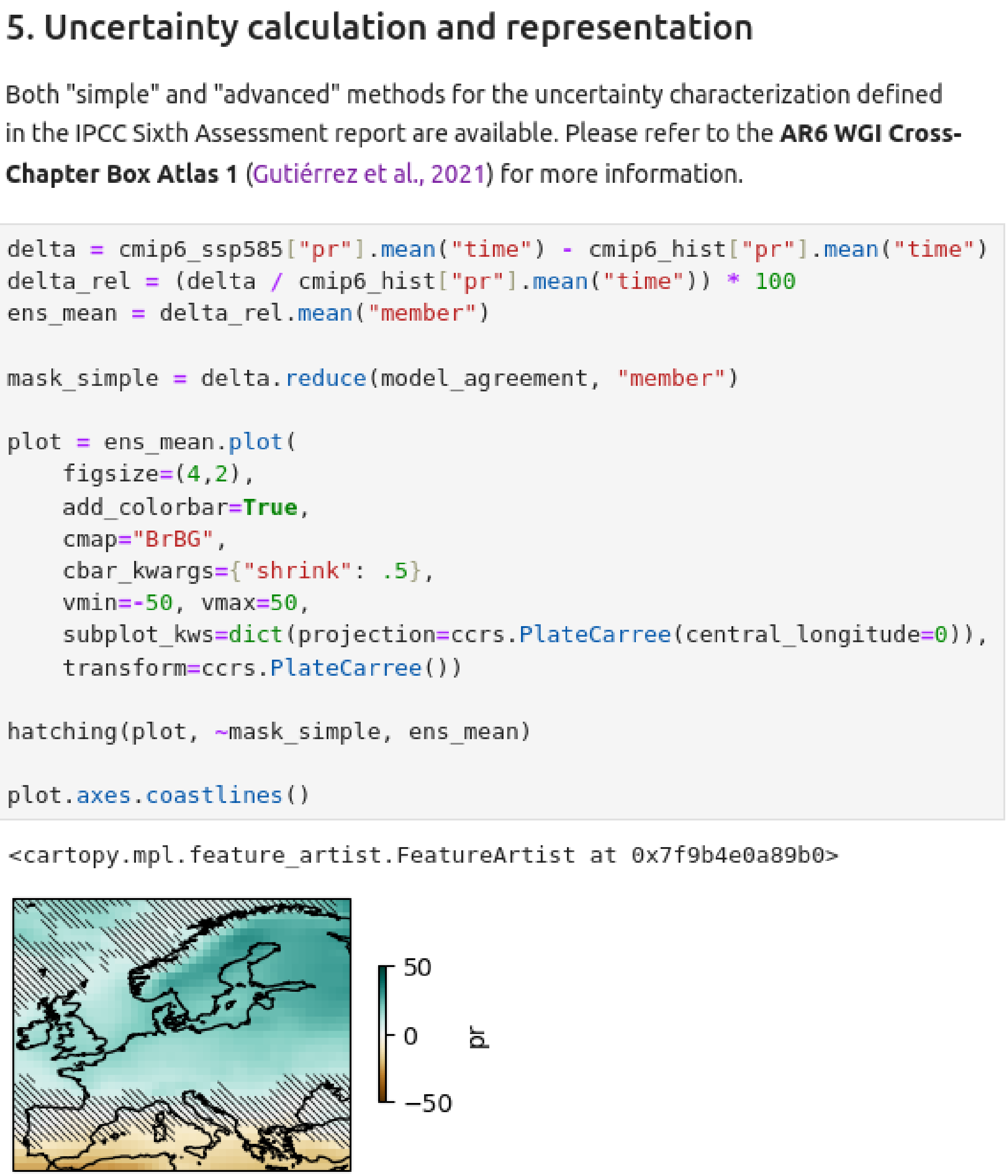
5.4.1 The Jupyter Project

The Jupyter Project [96] was born out of the desire to create an open-source, interactive computing environment that would be dedicated to the growing needs of the data science community. Originally inspired by the IPython project [97], Jupyter aimed to support multiple languages and provide a platform for interactive and reproducible scientific computing. Jupyter Notebooks, the primary product of the initiative, enabled users to mix code execution with rich text, visualizations, and multimedia. As the project expanded, JupyterLab was developed as the next-generation user interface, offering an enhanced, web-based environment that extends the capabilities of Jupyter Notebooks and addresses the growing demands of data science, machine learning, and scientific computing.

JupyterLab has massively improved the field of climate data science by moving away from traditional HPC (High-Performance Computing) command-line infrastructures to a flexible and interactive web-based environment. Before the advent of JupyterLab, working with HPC systems often involved submitting jobs via command-line interfaces, scripting complex batch jobs, and waiting for job completion. This process was not only cumbersome but also required a deep knowledge of job scheduling, cluster management, and system administration. The lack of real-time feedback and the steep learning curve associated with such setups meant that researchers and data scientists were often disconnected from their computations, relying on logs and job statuses to monitor progress.

With JupyterLab, this process was streamlined and brought into a more accessible and interactive environment. One of the key benefits of JupyterLab is its ability to provide real-time interactivity. Unlike traditional command-line interfaces, JupyterLab allows users to run code directly in a notebook format, which integrates seamlessly with visualizations and outputs. As users interact with their data, they receive immediate feedback, making it easier to iterate and refine climate data analyses. Figure 5.7 illustrates a Jupyter Notebook, which is executed in a JupyterLab environment, that executes and displays a climate result in real-time.

The shift to a web-based interface also fosters collaboration in ways that traditional HPC setups couldn't. Data scientists can now easily share notebooks with colleagues or collaborators in real time, allowing for smoother and more effective communication. Tools like Git integration and version control further facilitate collaboration, enabling teams to track changes and collaborate on complex climate data analyses. Services like Binder [98] make it easier to share notebooks and datasets without worrying about dependencies, as they allow others to launch an interactive session of the notebook



5. Uncertainty calculation and representation

Both "simple" and "advanced" methods for the uncertainty characterization defined in the IPCC Sixth Assessment report are available. Please refer to the **AR6 WGI Cross-Chapter Box Atlas 1** (Gutiérrez et al., 2021) for more information.

```
delta = cmip6_ssp585["pr"].mean("time") - cmip6_hist["pr"].mean("time")
delta_rel = (delta / cmip6_hist["pr"].mean("time")) * 100
ens_mean = delta_rel.mean("member")

mask_simple = delta.reduce(model_agreement, "member")

plot = ens_mean.plot(
    figsize=(4,2),
    add_colorbar=True,
    cmap="BrBG",
    cbar_kwargs={"shrink": .5},
    vmin=-50, vmax=50,
    subplot_kws=dict(projection=ccrs.PlateCarree(central_longitude=0)),
    transform=ccrs.PlateCarree())

hatching(plot, ~mask_simple, ens_mean)

plot.axes.coastlines()
```

```
<cartopy.mpl.feature_artist.FeatureArtist at 0x7f9b4e0a89b0>
```

FIGURE 5.7: A snapshot of a climate data laboratory illustrating how descriptive text and source code can be integrated. Additionally, climate figures can be displayed interactively. Data laboratories may also offer pre-installed software dependencies, significantly enhancing the user experience.

directly from a web browser. Moreover, *DataLab as a Service* platforms [99] researchers no longer need to manually provision infrastructure or worry about scaling resources for different computational tasks.

Part II

Main results

Chapter 6

The IPCC AR6 Interactive Atlas DataLab

6.1 Introduction

The Sixth Assessment Report (AR6) of the Intergovernmental Panel on Climate Change (IPCC) highlights that climate change is impacting all regions worldwide and will increasingly do so in the coming decades [26, 100]. Each region experiences diverse changes in the mean and extreme climate conditions, which drive manyfold socio-economic impacts [101]. To assess these changes, AR6 introduced Climatic Impact-Drivers (CIDs), encompassing key hazards characterizing climate risks such as heat and droughts [102] characterized by Essential Climate Variables (ECVs) and (extreme) indices derived from climate data. The AR6 report assessed regional past trends and future changes across emission scenarios and utilized Global Warming Levels (GWLs) as a new policy-relevant dimension to convey information about what the future will look like, depending on the mitigation efforts we make today.

The AR6 and the Interactive Atlas are invaluable sources of information for national adaptation plans and regional climate change studies, serving as authoritative references for aligning regional analyses with global assessments. Facilitating the accessibility and reusability of this information was a key objective of AR6, guided by the FAIR principles and initially focusing on the final datasets and the code underlying the figures in the report. This work was supervised by the IPCC Task Group on Data (TG-Data), the IPCC Technical Support Unit (TSU) of the WGI, and the IPCC Data Distribution Centers (IPCC-DDC) [88–91].

The implementation of FAIR principles was particularly critical in developing the Interactive Atlas tool as part of the WGI report, as it facilitates traceability and reproducibility of results. The Atlas served as a comprehensive test case for the development and implementation of FAIR principles, and the associated code recipes (e.g., regridding, index calculation, bias adjustment) and auxiliary information (e.g., common grids, masks, shapefiles for regions) have been made available for review and reuse through the IPCC-WGI/Atlas repository (`https://github.com/IPCC-WG1/Atlas`; [27]).

Here, additional work is described that extends this effort to the remits of the Interactive Atlas, enabling reproducibility and reusability of the underlying code and data. First, the gridded monthly dataset that underpins the Interactive Atlas is presented. This comprehensive dataset includes over twenty CIDs from multiple global and regional projection data sources (CMIP5, CMIP6, and CORDEX), all harmonized to a common grid, units, calendar, and format. With this dataset, users can develop customized products not directly available from the IPCC Interactive Atlas, such as regional information at national or subnational scales. The dataset is accessible for download from the IPCC Data Distribution Centre (IPCC-DDC, `https://www.ipcc-data.org/`), through its long-term archival site at DIGITAL.CSIC (`https://digital.csic.es/`), as well as from a copy stored at the Copernicus Data Store [103]. Through these access points, users are directed toward the traditional “download-analyze” processing model [44]. While this approach may appeal to those wishing to store a copy of the dataset on their own servers, it becomes unsustainable for many due to the large volume of data. Consequently, “next-to-data“ computing services, including cloud-based solutions, are gaining widespread adoption in this context (see e.g. [44, 92, 93]).

The second component of the work described in this work addresses this challenge by leveraging recent advancements in data storage and analysis, culminating in the Interactive Atlas DataLab [104]. This notebook-oriented online platform facilitates next-to-data analysis and visualization by streamlining the data processing pipeline, providing transparent data access, and enabling advanced climate data processing through the integration of specialized tools from R (based on Climate4R [61]) and Python (based on xarray [60]). In doing so, it highlights the importance of user-friendly platforms for the reproducibility and reusability of climate data and products.

6.2 The IPCC WGI Atlas Dataset

The *IPCC-WGI AR6 Interactive Atlas Gridded Monthly Dataset* provides global and regional climate change projections for 22 Climate Impact Drivers (CIDs) featured in the IPCC Interactive Atlas (see Table 6.1). Bias-adjusted results (TX35ba, TX40ba) are included for the two threshold-dependent indices (TX35, TX40) using the ISIMIP3 bias adjustment method [105], enabling assessment of model bias impacts on these indices [106]. This dataset offers gridded information at monthly (and in some instances, annual) temporal resolution, derived from historical and future emission scenarios (Representative Concentration Pathways and Shared Socioeconomic Pathways) in CMIP5 [86], CMIP6 [14], and CORDEX [87]. An overview of the indices available for each dataset is provided in Table 6.2. Note that certain indices are excluded for specific CORDEX domains where they are not relevant, such as days with maximum near-surface temperature above 35 °C (TX35) in the CORDEX-ANT domain (Antarctica).

This dataset was derived from a volume of 200 TB of data accessed through the ESGF [77]. The ensembles were harmonized using common regular grids with horizontal resolutions of 2° (CMIP5), 1° (CMIP6), 0.5° (CORDEX), and 0.25° (European CORDEX domain) resulting in a collection of 863 NetCDF files totaling 500 GB, with NetCDF compression enabled. They include comprehensive metadata that adhere to CF conventions, ensuring correct data interpretation by applications and users. The corresponding data source inventory, reference grids, and code are available in the IPCC-WGI/Atlas repository (`https://github.com/IPCC-WG1/Atlas`; [27]).

6.3 The IPCC WGI Atlas DataLab

The traditional “download-analysis“ processing paradigm is increasingly unsustainable due to the growing heterogeneity and volume of data [44]. Climate researchers are quickly adopting alternative seamless approaches for data analysis that support next-to-data processing. In this thesis, the term DataLab is used to refer to such systems [99]. Despite the differences in their architectures, available resources, and scope, all these systems share the common goal of providing more effective ways to use computer systems for information sharing and processing.

In response to this growing need, the Atlas DataLab presented here was developed in the framework of IPCC-DDC CSIC activities to provide a notebook-based reproducibility and reusability platform, integrating the software and the IPCC-WGI AR6 Interactive Atlas Dataset (see Sec. 6.2). Figure 6.1 shows a schematic illustration of the architecture highlighting the different computing infrastructures supporting the DataLab. The foundation of the Atlas DataLab is a GitHub repository (see Sec. 6.3.1) including the software and data access components together with illustrative notebooks to reuse the dataset and reproduce the products of the Interactive Atlas. The GitHub repository contains two *launchers* connecting with interactive computing environments for the execution of the notebooks building on JupyterHub: BinderHub and the IPCC-DDC

Type	Name	Description
heat & cold	t	Monthly mean of daily mean near-surface (usually, 2 meters) air temperature
	tn	Monthly mean of daily minimum near-surface (usually, 2 meters) air temperature
	tnn	Monthly minimum of daily minimum near-surface (usually, 2 meters) air temperature
	fd	Monthly count of days with minimum near-surface (usually, 2 meters) temperature below 0 degC
	hd	Cumulative degree days by which the daily average temperature falls below 15.5 degC (see IPCC AR6 WGI Annex VI)
	tx	Monthly mean of daily maximum near-surface (usually, 2 meters) air temperature
	txx	Monthly maximum of daily maximum near-surface (usually, 2 meters) air temperature
	tx35	Monthly count of days with maximum near-surface (usually, 2 meters) temperature above 35 degC
	tx35ba	Bias adjusted (ISIMIP3 trend preserving method) monthly count of days with maximum near-surface (usually, 2 meters) temperature above 35 degC
	tx40	Monthly count of days with maximum near-surface (usually, 2 meters) temperature above 40 degC
	tx40ba	Bias adjusted (ISIMIP3 trend preserving method) monthly count of days with maximum near-surface (usually, 2 meters) temperature above 40 degC
	cd	Annual energy consumption to cool the excess of temperature above 22 degC (see IPCC AR6 WGI Annex VI)
wet & dry	pr	Monthly mean of daily accumulated precipitation of liquid water equivalent from all phases
	rx1day	Monthly maximum of 1-day accumulated precipitation of liquid water equivalent from all phases
	rx5day	Monthly maximum of 5-day accumulated precipitation of liquid water equivalent from all phases
	cdd	Annual maximum of consecutive days when daily accumulated precipitation amount is below 1 mm
	spi6	Monthly Index that compares accumulated precipitation for 6 months with the long term precipitation distribution for the same location and cumulation period (see IPCC AR6 WGI Annex VI)
snow & ice	prsn	Monthly mean of daily accumulated liquid water equivalent thickness snowfall
	siconc	Percentage of sea grid cell area covered by ice
wind	sfcwind	Monthly mean of daily mean near-surface (usually, 10 meters) wind speed
ocean	sst	Monthly mean of temperature of upper boundary of the liquid ocean, including temperatures below sea-ice and floating ice shelves
	ph	Monthly mean of negative log of hydrogen ion concentration with the concentration expressed as mol H kg^{-1}

Table 6.1: Table of variables/indexes available in the Interactive Atlas Dataset, grouped by types of variables/indices. See Table 6.2 for the availability of the variables across different dataset sources.

CID	CMIP5	CMIP6	AFR	ANT	ARC	AUS	CAM	EAS	EUR	NAM	SAM	SEA	WAS	Freq.
t	*	*	*	*	*	*	*	*	*	*	*	*	*	mon
tn	*	*	*	*	*	*	*	*	*	*	*	*	*	mon
tnn	*	*	*	*	*	*	*	*	*	*	*	*	*	mon
fd	*	*	*		*	*	*	*	*	*	*		*	mon
hd	*	*	*		*	*	*	*	*	*	*	*	*	yr
tx	*	*	*	*	*	*	*	*	*	*	*	*	*	mon
txx	*	*	*	*	*	*	*	*	*	*	*	*	*	mon
tx35	*	*	*			*	*	*	*	*	*	*	*	mon
tx35ba	*	*	*			*	*	*	*	*	*	*	*	mon
tx40	*	*	*			*	*	*	*	*	*	*	*	mon
tx40ba	*	*	*			*	*	*	*	*	*	*	*	mon
cd	*	*	*			*	*	*	*	*	*	*	*	yr
pr	*	*	*	*	*	*	*	*	*	*	*	*	*	mon
rx1day	*	*	*	*	*	*	*	*	*	*	*	*	*	mon
rx5day	*	*	*	*	*	*	*	*	*	*	*	*	*	mon
cdd	*	*	*	*	*	*	*	*	*	*	*	*	*	yr
spi6		*	*			*	*	*	*	*	*	*	*	mon
prsn		*												mon
siconc		*												mon
sfcwind		*	*	*	*	*	*	*	*	*	*	*	*	mon
sst		*												mon
ph		*												mon

Table 6.2: Summary of data availability for the IPCC-WGI AR6 Interactive Atlas Dataset. Asterisks denote dataset source availability, with variables and indices grouped as in Table 6.1. The columns correspond to CMIP5, CMIP6, and the various CORDEX domains—AFR: Africa, ANT: Antarctica, ARC: Arctic, AUS: Australasia, CAM: Central America, EAS: East Asia, EUR: Europe, NAM: North America, SAM: South America, SEA: Southeast Asia, and WAS: South Asia—as well as the temporal frequency. CMIP5 and CMIP6 data are available at 2° and 1° resolution, respectively, and CORDEX data is provided at 0.5° resolution, except for CORDEX-EUR, which is available at a finer 0.25° resolution.

CSIC Hub (see Section 6.3.2) provide two main computational environments for running the notebooks. The environment offered by BinderHub is freely available but subject to hardware and resource limitations [98]. In contrast, the IPCC-DDC CSIC Hub offers a scalable infrastructure with greater CPU, memory, and bandwidth performance, built on cloud and HPC resources hosted at the CSIC premises—specifically the Instituto de Física de Cantabria (IFCA)—to support IPCC-DDC activities. Currently, access to the CSIC Hub is limited to authorized users under IPCC initiatives, although efforts are underway to publish it as a Software as a Service (SaaS) product in the European Open Science Cloud (EOSC) marketplace [74], enabling broader open use by relying on EOSC resources.

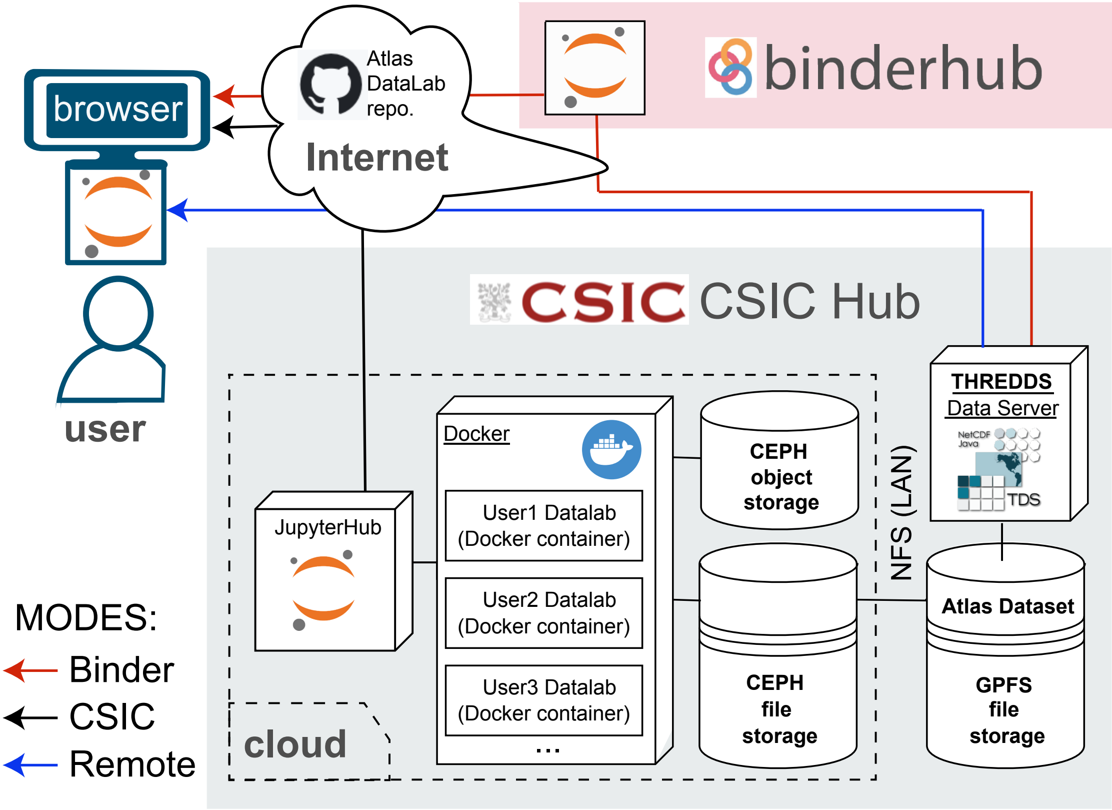


FIGURE 6.1: Arquitecture of the DataLab and its deployment on the premises of IFCA (the CSIC-DDC Hub). Users with access to the JupyterHub infrastructure (green arrow) enjoy the benefits of local data access. External users to the CSIC-DDC Hub may still reproduce the examples of the DataLab by using remote data access provided by the THREDDS Data Server, at the expense of lower bandwidth (blue arrows). Further details on the performance of each data access type if provided in section 6.3.4. **Source:**Cimadevilla et al., 2025.

The Atlas DataLab was developed with particular attention to the advantages and limitations of various approaches commonly employed in the climate community for implementing the data access layer. On the one hand, traditional climate data services (e.g. those based on the OPeNDAP protocol [57]) rely on remote data access allowing federated services. On the other hand, modern cloud-based services offer integration of computing and storage resources within cloud infrastructures. The former has the advantage of not requiring duplication and reformatting of the information, so it is cost-effective in cases where the information is originally available in standard data repositories operated by the climate community, such as the ESGF [77], where information is updated and versioned (bug fixing) frequently. The latter benefits from quick developments of data-intensive technologies and are more aligned with ongoing technological developments. This work compares these two alternatives showcasing their performance when reproducing key products from the Interactive Atlas in Section 6.3.4.

6.3.1 The DataLab GitHub Repository

As previously noted, the Atlas DataLab uses a dedicated GitHub repository (`https://github.com/SantanderMetGroup/IPCC-Atlas-Datalab`; [107]) to manage its software components, including illustrative notebooks with seamless data access to the Interactive Atlas Dataset (see Figure 6.1).

GitHub provides a collaborative environment for code development with integrated version control and issue tracking [108] which has been previously used in IPCC activities, in particular for the reproducibility and reusability of some of the AR6 products, including the Interactive Atlas via the IPCC-WGI/Atlas repository [27]. Researchers with a GitHub account can contribute to the repository of the DataLab by suggesting improvements, reporting issues, or submitting pull requests, for example, to modify or add notebooks or additional software components. The repository is self-documented using Markdown README files.

The Atlas DataLab repository includes configuration files used to set up a research environment conforming to the Reproducible Execution Environment Specifications (REES). Software dependencies for data access and analysis using Conda (`https://docs.conda.io`), while system dependencies are handled with Docker [109]. Users can easily build this environment using the *launch* buttons on the main README page. Once initiated, the repository content is automatically cloned, granting users access to ready-to-run resources through interactive online services such as BinderHub [98], where the data is accessed remotely (red line in Figure 6.1), or the CSIC Hub, which supports next-to-data computing (black line in Figure 6.1). Alternatively, users can run Docker images on their local workstations, allowing them to reproduce the environment and run notebooks locally, thanks to remote data access (blue line in Figure 6.1). The CSIC Hub infrastructure is explained in more detail in Section 6.3.2.

The Atlas DataLab repository extends the functionalities of the IPCC-WGI/Atlas repository. While the IPCC-WGI/Atlas repository includes code recipes (e.g., regridding, index calculation, bias adjustment) and auxiliary resources (such as common grids, masks, and shapefiles for regions) used to prepare Interactive Atlas products, the DataLab repository offers direct access to the Interactive Atlas dataset (see Sec. 6.3.2). Additionally, it includes notebooks that demonstrate how to access and load the data within the working environment, as well as perform subsequent step-by-step operations to reproduce, reuse, and customize key Interactive Atlas products, thereby enhancing their reproducibility and reusability. These products include global change maps (for future periods across different scenarios or Global Warming Levels) and regionally aggregated outputs, such as time series and climate stripes. Two typical use case examples are presented in this work (Section 6.4).

End-to-end data processes, from data access and loading to the generation and visualization of final results, are performed using specialized Python and R tools within separate Conda environments. In Python, the primary tool is `xarray` [60], while in R, the `Climate4R` [61] framework is utilized. This approach ensures a streamlined workflow tailored to each programming language's strengths.

6.3.2 The DataLab CSIC Hub Infrastructure

The CSIC Hub supports IPCC-DDC activities with scalable cloud and HPC infrastructures hosted in CSIC premises at the Instituto de Física de Cantabria (IFCA). The next-to-data computing component of the DataLab (Figure 6.1) has been deployed in the cloud infrastructure [99], providing access to cloud computing machines with sufficient capacity to handle workloads that require extensive CPU and memory resources, which is essential for some specialized data analysis tasks required to reproduce some Interactive Atlas products.

The data access component of the Atlas DataLab is deployed in the HPC infrastructure (Figure 6.1), with the NetCDF files of the Atlas Dataset (see Sec. 6.2) in a GPFS file system. A THREDDS Data Server (TDS) [58] facilitates ex-situ remote access to the files of the Atlas Dataset via OPeNDAP [57]. This access method is essential for retrieving data from BinderHub or from environments installed on local workstations. In the case of next-to-data computing within the CSIC Hub, in-situ access is also possible by pointing directly to the path of the stored NetCDF files. In this case, the volume of the GPFS file system, including the Interactive Atlas dataset, is shared via NFS (Network File System) within the OpenStack cloud environment, so files are available via a local area network where JupyterHub is hosted. This hybrid in-situ/ex-situ data access setup allows seamless access to the Atlas Dataset from different infrastructures and allows testing the performance of the alternative approaches for the calculation of illustrative products from the IPCC Interactive Atlas (see Section 6.3.4).

Both in-situ (NFS) and ex-situ (OPeNDAP) data endpoints are cataloged in a CSV inventory file (*data_inventory.csv*), which enables convenient querying across various programming environments, such as Python (via Pandas) and R (through native CSV processing). This catalog allows users to locate dataset endpoints based on several facets: type (indicating whether the data is accessed via OPeNDAP or NFS); variable (the climate variable or index, as detailed in Table 6.1); project (the dataset source, including CMIP5, CMIP6, or the different CORDEX domains, as shown in Table 6.2); experiment (including historical, RCP26, SSP126, etc.); and frequency (with values "mon" for monthly and "yr" for yearly, as shown in Table 6.2). All Jupyter Notebooks feature step-by-step examples showing how to search for relevant data in the CSV inventory, as well as how to load and work with it efficiently.

6.3.3 CSIC-DDC Hub for Climate Data Analysis

A version of the DataLab has been deployed in the cloud infrastructure of the CSIC IPCC-DDC node [99], providing access to cloud computing machines with sufficient capacity to handle workloads that require extensive CPU and memory resources, which is essential for some specialized data analysis tasks. In this case, a volume of the GPFS file system is shared via NFS within the OpenStack cloud environment, so files are available via a local area network where JupyterHub and the OPeNDAP server are hosted. This hybrid setup allows seamless access to the IA-Dataset from both infrastructures, either through the graphical interactive interface or via command-line jobs managed by a queue system. Figure 6.1 shows the architecture of the system. Access to this Hub is limited to authorized users in the framework of IPCC activities, although there is work in progress to publish it as a Software as a Service (SaaS) product in the marketplace of the European Open Science Cloud (EOSC), so it can be used openly relying on EOSC resources [74].

The data access component of the Atlas DataLab is provided by the Climate Data Service (CDS) deployed in the HPC infrastructure of the CSIC node of the IPCC Data Distribution Center (IPCC-DDC), based on based on a THREDDS Data Server [58] (TDS) facilitating remote access to the NetCDF files stored on a GPFS file system via OPeNDAP [57]. Both in situ and OPeNDAP data endpoints are cataloged in a CSV inventory file (*data_inventory.csv*), which enables convenient querying across various programming environments, such as Python (via Pandas) and R (through native CSV processing). The use of the CSV format provides a pragmatic balance between interoperability, simplicity, and sustainability for maintaining and sharing the inventory across diverse systems. Although more standardized or domain-specific formats (e.g. JSON, or XML) are sometimes used, they often require additional tooling and increase complexity for users. In contrast, CSV files can be read and processed by almost any data analysis tool without the need for specialized software. This catalog allows users to locate dataset endpoints based on several facets: type (indicating whether the data is accessed via OPeNDAP or a local NetCDF file); variable (the climate variable or index, as detailed in Table 1); project (the dataset source, including CMIP5, CMIP6, or the different CORDEX domains, as shown in Table 6.2); experiment (the mitigation scenario, including historical, RCP26, SSP126, RCP45, etc.); and frequency (with values "mon" for monthly and "yr" for yearly, as shown in Table 6.2).

Once the appropriate endpoint is identified, data access, loading, and processing are performed using specialized tools. In Python, the primary tool is `xarray` [60], while in R, the `Climate4R` [61] framework is used. All Jupyter Notebooks feature step-by-step examples showing how to search for relevant data in the CSV inventory, as well as how to load and work with it efficiently. This work presents two typical use case examples (Section 6.4). Additionally, performance analysis results comparing remote and in situ data access are provided. Details of the performance analysis are presented in Section 6.3.4.

6.3.4 Performance Analysis: Next-to-Data vs Remote Access

Traditional climate data services rely on client/server technologies such as OPeNDAP, which facilitates remote access to climate data subsets using a flexible protocol and a well-defined transmission format. Moreover, several software frameworks, such as Climate4R and xarray, are OPeNDAP-enabled, allowing transparent data processing with no file downloads. This approach ensures resource-efficient access to the specific data subsets required for particular tasks. The Atlas DataLab implements this strategy, as does the ESGF, which manages and distributes the massive volumes of data from global CMIP and regional CORDEX climate change projection initiatives. Additionally, a next-to-data version of the Atlas DataLab is deployed in the CSIC Hub, with local data access.

This section compares the performance of the different data access alternatives using an illustrative product of the Interactive Atlas (time-series of global warming signals for different scenarios, see Figure 6.3a) and focusing on three primary metrics: latency, throughput and amount of data transferred over the network. Two usage scenarios were tested:

- **Next-to-Data:** The Atlas DataLab instance deployed on the CSIC Hub, leveraging a cloud infrastructure (black line in Figure 6.1).
- **Remote Data Access:** An Atlas DataLab instance running on a user's local workstation with remote data access via a broadband connection, which may introduce latency-related overhead (blue line in Figure 6.1).

The remote data access use cases were tested with and without HTTP compression, using a different number of processes to account for parallelism. HTTP compression allows the HTTP server to deflate the information it sends over the network. Thus, the latency of compressing the data is compensated by smaller data transfers, particularly if the compression ratio is high enough. This effect is further enhanced when dealing with slow Internet connections. Note that although NetCDF already uses chunking for data compression, OPeNDAP decompresses the data before sending it through the network.

Figure 6.2 summarizes the computing times from five executions of the global warming time-series notebook (see Sec. 6.4.1). These results demonstrate the repository's capability to handle extensive data processing with consistent performance metrics. The rows show the results for the two usage scenarios: remote data access from an external network (first row) and CSIC Hub with intranet access to data services (second row) . Access to data via OPeNDAP to the THREDDS Data Server is tested in both cases (first two columns, with and without compression, respectively), whereas in-situ access via NFS is only available from the CSIC Hub (third column). As expected, direct in-situ access within the CSIC Hub demonstrated the best performance (smallest runtimes) since no intermediate servers were involved. Remote data access via the THREDDS server with HTTP compression effectively reduced network data transfer by nearly 50% (see the inset numbers), albeit at the cost of throughput due to the additional CPU load for server-side compression. Parallel requests improved overall performance, allowing the server to leverage multiple cores for performing network traffic compression.

This analysis highlights the importance of physical proximity to the THREDDS server. HTTP compression proves advantageous when multiple CPUs are available to perform compression, reducing latency and nearly halving the data volume transmitted over the network. These findings underscore the critical role of compression in optimizing remote data access. Emerging cloud-optimized file formats [44] are anticipated to further enhance remote data access by eliminating redundant decompression/compression cycles inherent in OPeNDAP data access.

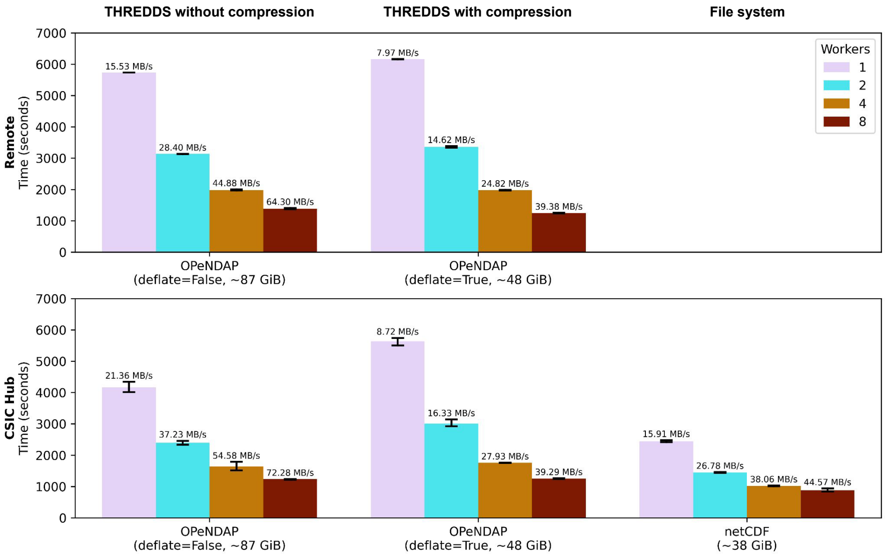
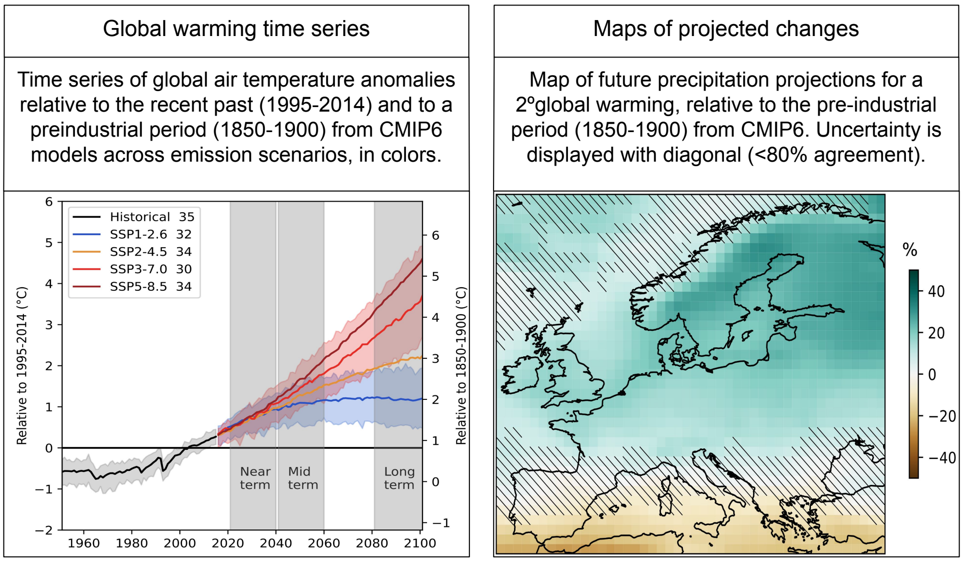


FIGURE 6.2: Experimental results of data retrieval from two Atlas DataLab setups: next-to-data deployment in the CSIC-DDC Hub (bottom) and remote data access from a user's workstation (top). The bars represent the mean retrieval time across ten experiment replicates for each worker configuration, with error bars indicating variability (minimum and maximum times). Inset numbers denote the volume of network data transferred. The first two columns illustrate THREDDS data access via OPeNDAP, while the last column shows results for in-situ access via file system. **Source:**Cimadevilla et al., 2025.

6.4 Illustrative Case Studies

The getting started notebooks of the GitHub Atlas DataLab [107], implemented both in Python and R, provide an introduction to the capabilities of the DataLab for both programming languages. Additional notebooks illustrate end-to-end reproducibility and reusability workflows for different case studies and key AR6 products [100], including those shown in Sections 6.4.1 and 6.4.2, and Figure 6.3: 1) global warming time series across different scenarios and 2) maps of projected changes and robustness for various Global Warming Levels (GWLs). The first example replicates the time series of global surface air temperature changes from Figure 4.2 (first panel) in WGI AR6 Chapter 4 [51], while the second example recreates an Interactive Atlas result for Europe, depicting projected precipitation changes at +3°C warming. These case studies are implemented in separate Python and R notebooks, providing flexibility to analyze different variables/indices, scenarios, GWLs, regions, and other parameters.

Additionally, users will find other notebooks for performing additional analyses, such as extracting results for land-only or pre-delimited regions (e.g., the IPCC-WGI Reference Regions [110]), or creating different types of visuals, such as stripes (`https://showyourstripes.info`) or global warming level plots [111].



Global warming time series

Time series of global air temperature anomalies relative to the recent past (1995-2014) and to a preindustrial period (1850-1900) from CMIP6 models across emission scenarios, in colors.

Maps of projected changes

Map of future precipitation projections for a 2ºglobal warming, relative to the pre-industrial period (1850-1900) from CMIP6. Uncertainty is displayed with diagonal (<80% agreement).

FIGURE 6.3: Case studies that can be reproduced from the DataLab [112]. The first example (a) replicates the time series of global surface air temperature changes from Figure 4.2 (first panel) in WGI AR6 Chapter 4 [51]. The second example (b) recreates an Interactive Atlas result for Europe, depicting projected precipitation changes at +3ºC warming. Locations with low model agreement are displayed using diagonal lines [26]. **Source:**Cimadevilla et al., 2025.

6.4.1 Global Change Time Series across Scenarios

The notebook `global_change_time-series_python.ipynb` provides step-by-step, annotated code to generate an annual time series of global temperature changes projected by CMIP6 (see Figure 6.3a). This analysis covers the period from 1950 to 2100 and includes various radiative scenarios representing low, medium, and high emissions (SSP1-2.6, SSP2-4.5, SSP3-7.0, and SSP5-8.5). For this case study, the notebook retrieves and processes data for the entire global domain, all available SSP scenarios, and CMIP6 models, thereby demonstrating the DataLab's capability to handle large datasets effectively. The graph is presented relative to the 1995-2014 period (left axis) and the pre-industrial 1850-1900 period (right axis). This replicates one of the figures from WGI AR6, specifically the first panel of Figure 4.2 in Chapter 4 [51]. Although slight differences exist in the ensemble of models used compared to AR6, the results are virtually identical. The notebook allows users to easily change the variable of interest, enabling the generation of similar plots for each index and variable included in the IPCC WGI Atlas dataset. The execution of this notebook can take between 15 minutes and 1 hour, depending on whether it is run from the local copy of the CSIC-DDC Hub or an external home network connection, respectively.

6.4.2 Maps of Changes and Robustness for Different GWLs

The notebook *Maps_of_change_under_global-warming-levels.ipynb* reproduces climate change anomaly maps for a specified global warming level (GWL). To do so, it extracts information on the time periods during which various GCMs reach different levels of global surface temperature warming (relative to the 1850-1900 period). This example builds on the information produced for the Atlas chapter in the framework of the IPCC FAIR activities described above [27], such as GWL periods for the different CMIP6 (also for CMIP5 and CORDEX), the IPCC-WGI Reference regions [110], land-sea masks, etc., thus ensuring end-to-end reproducibility.

This example focuses on the +3°C GWL by extracting the corresponding time windows from the SSP5-8.5 scenario. Data for each GCM is then loaded separately by requesting the corresponding warming level period, along with other analysis dimensions, such as the target season and variable (DJF precipitation in this example) and the coordinates delimiting the desired study area (Europe in this example). Historical data is also retrieved to compute the relative anomaly and associated uncertainty, both of which are presented in a map of the ensemble mean of Figure 6.3b. Uncertainty is calculated for both the simple and advanced methods described for the Interactive Atlas [26]. The notebook allows the reproduction of the final figure but also provides auxiliary relevant context information and figures. For instance, Figure 6.4 shows the periods when the different models forming the ensemble (in rows) reach the +3°C global warming. The execution of this notebook takes just a handful of minutes even when executed from remote workstations.

6.5 Conclusions

Reproducibility of data and climate figures is paramount within the context of the Intergovernmental Panel on Climate Change (IPCC), as it ensures transparency, credibility, and robustness in the findings that inform global climate policy. By enabling scientists and policymakers to verify results and reproduce analyses, reproducibility fosters trust in conclusions drawn from complex datasets and models. Moreover, it facilitates collaboration and accelerates scientific progress. In the context of the IPCC report, where decisions have far-reaching implications for climate action and adaptation, reproducibility ensures that all stakeholders can critically evaluate the evidence supporting recommendations.

To address these needs, a Data Laboratory (DataLab) has been developed to reproduce real examples from the Sixth Assessment Report (AR6) and assess the contents of the

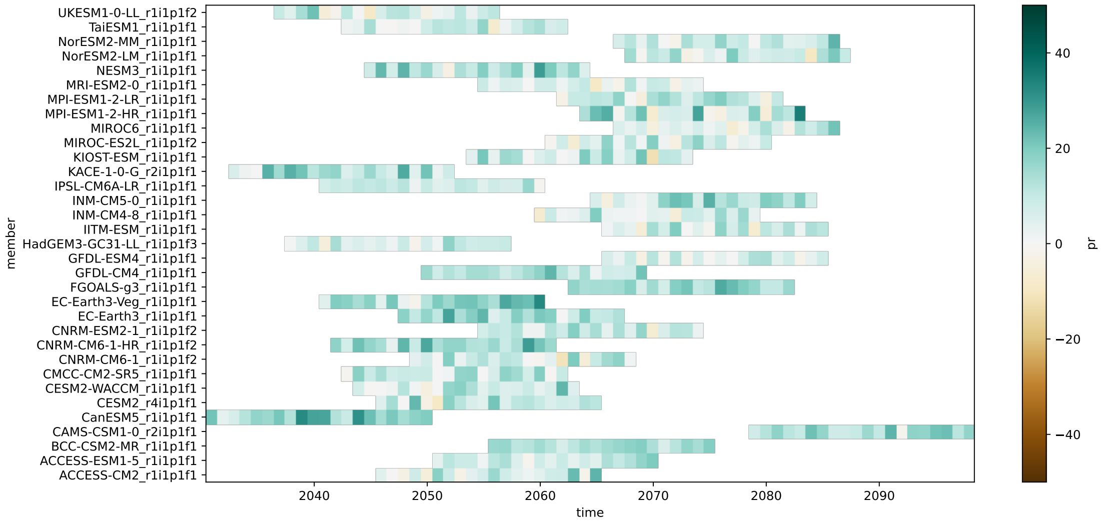


Figure 6.4: Auxiliary information showing the 20-year periods during which each model in the ensemble (rows) reaches the +3 °C global warming level under the SSP5-8.5 scenario. Each period is centered on the year when the respective model reaches the global warming level (GWL). The figure illustrates the variability in the timing of reaching the GWL across the ensemble, providing contextual information for the case study maps of projected changes. Colors represent the relative changes in precipitation relative to the reference period for each of the 20 years.

Interactive Atlas developed for the IPCC. This DataLab strikes an effective balance between advancing the current state of FAIR (Findable, Accessible, Interoperable, Reusable) data principles and managing the costs associated with supporting infrastructure. The DataLab explores and demonstrates technologies for data sharing, emphasizing their practical application in public climate services within the framework of climate change and the international exchange of Earth system data. Our initiatives address critical challenges related to the reproducibility of climate research, especially given the increasing volumes of data generated by model intercomparison projects.

To evaluate the performance of various data access methods—specifically in-situ versus remote access, a performance analysis experiment was conducted. This experiment highlights the advantages and costs associated with different data access approaches, demonstrating the benefits of executing computations close to the data source. Future work could explore variations in performance and reliability when accessing data from other regions, particularly those with slower internet connections, such as developing countries. Such studies would provide valuable insights into the challenges faced in regions with less robust digital infrastructure and help design more inclusive solutions for global accessibility. Additionally, this analysis could identify potential optimizations to improve performance for users in diverse geographic locations.

The design of the DataLab aims to enhance the FAIR data practices of the IPCC-WGI AR6 Interactive Atlas Dataset. Comprehensive documentation and accompanying notebooks guide users through the process of analyzing this data, encompassing steps such as locating, loading, manipulating, and visualizing real-world data outputs. Real examples from AR6 have been presented, demonstrating reproducibility from different workstations and infrastructures with internet connectivity.

Overall, the DataLab represents a central contribution of this thesis and a significant step forward in enhancing the reproducibility and reusability of climate data and analyses within the IPCC AR6 framework. In line with the objectives of the thesis, the DataLab advances the concept of a climate data infrastructure by evolving it into a true research environment, where data storage is complemented by FAIR-aligned workflows, virtualized access, and reproducible computational practices. This integration strengthens the reproducibility of AR6 products and lowers barriers for researchers to interact with authoritative climate information. Its relevance extends beyond AR6, as the methodologies and technologies demonstrated here establish a transferable model for other data-intensive scientific domains seeking to enhance transparency, collaboration, and reproducibility.

Despite these advances, limitations remain: users still face burdens related to file-level manipulation. To address this, the next chapter introduces the ESGF Virtual Aggregation methodology, which enables ARD based on virtual analysis-ready data and provides remote data access to climate data in the ESGF. Finally, Chapter 8 integrates this methodology into a new DataLab, demonstrating how the reproducibility of AR6 results can be achieved directly from ESGF, thus consolidating the overall contribution of the thesis to the field of technology in climate data-intensive science.

Chapter 7

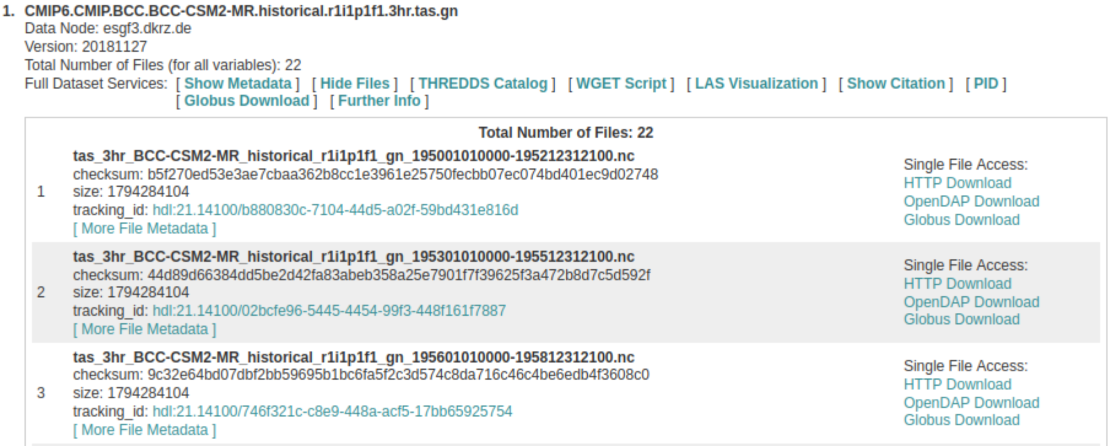
The ESGF Virtual Aggregation

7.1 Introduction

The importance of effective and efficient climate data analysis continues to grow as the demand for understanding the climate system intensifies. Traditionally, climate data repositories have been structured as file distribution systems, primarily designed to facilitate file downloads. However, this conventional approach presents significant challenges for climate data analysts, often requiring them to dedicate substantial time to managing data access—time that could be better spent on their core research. The Earth System Grid Federation (ESGF) is a global infrastructure and network that consists of internationally distributed research centers that follows this approach [76, 77].

Several methodologies are emerging to streamline climate data analysis, harnessing the potential of Analysis Ready Data (ARD, Dwyer et al. 56), remote data access and new formats and infrastructures for climate data storage [44]. The ESGF Virtual Aggregation (ESGF-VA) [113], an innovative method for climate data analysis leveraging rarely exploited aspects of the ESGF. It is based on the capabilities of virtual aggregations built on top of the architecture of the ESGF and designed to be included in the federation as an external service.

In the ESGF, research centers collectively serve as a federated data archive, supporting the distribution of global climate model simulations representing past, present, and future climate conditions [12]. The ESGF enables modeling groups to upload model output to federation nodes for archiving and community access at any time. To facilitate multi-model analyses, the ESGF ensures standardization of model output in a specified format. It also facilitates the collection, archival, and access of model output through the ESGF data replication centers. As a result, the ESGF has emerged as the primary



1. **CMIP6.CMIP.BCC.BCC-CSM2-MR.historical.r1i1p1f1.3hr.tas.gn**
Data Node: esgf3.dkrz.de
Version: 20181127
Total Number of Files (for all variables): 22
Full Dataset Services: [Show Metadata] [Hide Files] [THREDDS Catalog] [WGET Script] [LAS Visualization] [Show Citation] [PID] [Globus Download] [Further Info]

Total Number of Files: 22

1 **tas_3hr_BCC-CSM2-MR_historical_r1i1p1f1_gn_195001010000-195212312100.nc**
checksum: b5f270ed53e3ae7cbaa362b8cc1e3961e25750fecbb07ec074bd401ec9d02748
size: 1794284104
tracking_id: hdl:21.14100/b880830c-7104-44d5-a02f-59bd431e816d
[More File Metadata]
Single File Access:
HTTP Download
OpenDAP Download
Globus Download

2 **tas_3hr_BCC-CSM2-MR_historical_r1i1p1f1_gn_195301010000-195512312100.nc**
checksum: 44d89d66384dd5be2d42fa83abeb358a25e7901f7f39625f3a472b8d7c5d592f
size: 1794284104
tracking_id: hdl:21.14100/02bcfe96-5445-4454-99f3-448f161f7887
[More File Metadata]
Single File Access:
HTTP Download
OpenDAP Download
Globus Download

3 **tas_3hr_BCC-CSM2-MR_historical_r1i1p1f1_gn_195601010000-195812312100.nc**
checksum: 9c32e64bd07dbf2bb59695b1bc6fa5f2c3d574c8da716c46c4be6edb4f3608c0
size: 1794284104
tracking_id: hdl:21.14100/746f321c-c8e9-448a-acf5-17bb65925754
[More File Metadata]
Single File Access:
HTTP Download
OpenDAP Download
Globus Download

FIGURE 7.1: ESGF listing three files of a CMIP6 dataset. A common practice in the ESGF consists in splitting the dataset into many files along the time dimension. Smaller files are easier to manage in the federation but performing data analysis becomes harder. This image is a screenshot obtained from the ESGF web portals. Credit is attributed to the ESGF partners supporting these portals. For further details, please refer to https://esgf.llnl.gov/acknowledgments.html.

distributed data archive for climate data, hosting data for international projects such as CMIP6 [14] and CORDEX [79]. It catalogues and stores tenths of millions of files, with more than 30 petabytes of data, distributed across research institutes worldwide [80], and it serves as the reference archive for Assessment Reports (AR) [81] on Climate Change produced by the Intergovernmental Panel on Climate Change (IPCC, [82]).

The significant growth of data poses a scientific scalability challenge for the climate research community [12]. Contributions to the increase in data volume include the systematic increase in model resolution and the complexity of experimental protocols and data requests [13]. Although the ESGF infrastructure is designed as a file distribution system, scientific research often requires multidimensional data analysis on datasets encompassing multiple variables, spanning the entire time period, multiple model ensembles and different climate model runs. Several ongoing developments in scientific data research try to address the issues of growing data volume and variety and provide new approaches to data analysis.

Climate Analytics-as-a-Service (CAaaS, [93]), GeoDataCubes [55, 114], cloud native data repositories [44] and Web Processing Services (WPS, 2015) are some of the systems that are being used to improve climate data analysis workflows. The data consolidation process in building these new systems may involve data duplication of an enormous volume of data, incurring in large costs of operational and storage requirements. However, the cost of data duplication is assumed to be compensated by a gain in efficiency in information synthesis. In order to overcome these costs, several technologies do allow the creation of virtual datasets, which provide ARD capabilities without the need to

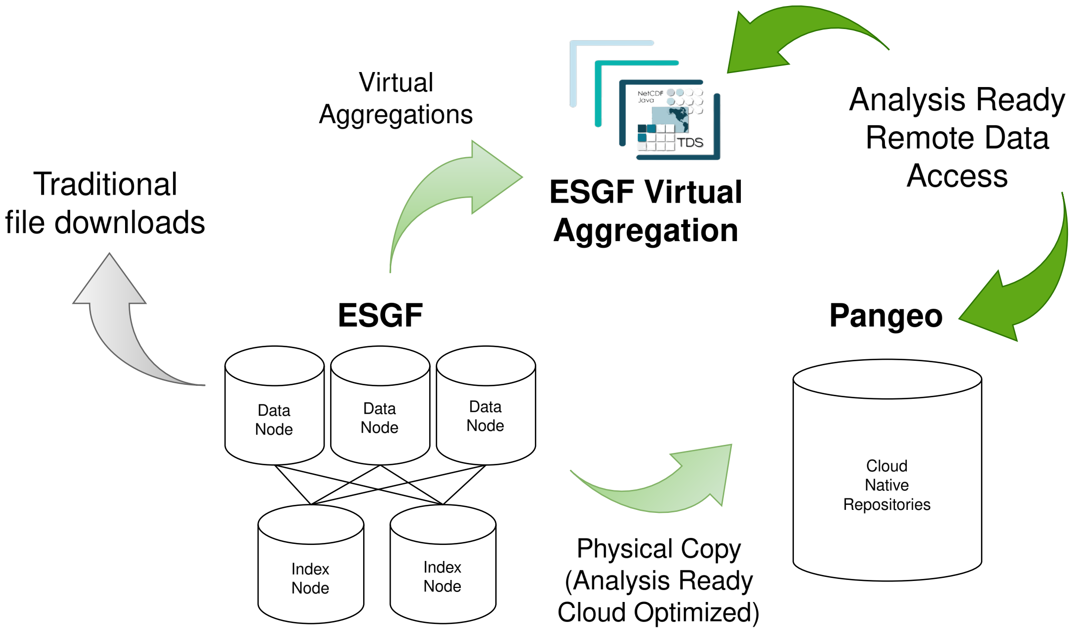

Virtual Aggregations
NetCDF Java
TDS
ESGF Virtual Aggregation
Analysis Ready Remote Data Access
Traditional file downloads
ESGF
Pangeo
Data Node
Data Node
Data Node
Index Node
Index Node
Cloud Native Repositories
Physical Copy (Analysis Ready Cloud Optimized)

FIGURE 7.2: The ESGF Virtual Aggregation aims to be a sustainable bridge that eases the technological transition between the current state of the ESGF and more ground-breaking and expensive solutions, based on data replication, such as cloud native repositories. **Source:**Cimadevilla et al., 2025.

duplicate the original data sources. These provide the opportunity for more sustainable approaches to enhancing climate data analysis capabilities.

The ESGF-VA serves as a bridge between the current implementation of the ESGF and the development of cloud native data repositories for climate research. Figure 7.2 shows how it fits the current ecosystem. It is implemented as an additional value-added user service on top of ESGF, running in conjunction with other value-added user services such as citation and PID handle services [78]. To satisfy sustainability requirements, a balanced strategy is adopted to manage operational costs and complexity. The ESGF-VA aims to advance the sharing and reuse of scientific climate data by building a catalog of logically aggregated datasets, facilitating remote access to the distributed data hosted in the ESGF. It offers data access (remotely) to convenient and adequate views of the data (ARD) that allow ad hoc complex queries without the need to duplicate data sources.

7.2 Implementation

The implementation of the ESGF-VA involves the following steps:

1. The search process involves querying the ESGF catalog and indexing service to obtain dataset information and metadata, which is then stored in a local database.

2. The aggregation process queries the local database to create virtual datasets (NcMLs) for the entire federation. These are the ARDs that the user utilizes for remote climate data analysis.

Figure 7.3 shows how NetCDF files from the ESGF that belong to the CMIP6 project are distributed between the virtual datasets. Most virtual datasets contain few references to NetCDF files inside (<=100) although some virtual aggregations provide access to hundreds or even thousands of NetCDF files. Table 7.1 shows the ratio of NetCDF per NcML for each CMIP6 activity [14]. The following sections detail the implementation of both the search and aggregation processes.

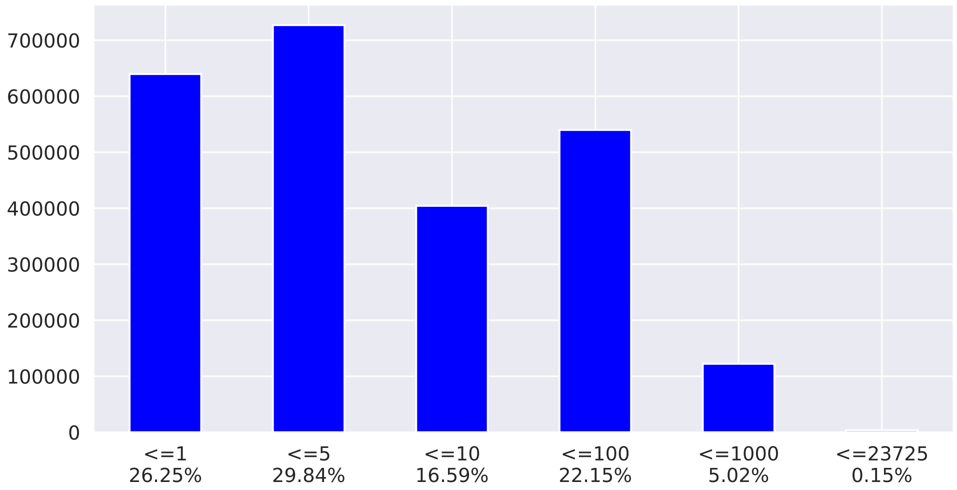


FIGURE 7.3: Distribution of NetCDF files in the virtual datasets (NcMLs). Most of the virtual aggregations are made of a relatively small number of files although some virtual datasets spawn hundreds or thousands of files. **Source:**Cimadevilla et al., 2025.

7.2.1 The ESGF Search Process

For the search process, the ESGF Search REST API [77] is used by the client to query the contents of the underlying search index, returning results matching the given constraints in the whole federation. The search service provides useful metadata that allow clients to obtain valuable information about the datasets being queried. However, in the context of the ESGF-VA, it is not as efficient as one would like - sufficient for the first implementation and experiments described here, but in an operational context one would want to see time coordinate information held in the index. This is because applications otherwise

CMIP6 activity	NcMLs	NetCDFs	Ratio (NetCDFs / NcML)
ISMIP6	2570	10864	4.23
GMMIP	9489	587501	61.91
LS3MIP	16041	188533	11.75
OMIP	17009	441578	25.96
PAMIP	19824	4931240	248.75
CDRMIP	21189	395444	18.66
PMIP	26277	645989	24.58
GeoMIP	28470	184666	6.49
FAFMIP	41324	208881	5.05
LUMIP	57140	581573	10.18
HighResMIP	63359	5806778	91.65
RFMIP	81548	745604	9.14
CFMIP	81599	309421	3.79
C4MIP	81964	847376	10.34
DAMIP	134708	3482721	25.85
AerChemMIP	199307	1850392	9.28
ScenarioMIP	250591	17317882	69.11
CMIP	505733	19090708	37.75
DCPP	506085	8152594	16.11
Total	**2144227**	**65779745**	**-**

Table 7.1: Number of virtual aggregations (NcMLs), NetCDF files for which metadata was retrieved from the federation and ratio of NetCDF per NcML generated for CMIP6 in the ESGF Virtual Aggregation. Note that the distribution of number of references to NetCDFs files on NcMLs does not follow a uniform distribution (see Figure 7.3).

need to read such information from each and every file in an aggregation, which may be a significant overhead, before any actual data transfer.

The search process is performed by an iterative querying the ESGF search service, requesting small chunks of data that are manageable by the service. The search service limits the number of records that can be obtained from a single request to ten thousand elements. Since the federation contains information on the order of tens of millions of records, several requests need to be made. The results are stored in a local SQL database and multiple ESGF Virtual Dataset labels are assigned to the record, in order to identify the virtual dataset in which the records participate in different virtual aggregations.

7.2.2 The Aggregation Process

The aggregation process is responsible for generating the virtual aggregations and mapping multiple ESGF individual files and their metadata to the appropriate virtual datasets. Although the number of records could be overwhelming, the use of SQL indexes allows the aggregation process to quickly retrieve the granules that belong to the different virtual datasets. The result from the aggregation process in the ESGF Virtual Aggregation is a collection of NcML files that represent the virtual datasets. The virtual datasets are stored in different directories in order to provide appropriate organization. Each virtual dataset is labeled with the data node to where each of the granules that form the virtual dataset belong. Additionally, the virtual datasets are generated in such a way that replicas from the same virtual dataset are easily identifiable.

The virtual datasets of ESGF-VA are made of two kind of aggregations. First, the ESGF *atomic dataset* aggregation is generated by concatenating the time series of each variable along the time dimension. This concatenation does not increment the rank of dimensions of the multidimensional array that represents the variable, it only increases the size of the time dimension. This kind of aggregation is ignored in time independent variables such as orography. Then the variables are aggregated by creating a new dimension that represents the variant label (i.e. ensemble members), the different model runs of a climate model. The rank of dimensions is incremented by one, to accommodate a dimension for the ensemble or variant label. It is important to note that for this kind of aggregation to be performed properly, climate variables involved must share a spatial and temporal coordinate reference system, with the exact same spatial coordinate values. If that were not the case, the resulting multidimensional array would expose incorrect data. Figure 7.4 displays the result of a data analysis which saves the user from accessing multiple NetCDF files.

Some issues were found during the development of the ESGF Virtual Aggregation. These involve the usage of the version facet in the publication process of the ESGF and data discrepancies between the ESGF data files and the data cubes offered by the ESGF Virtual Aggregation. In the first place, the version facet is supposed to distinguish between allegedly equal datasets that have changed due to different kinds of errors, such as incorrect data due to bad model execution or incorrect publication process. In practice, the version facet may, in some cases, end up dividing granules that should belong to the same aggregation, due to inappropriate usage of the facet. From an ESGF-VA point of view this could be avoided by using the latest value of the version facet, but that would lead to issues with maintenance.

7.3 Performance

To investigate the performance of accessing data using the ESGF-VA, an experiment was carried out to examine data access performance from a xarray client. This limited experiment is enough to show some of the benefits of, and issues with, the ESGF-VA. The experiment was carried out with the ESGF-VA utilising OPeNDAP, and for comparison, with Kerchunk aggregation. In both cases, virtual aggregations were generated first,

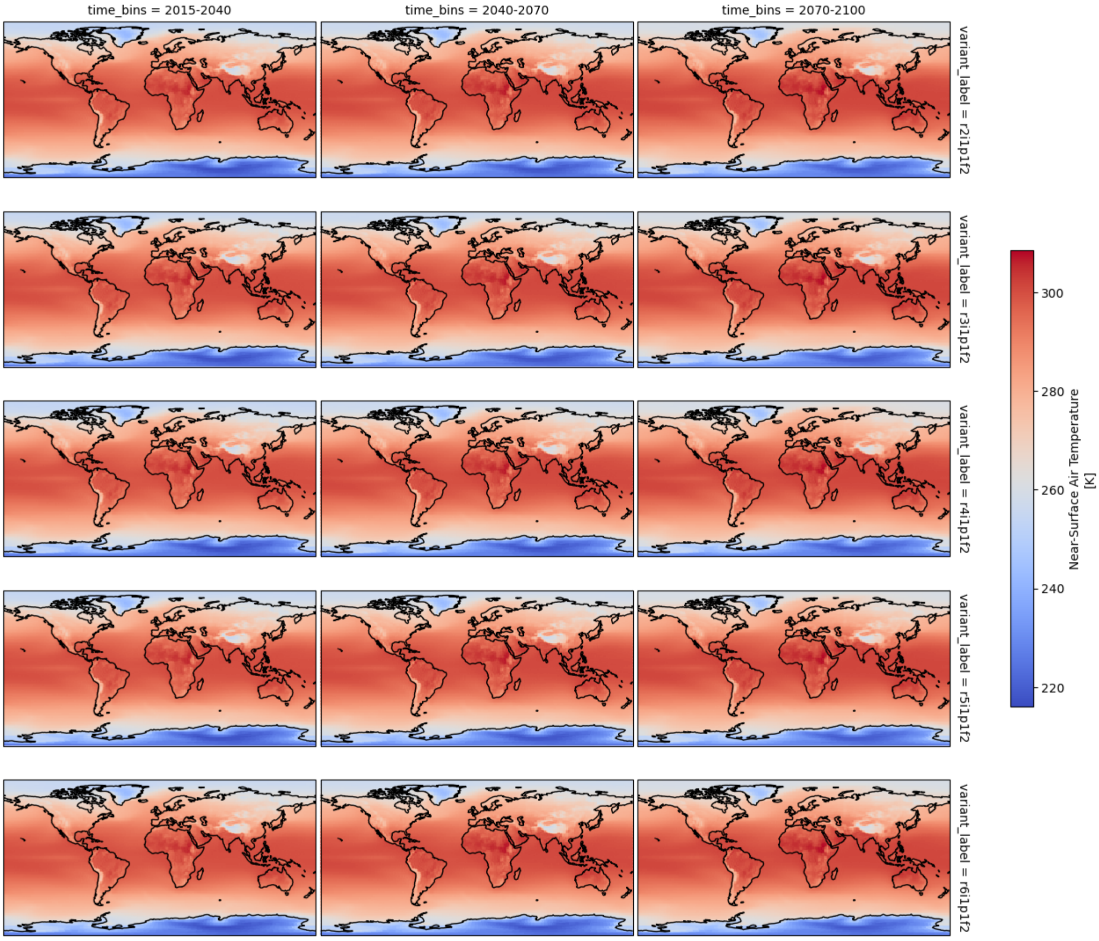


FIGURE 7.4: Mean Near-Surface Air Temperature for different time periods and different model runs. The code needed to obtain this result is minimal, enabled by the capabilities of the data cube. Because all the information is stored in one single ESGF Virtual Aggregation dataset, only one data source is needed to perform the data analysis. The data is fetched directly from ESGF data nodes on a remote data access basis. **Source:**Cimadevilla et al., 2025.

with each performed using varying numbers of Dask worker processes to test potential scalability, albeit in a context where server scalability is known to be limited, and little or no contention from other users was expected. Here *Kerchunk* refers to the use of Kerchunk files to access individual blocks of compressed data via Zarr and other Pangeo middleware on the client talking directly to an ESGF HTTPS server, whereas OPeNDAP is the vanilla usage of the ESGF-VA on the client talking to an ESGF OPeNDAP server.

The experiment was simple: a dataset comprising the entire atomic datasets (80 years) of daily values for one spatially two-dimensional variable (surface temperature, *tas*) from each simulation member was read, and a global mean of that data was calculated. The actual calculation was done on a cloud hosted virtual machine in Spain at Instituto de Física de Cantabria (IFCA), while the data was read from each of four ESGF servers. In each case, the dataset was chunked for Dask into segments of 400 daily values (so each chunk was about 50 MB in memory, the default maximum limit for OPeNDAP), in order to examine the benefit of using multiple Dask workers. The experiment was repeated five times on each of the ESGF servers for 2, 4, and 8 Dask workers. However, it was not possible to get OPeNDAP results from all four servers, or to get a fullset from each of the servers - the reasons for this are discussed below. No attempt was made to mitigate file system caching in this design, as although it could have affected the comparison, in practice the I/O time for reading the data (∼10 GB on disk, ∼20 GB on memory) would be small compared to the overall times reported.

The results are shown in figure 7.5. There are several obvious results: when using Kerchunk, considerable benefit was gained by using more workers, and that data nodes close to Spain (where the calculation was done) yielded much faster outcomes than remote data nodes. In each case, OPeNDAP is much slower than Kerchunk, and the benefit of geographical proximity on the OPeNDAP results is much less obvious (e.g. using 8 workers to process data loaded from Australia is faster than using 8 works on data from the UK, but for 2 workers, it is much faster to use the UK data). Unfortunately, DKRZ do not offer the OPeNDAP service and LLNL took the service down after the first experiments and before the replicas were added. It is also clear that the OPeNDAP results from the CEDA server are anomalous in terms of having no dependency on the number of workers.

As already noted, proximity matters. The benefit of client-side decompression, as implemented by Kerchunk, is evident. A priori, the OPeNDAP results might have been expected to be roughly a factor of two slower, given that OPeNDAP decompresses server-side and transmits the uncompressed data over the network, which is approximately what is observed at LLNL and NCI. The CEDA OPeNDAP results are anomalous, so no attempt is made to explain the disparity between Kerchunk and OPeNDAP speeds observed there. For this experiment, with the fastest times recorded (44s and 49s from CEDA and DKRZ, respectively), it is clear that the bottleneck lies in the data flow across the wide area.

Similar experiments with other data highlighted some suboptimal data practices within the ESGF archive. A significant number of CMIP6 datasets stored in the ESGF exhibit poor chunking configurations, specifically related to the time coordinate. Chunking in HDF5 is a crucial technique for optimizing data access performance. It involves organizing how data is stored on disk, enabling different arrangements based on desired data access patterns. Proper chunking can greatly enhance data access efficiency, similar to how SQL indexes improve database query performance. Conversely, incorrect or inappropriate chunking choices can have a detrimental impact on data access performance. Notably, the CMIP6 files within the ESGF often displayed a chunking configuration of *(1,)* for

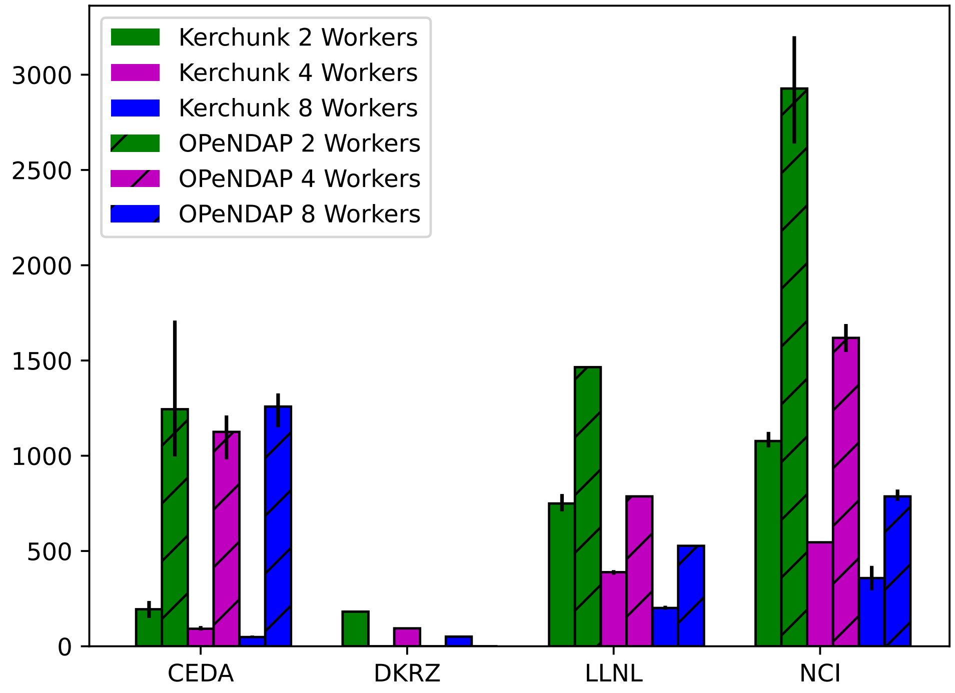


FIGURE 7.5: The results of the experimental retrieval of data for meaning using Kerchunk and OPeNDAP from a client in Spain (IFCA) to servers in the UK (CEDA), Germany (DKRZ), the US (LLNL) and Australia (NCI). The bars show the mean time, in seconds, taken across experiment replicants for each configuration of number of workers. Where error bars are shown, these reflect the minimum and maximum times taken. Kerchunk data is shown without hatching, and OPeNDAP data with. There is no OPeNDAP data for DKRZ, and no replicants – and hence no error bars – for the OPeNDAP experiments using the LLNL server. **Source:**Cimadevilla et al., 2025.

the time coordinate, resulting in severe degradation of dataset access times (figure 7.6). Sub-optimal chunking configuration negatively affected the efficiency of data retrieval and subsequent analysis tasks.

7.4 Conclusions

The ESGF Virtual Aggregation (ESGF-VA) has been introduced, demonstrating how it can be used to obtain data from the existing ESGF OPeNDAP servers. In doing so, the ESGF federated index and the ESGF OPeNDAP endpoints are shown to provide capabilities beyond conventional file search and download. By enabling remote data analysis over virtual analysis ready data, the use of the ESGF-VA could enhance the efficiency and productivity of climate data analysis tasks. It could empower researchers to access and analyze data directly within the ESGF environment, eliminating the necessity

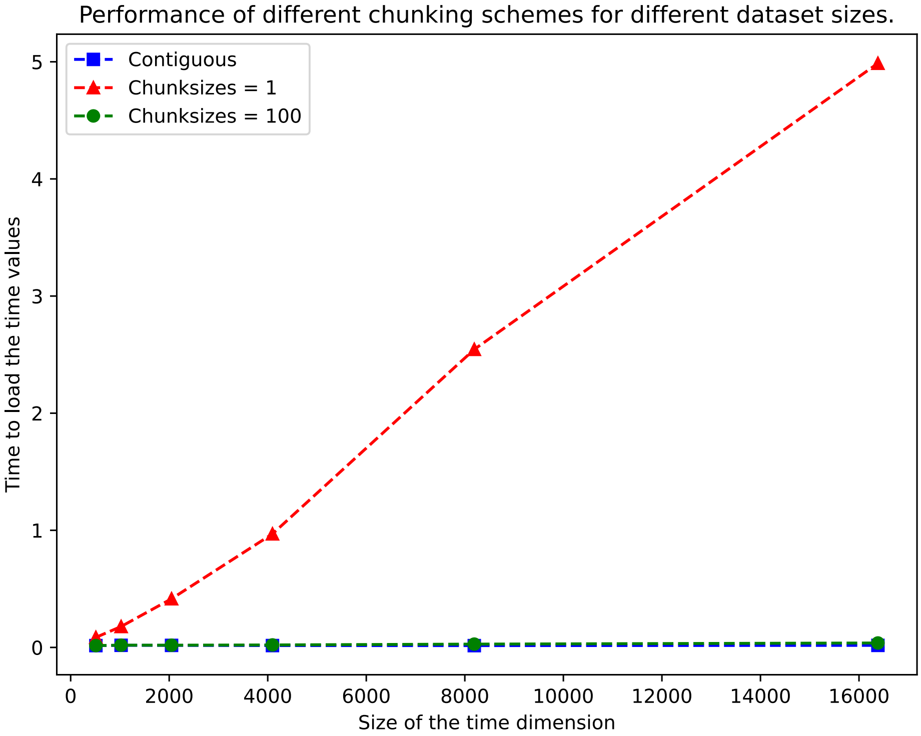


FIGURE 7.6: Required time to read a temporal coordinate in function of storage type. Contiguous storage does not incur into performance issues. If chunking storage with a bad chunking scheme is used, performance quickly deteriorates. **Source:**Cimadevilla et al., 2025.

for time-consuming data transfers and facilitating more streamlined and effective climate data analysis workflows.

The virtual datasets provided by the ESGF-VA facilitate an aggregated view of the time series as well as the ensemble model members of a particular model. Thus, data analysis comparing different runs of the same model can be performed by loading only the view of one dataset. In doing so, the details of the aggregation are hidden completely from the user, who sees the dataset as a single NetCDF. Using the OPeNDAP endpoints of the federation, data analysis can be performed from anywhere. While this implementation of the ESGF-VA exploits NcML and NetCDF-java, the concept is readily extensible across any NetCDF client which supports OPeNDAP - i.e. any which utilise the NetCDF library itself, rather than directly using HDF5. However, because OPeNDAP performs chunk decompression in the server, it is not as efficient as other data access methods as more data is sent over the network.

The creation of the virtual aggregations presented in this work follows a much more maintainable approach than alternatives focusing on duplication of the data, such as cloud native repositories. The storage requirements of the virtual aggregations are minimal compared to the relative size of the raw data. In addition, the generation of the virtual aggregations can be performed in few hours, where most of the time is spent querying the ESGF distributed index. As the ESGF-VA aggregation information is obtained directly from the existing ESGF index, it can be generated much faster than the process needed to generate Kerchunk indices, which requires access to each file. The speed of creation of new virtual aggregations, coupled with the lack of actual data duplication, means that the system can cope well with an environment where datasets are being updated as data processing issues are found and fixed, since the ESGF-VA can be quickly updated. However, whatever system is used to create analysis ready data, it is necessary to know that such updates are necessary - it would be helpful for a future ESGF to have some sort of automated alert system for data updates.

Certain issues regarding data distribution of the ESGF were identified during the creation of the ESGF Virtual Aggregation. There is often inconsistent use of the version facet, and a significant portion of the data stored in the federation does not adhere to best practices regarding the chunking of HDF5. In the first place, the version facet is supposed to distinguish between allegedly equal datasets that have changed due to different kinds of errors, such as incorrect data due to bad model execution or incorrect publication process. In practice, the version facet may, in some cases, end up dividing granules that should belong to the same aggregation, due to inappropriate usage of the facet. From an ESGF-VA point of view, this could be avoided by using the latest value of the version facet, but that would lead to issues with maintenance. There may be value in both providing better guidance to modelling centres about how to use version facets and in adding some chunk checking to future ingestion processes.

7.4.1 Discussion

The performance analysis presented in this work suggests a declining interest from the ESGF community in supporting OPeNDAP, given the instability of this service compared to data access based on HTTP. While the details of the individual server configurations are unknown, the fact that the CEDA OPeNDAP results are so odd, and that both DKRZ and LLNL no longer offer OPeNDAP servers, it is plausible to conclude that it is a) difficult to deploy OPeNDAP and b) currently not enough usage to justify it. However, our results suggest that there may yet be mileage in deploying properly configured OPeNDAP services in the future ESGF (maybe with a different server, such as that of Gallagher et al. 115) - at least until such time that remote direct access to chunks via HTTP is available to a much greater proportion of NetCDF clients.

In doing so, the use of HTTP compression could mitigate the issue of server side decompression of the chunks. While Kerchunk represents a promising alternative for optimizing data access—offering very good performance by transferring only compressed chunks over the network—its use requires generating reference files through inspection of the internal structure of NetCDF datasets, which becomes impractical at the scale of the ESGF-VA. In contrast, OPeNDAP was chosen because it is widely supported standard within the ESGF ecosystem, providing remote data access with subsetting. HTTP compression is currently supported by NetCDF clients but is currently provided by few, if any, ESGF nodes. Finally, it would also be helpful if the time coordinate information could be stored in the ESGF index to be used by virtual aggregation clients in a way to avoid the need to read time coordinate values from each file when opening the virtual dataset.

While the ESGF-VA provides many benefits for users, albeit with the cost of moving the uncompressed data selections, such benefits would only transpire if there was sufficient server capacity to support demand. Although the ESGF-VA itself requires no change to the ESGF architecture itself, support for access to ESGF data via the OPeNDAP protocol is currently delivered by the use of THREDDS Data Server (a Java web application). While scaling out server infrastructure with THREDDS is possible, it requires both sufficient hardware and significant configuration knowledge. The pros and cons of wider usage of the ESGF-VA or similar OPeNDAP based tools and the consequential need for server capacity and issues of configuration should form part of future ESGF discussions.

Chapter 8

A DataLab for Remote Data Access and Virtual Analysis Ready Data

8.1 Introduction

The operational challenges and complexities of data access within the ESGF—primarily limited to file downloads—hinder the provision of reproducibility and reusability services for scientific products developed in the context of IPCC Assessment Reports. These complexities may be assumed by intermediate infrastructures that provide reproducibility and reusability for a small subset of the climate data available in the ESGF to reproduce a small number of user cases that appear on the ARs. This is the case of the IPCC Interactive Atlas DataLab [104], which is based on a preprocessed dataset of 500 GB extracted from downloading and processing over 200 TB of ESGF data to provide reproducibility and reusability of products of the Interactive Atlas included in the AR6 6. These services have a significant cost in terms of deployment and maintenance, due to the complex process of creating a proper dataset that sustains the service and the infrastructure required to make it accessible and operational to end users.

Alternatively, initiatives like Pangeo replicate ESGF data on the premises of cloud providers. By creating Analysis-Ready Cloud-Optimized (ARCO) copies of ESGF datasets, Pangeo enables more efficient data access, allowing researchers to perform computations directly in the cloud without the need to download large datasets. This approach enhances scalability, facilitates collaborative research, and leverages modern cloud computing to accelerate climate model analysis. By reducing barriers to data access and processing, cloud-based initiatives contribute to more effective and timely climate change assessments [44]. Climate infrastructures based on the accessibility of climate data from cloud providers reduce barriers to scientific progress by fostering a collaborative community centered on shared scientific workflow knowledge through *cookbooks* [116]. However, duplicating data to cloud providers incurs significant storage costs, as transferring and storing the vast amount of data available in ESGF is non-trivial. The total volume of unique CMIP6 data in ESGF reaches up to 10 PB, expanding to 20 PB when including replicated datasets within ESGF.

ARD can also be achieved using techniques that avoid data duplication by providing virtual aggregations 4.4. These aggregations create the illusion of a single data source, while in reality, the data is backed by multiple files that have been combined into a single logical aggregation. Virtual aggregations work by creating an intermediate file of negligible size relative to the data sources it references. This intermediate file contains the instructions that compatible applications or middleware understand to generate the aggregated view of the data sources for the user. The advantage of relying on virtual dataset capabilities is that data duplication is avoided, and the existing infrastructure can be reused to achieve ARD capabilities without incurring significant costs. Examples of virtual aggregations that follow this approach include, but are not limited to, NcML [66] and Kerchunk [67]. The ESGF Virtual Aggregation represents such an endeavor to facilitate climate data analysis within the ESGF 7. Figure 8.1 illustrates these three different methods for performing data analysis.

This chapter introduces a data laboratory for climate analysis of data stored within the ESGF premises. This laboratory takes advantage of the methodology proposed in the ESGF Virtual Aggregation 7 to allow climate data analysis without intermediate climate data infrastructures nor duplication of the original data. Examples of climate data analysis are showcased, along with an evaluation of the data laboratory's capabilities, based on a comparison between data-lightweight workflows using subsetting and data-intensive workflows reproducing examples from the IPCC Interactive Atlas DataLab 6. The limitations of the ESGF regarding remote data access technology are highlighted, along with justification for adopting cloud repositories to improve accessibility and usability of climate data.

8.2 The ESGF-VA Data Laboratory

The ESGF-VA Data Laboratory leverages the advantages of virtual and remote analysis ready data to enhance data access within the ESGF. It is built on top of the ESGF Virtual Aggregation to provide remote and virtual analysis ready data 7. The ESGF-VA provided the methodology to generate analysis ready data in the form of NcML files, which can

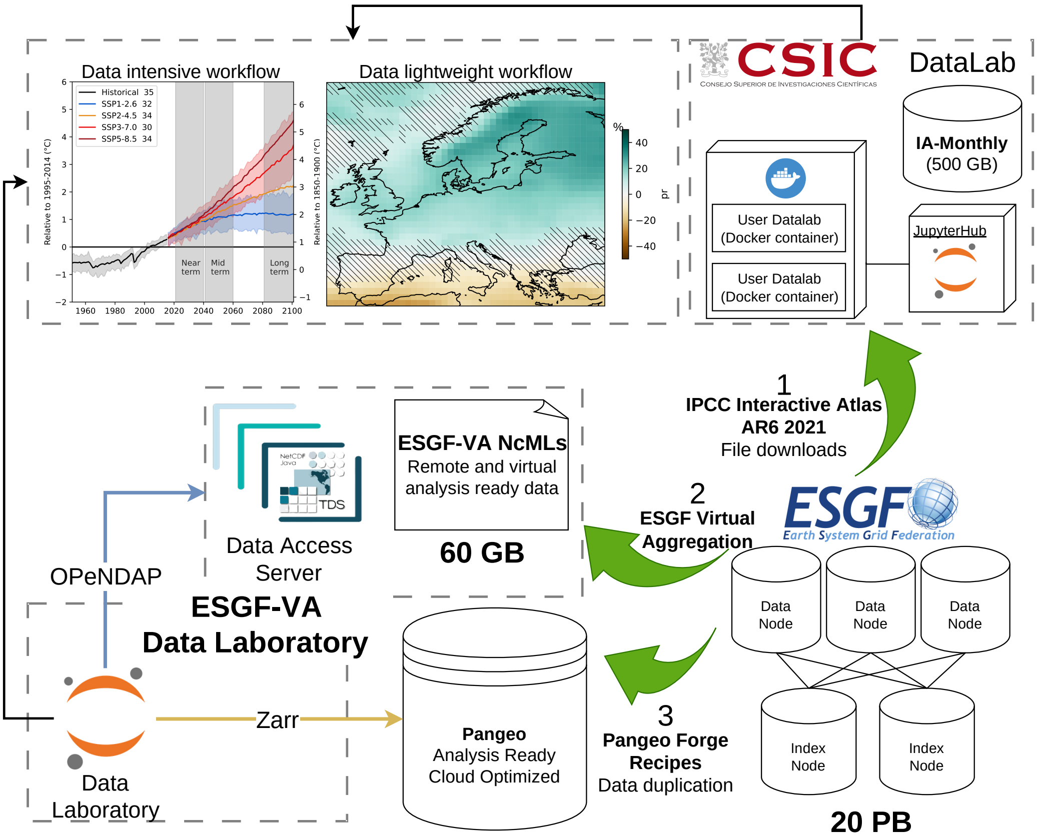


Figure 8.1: Overview of the different methods for performing climate data analysis: 1. traditional file downloads from ESGF are processed and made available in a dedicated platform, 2. virtual aggregations with remote data access to the ESGF enabled and 3. data duplication to cloud providers in cloud-optimized formats.

be consumed by compatible clients. An NcML file consists of logical aggregations of NetCDF files hosted in ESGF data nodes. By consuming NcML files, climate data analysis tasks don't need to deal with greety details of file system manipulation, greatly simplifying programming tasks. The ESGF-VA Data Laboratory consists of two main components: the data access server and the data laboratory. The data access server manages data transfers from ESGF data nodes. The data access server is populated with ARD generated by the ESGF Virtual Aggregation, climate datasets that have been virtually aggregated based on logical aggregations from source NetCDF files in the ESGF. Figure 8.2 illustrates the different options available to users who wish to perform data analysis on climate model output from the latest phase of CMIP (CMIP6).

The landing page of the ESGF-VA Data Laboratory is a GitHub repository that contains several launchers, allowing users to run the data laboratory on different infrastructures. The most accessible option, available to anyone on the Internet, is the BinderHub launcher, which leverages freely available cloud computing resources. This option fairly

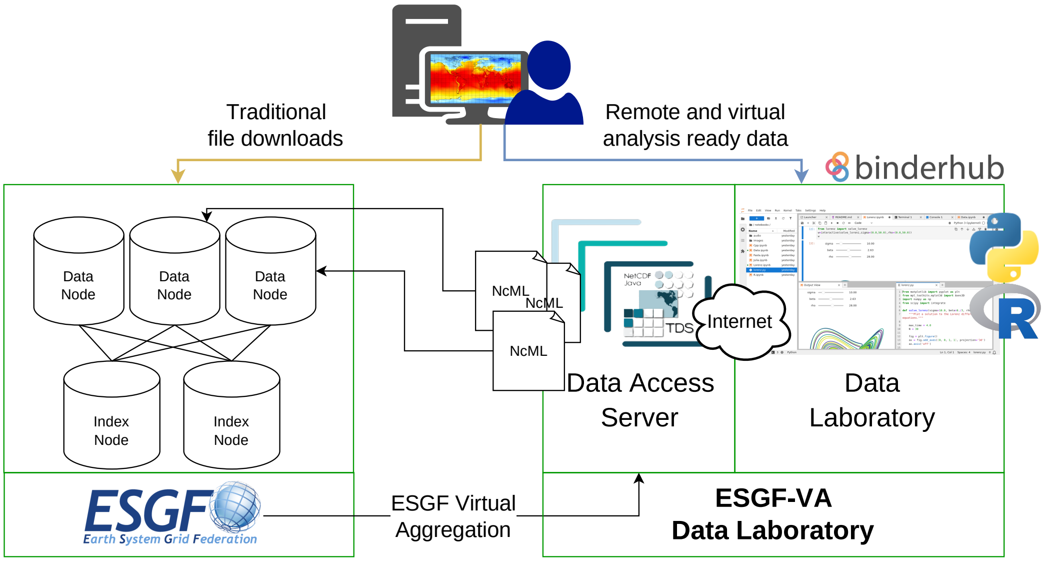


FIGURE 8.2: Users can choose from different approaches to perform climate data analysis. File downloads are the traditional mechanism used by ESGF to provide access to climate data. The ESGF-VA Data Laboratory offers an enhanced analysis experience by providing ARD on top of a laboratory environment preloaded with climate data analysis software libraries, enabling seamless access to climate data. Based on ESGF-VA, data is retrieved from ESGF data nodes using remote data access.

democratizes data access, as it is available to anyone on the Internet. However, the computational capacity of these infrastructures is limited, due to the inherent constraints of offering unrestricted computing power to all users. It should be noted that, thanks to the advantages of remote data access, deploying the data laboratory on such infrastructures does not involve any data duplication. Data is fetched directly and on-demand from ESGF data nodes when the notebooks are executed. Because remote data access inherently supports data subsetting, only the data required for the specific climate analysis task is transferred over the network from the origin servers to the computing clients.

The ESGF-VA Data Laboratory offers cookbooks that can be executed in JupyterLab environments, which may be deployed in different infrastructures such as local workstations, HPC systems or in the cloud [98]. The data laboratory provides an inventory of the climate datasets available on the data access server, making it easier for users to locate data sources. Additionally, software packages for data analysis in both Python and R are pre-installed and made available to users via Reproducible Execution Environment Specifications (REES). Thus, all software dependencies are made available to users of the data laboratory, allowing them to focus on their research and avoid tasks related to software dependency management and data handling.

8.2.1 Model Evaluation

This section illustrates a model evaluation task focused on studying the agreement of the model members on precipitation outputs of the CanESM5 global climate model [117] in the region of Europe. The data analysis tasks computes relative anomalies of precipitation for two future scenarios relative to the historical period. Due to the convenience of dealing with ARD datasets and remote data access from the ESGF Virtual Aggregation 7, this workflow saves the user from locating and downloading the 54 NetCDF files required to perform the task from the ESGF. Instead, only three URLs will be used. These URLs can be easily obtained from the ESGF-VA. The three URLs correspond to the ESGF-VA endpoints of the CanESM5 multi-member data sources of the historical, SSP1-2.6 and SSP5-8.5 of the CMIP and ScenarioMIP ESGF activities respectively. Moreover, OPeNDAP automatically performs spatial and temporal subsetting on behalf of the user, regardless of how the NetCDF files are split along the time coordinate in the ESGF.

The data analysis task involves calculating model agreement on precipitation anomalies by computing the difference between the climatologies of both future scenarios relative to the historical period. The climatologies for each scenario have been computed as the temporal and model ensemble member mean of 18 model runs, given that this information is available out-of-the-box in the ARD dataset from the ESGF-VA. The years 1995 to 2014 are chosen as the reference for the historical period, and the years 2080 to 2100 represent the future period. Figures 8.3 and 8.4 show the relative precipitation deltas of all model members for scenarios SSP1-2.6 and SSP5-8.5 respectively.

Model member agreement will be computed following the *low model agreement simple approach* methodology proposed in the Intergovernmental Panel on Climate Change (IPCC) Sixth Assesment Report [26]. This methodology aims to display robustness and uncertainty in maps of multi-model mean changes. Model agreement is computed using *model member democracy* without discarding/weighting model members. The low locations of model member agreement, those with $< 80\%$ agreeing on the sign of change, are marked using diagonal hatched lines [26]. The results for both future scenarios are shown in Figure 8.5.

8.3 Reproducibility from the AR6

The next step is to reproduce figures from the IPCC's AR6 report using the ESGF-VA Data Laboratory. In doing so, the feasibility of using ESGF to produce relevant climate results directly within its infrastructure is evaluated, while providing an ARD interface

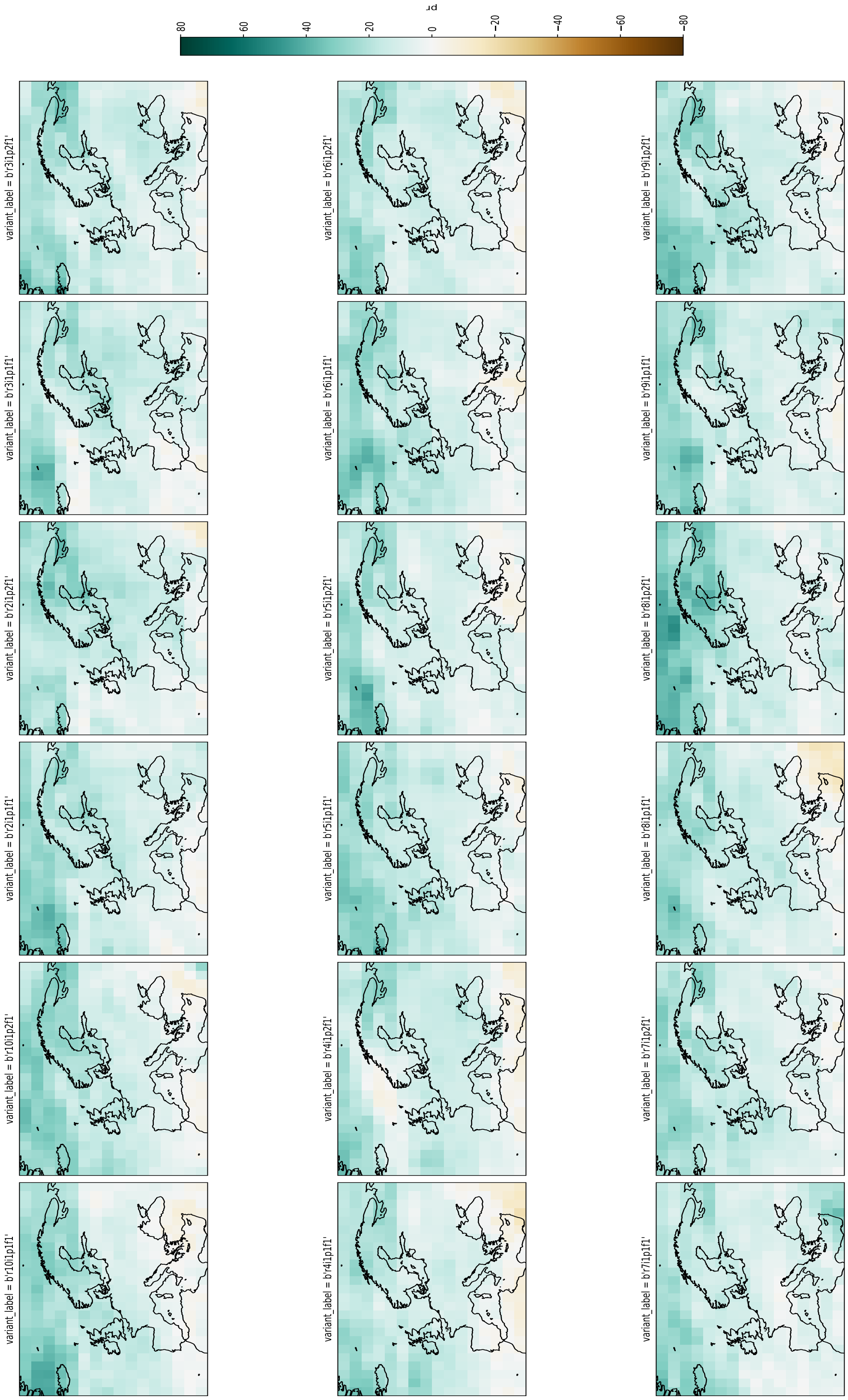

pr
80
60
40
20
0
−20
−40
−60
−80
variant_label = b'r3i1p2f1'
variant_label = b'r3i1p1f1'
variant_label = b'r2i1p2f1'
variant_label = b'r2i1p1f1'
variant_label = b'r10i1p2f1'
variant_label = b'r10i1p1f1'
variant_label = b'r6i1p2f1'
variant_label = b'r6i1p1f1'
variant_label = b'r5i1p2f1'
variant_label = b'r5i1p1f1'
variant_label = b'r4i1p2f1'
variant_label = b'r4i1p1f1'
variant_label = b'r9i1p2f1'
variant_label = b'r9i1p1f1'
variant_label = b'r8i1p2f1'
variant_label = b'r8i1p1f1'
variant_label = b'r7i1p2f1'
variant_label = b'r7i1p1f1'

FIGURE 8.3: Relative precipitation changes for 18 model members for the SSP1-2.6 future scenario from CanESM5. These changes are calculated as the difference between the projected future scenario period and the historical reference.

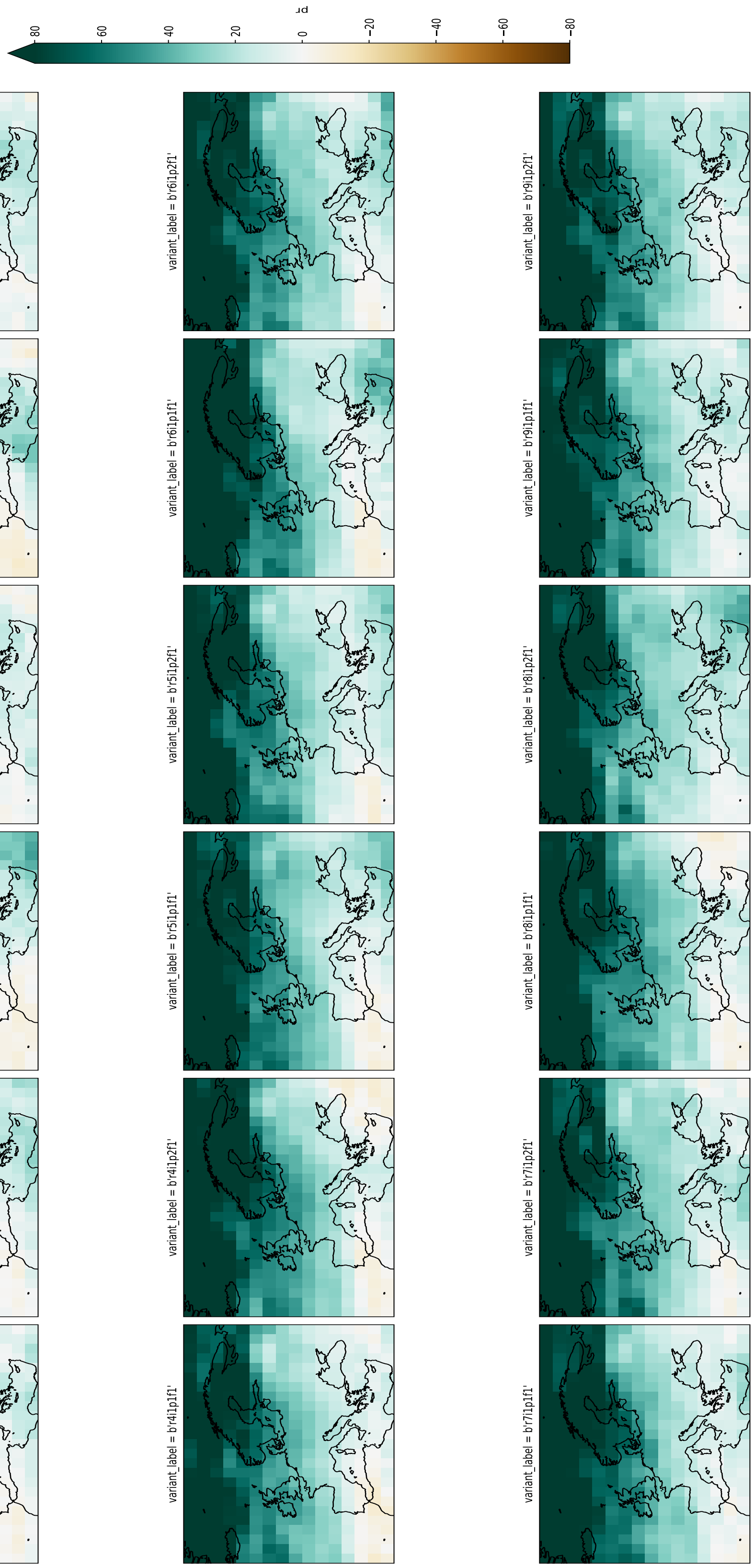

pr
80
60
40
20
0
−20
−40
−60
−80
variant_label = b'r3i1p2f1'
variant_label = b'r3i1p1f1'
variant_label = b'r2i1p2f1'
variant_label = b'r2i1p1f1'
variant_label = b'r10i1p2f1'
variant_label = b'r10i1p1f1'
variant_label = b'r6i1p2f1'
variant_label = b'r6i1p1f1'
variant_label = b'r5i1p2f1'
variant_label = b'r5i1p1f1'
variant_label = b'r4i1p2f1'
variant_label = b'r4i1p1f1'
variant_label = b'r9i1p2f1'
variant_label = b'r9i1p1f1'
variant_label = b'r8i1p2f1'
variant_label = b'r8i1p1f1'
variant_label = b'r7i1p2f1'
variant_label = b'r7i1p1f1'

FIGURE 8.4: Relative precipitation changes for 18 model members for the SSP5-8.5 future scenario from CanESM5. These changes are calculated as the difference between the projected future scenario period and the historical reference.

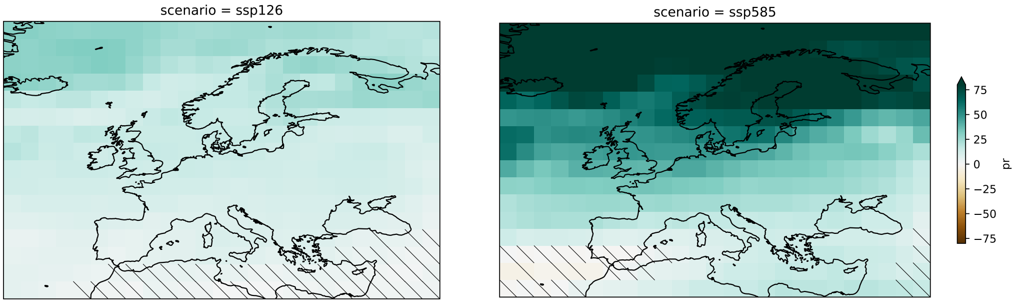


FIGURE 8.5: Model member agreement on relative precipitation changes for 18 model members across two future scenarios from CanESM5. These changes are calculated as the difference between the projected future scenario period and the historical reference. The results highlight significant projected increases in precipitation over northern Europe by the end of the century, under the fossil-fueled development scenario. Diagonal lines indicate areas in southern Europe where model member agreement is low. **Source:**Cimadevilla et al., 2025.

to the user. This approach eliminates the need for users to download or manually handle NetCDF files when performing climate data analysis tasks. Moreover, this approach would also avoid the necessity for intermediate infrastructures that replicate custom datasets extracted from the ESGF 6.

It is shown that some reproducibility is feasible from the ESGF-VA Data Laboratory, although ESGF limitations restrict the scalability of reproducibility to specific use cases referred to as *data-lightweight* workflows. These workflows take advantage of subsetting and consist in climate data analysis tasks that take place on a temporal or spatial subset of the full dataset. In contrast, *data-intensive* workflows require most, if not all of the data, to travel from the ESGF servers to the client. Due to the operational limitations of the OPeNDAP service within the ESGF, the ESGF-VA Data Laboratory is better suited to data-lightweight workflows. In this context, a justification is presented for deploying analysis-ready, cloud-optimized repositories to enhance climate data analysis, particularly for data-intensive workflows.

8.3.1 Data-Lightweight Workflows

Data-lightweight workflows are defined as those that employ subsetting to reduce the volume of data transferred over the network, compared to the amount required in a traditional file-download approach. By leveraging the ESGF-VA, users of the ESGF-VA Data Laboratory interact not with individual NetCDF files, but with virtual aggregations that abstract away the underlying files and present applications with an aggregated view of the data. The benefits of remote data access through ESGF-VA become evident when one considers that subsetting across multiple files is handled automatically by the data access layer, transparently to the user.

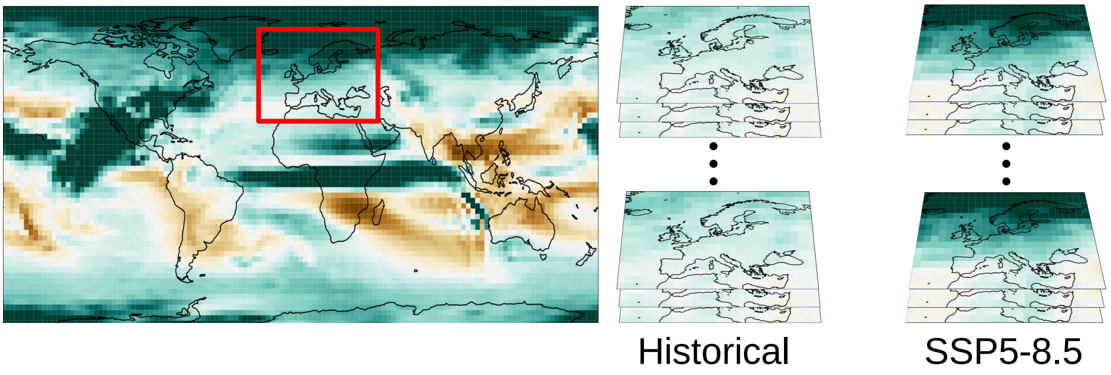


FIGURE 8.6: Illustration of the spatial subsetting requirements for performing the climate data analysis task on model agreement. The original datasets cover the entire globe, but the analysis focuses on a relatively small area - the region of Europe. File downloads would require transferring the entire spatial domain, whereas remote data access, with implicit subsetting capabilities, allows users to retrieve only the region of interest. This makes the analysis task significantly more efficient.

As an example of a data-lightweight workflow, the model agreement result of the Interactive Atlas DataLab for Europe is reproduced, which depicts projected precipitation changes under a +3ºC warming scenario. In this analysis, locations with low model agreement are displayed using diagonal lines. This constitutes a data-lightweight workflow due to the intrinsic subsetting required for the analysis. Although the original dataset covers the entire globe, the analysis only requires data for the European region. Figure 8.6 illustrates the original spatial extent of the dataset and the smaller region necessary for the analysis.

To carry out the workflow, the list of datasets with their exact versions was obtained from the corresponding data source inventory. Reference grids and code are available in the IPCC-WGI/Atlas repository (`https://github.com/IPCC-WG1/Atlas`; [27]). Unfortunately, datasets in the ESGF are subject to change over time, and the exact versions used during the development of the IPCC Interactive Atlas DataLab are no longer available. Therefore, the workflow is restricted to datasets whose exact versions remain available in the ESGF. This reduces the model ensemble from the original 33 members to 15 members that match the required versions. Figure 8.7 shows the results of the model agreement and GWL for the 15 member ensemble.

In a file downloads approach, the entire spatial domain must be retrieved and stored locally, resulting in unnecessary time and storage consumption. By leveraging remote data access, only the required spatial subset is transferred over the network, offering a significantly more efficient method of data access. In this particular case, the file-download approach requires transferring 739 NetCDF files totaling `238 GB`, whereas remote data access only involves approximately `200 MB` of data transfer.

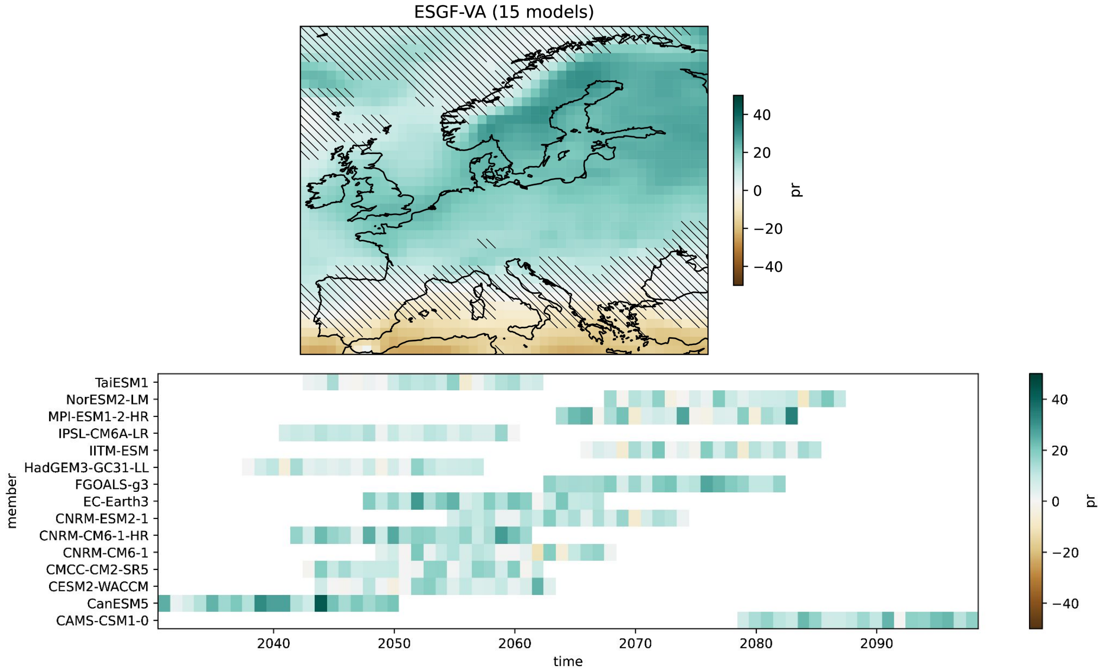


FIGURE 8.7: Top: Reproduction of the Interactive Atlas DataLab output for Europe, illustrating projected precipitation changes under a +3 °C global warming scenario for the 15-model ensemble considered. Areas with low model agreement are indicated by diagonal hatching [26]. Bottom: Visualization of the 20-year periods during which each model in the ensemble (rows) reaches the +3 °C global warming threshold under the SSP5-8.5 scenario. Colors represent the relative precipitation change compared to the reference period for each year within those 20-year windows.

8.3.2 Data-Intensive Workflows

This work highlights the benefits of using Analysis-Ready Cloud-Optimized repositories [44] for remote data access with chunk compression. As argued in Section 4.2.2 and demonstrated in Chapter 7, remote data access based on NetCDF combined with Kerchunk (a Zarr interface to existing NetCDF files) offers superior performance, as chunks are transferred without decompression—unlike with OPeNDAP. This advantage is further amplified by the fact that cloud storage providers operate object stores in an enterprise-grade manner, in contrast to ESGF data nodes, which function on a *best-effort* basis. As a result, ESGF data nodes are frequently inoperative for extended periods, casting doubt on the practical viability of remote data access initiatives that rely on them.

To assess the limitations of the ESGF-VA Data Laboratory for data-intensive workflows, the GSAT time series evolution from the Interactive Atlas DataLab is reproduced. This constitutes a data-intensive workflow because subsetting cannot be leveraged to perform the climate data analysis task; the entire spatial and temporal domain of the datasets must be processed to produce the final plot. It is demonstrated that the volume of data required for this workflow amounts to approximately half a terabyte, even for a reduced ensemble of models, representing a significant amount of data to be transferred over the network.

In performing this workflow, several considerations must be taken into account. The GSAT plot produced for the Interactive Atlas DataLab represents a weighted mean of surface temperature, which accounts for the reduced spatial area represented by grid cells at higher latitudes. These cells contribute less to the global mean due to their smaller real-world surface area. When computing the weighted mean, the graph of operations executed by the computation runtime (in this case, `xarray` and `Dask`) changes significantly. As a result, measurements of I/O performance must be interpreted with caution, since blocking tasks can easily occur, leading to misleading assessments of I/O throughput.

Figure 8.8 presents the results of computing both the weighted and unweighted GSAT time series, along with the corresponding Dask computation graphs for each task. It is evident that the Dask graph for the weighted mean GSAT is significantly more complex compared to its unweighted counterpart, reflecting the additional computational overhead introduced by the weighting process. The weighted mean is computed by the formula

$$\bar{x}_w = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$$

Workflows to compute weighted and unweighted GSAT time series were carried out using data from the ESGF-VA and the ARCO dataset provided by the Pangeo initiative. Each run of this workflow involves transferring a substantial amount of data from server nodes—either ESGF data nodes or Google Cloud infrastructure. As with the data-lightweight workflow, the workflow is restricted to datasets whose exact versions remain available on the ESGF. Table 8.1 presents the list of available datasets per scenario.

For a single run, approximately half a terabyte of data is transferred when reading from ESGF, compared to about 250 gigabytes when reading from Google Cloud Storage. This discrepancy arises from differences in the data access protocols. ESGF uses OPeNDAP, which decompresses data chunks before transmitting them over the network. In contrast, data from Google Cloud Storage is accessed using Zarr, which retrieves compressed chunks and decompresses them on the client side. As a result, Zarr benefits from the compression of data sources, significantly reducing the volume of data that needs to be transferred from the servers. Five different runs of each workflow have been carried to obtain multiple samples of the workflow. Figure 8.9 shows the performance of the Pangeo and ESGF infrastructures in computing the weighted and unweighted mean surface temperature using different number of parallel worker processes.

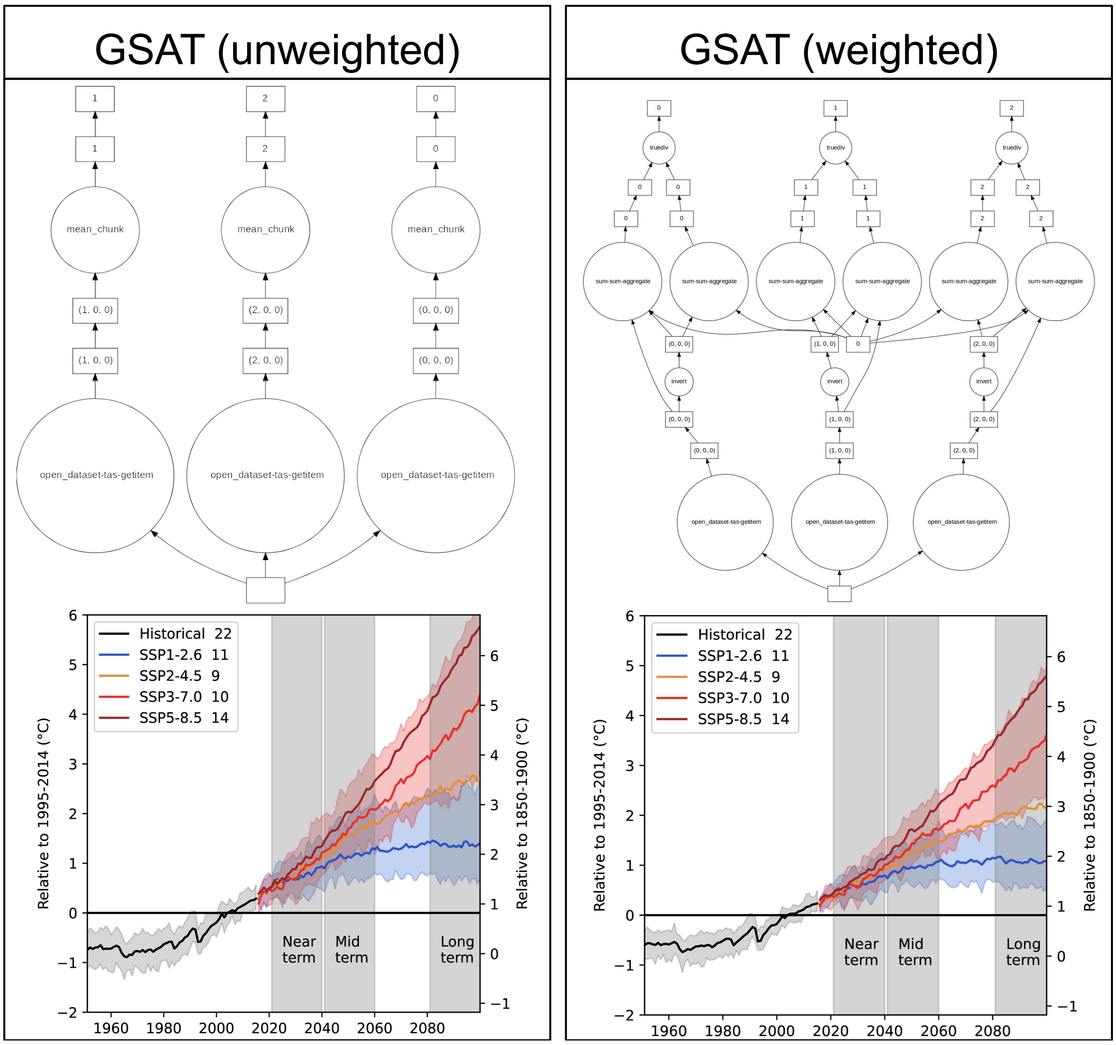


FIGURE 8.8: Evolution of GSAT according to a subset of the model ensemble available from the IPCC Interactive Atlas DataLab. The results are computed using data from the ESGF-VA, considering only models whose versions exactly match those used in the Interactive Atlas. The Dask computation graph for each data analysis workflow is shown above each plot. It is evident that the weighted mean results in a more complex Dask graph compared to the simpler structure of the unweighted mean operation.

8.4 Conclusions

The ESGF-VA Data Laboratory has been presented as an environment that enables climate researchers to perform data analysis while avoiding common obstacles, such as downloading files and managing them within a traditional file system, instead of working directly with analysis-ready data. The ESGF-VA takes advantage of the ESGF Vitual Aggregation methodology to offer virtual analysis-ready data, which means that the analysis-ready data sources don't consume additional storage relative to the original data sources. Moreover,

Using ESGF remote data access technology, the limitations of the ESGF-VA Data

Model	Run	historical	ssp585	ssp126	ssp245	ssp370
AWI-CM-1-1-MR	r1i1p1f1	🟩	🟩	🟩	🟩	🟩
CAMS-CSM1-0	r2i1p1f1	🟩	🟩	🟩	🟩	🟩
CanESM5	r1i1p1f1	🟩	🟩	🟥	🟥	🟥
CESM2-WACCM	r1i1p1f1	🟩	🟩	🟥	🟥	🟥
CMCC-CM2-SR5	r1i1p1f1	🟩	🟩	🟩	🟩	🟩
CNRM-CM6-1	r1i1p1f2	🟩	🟩	🟩	🟩	🟩
CNRM-CM6-1-HR	r1i1p1f2	🟩	🟩	🟥	🟥	🟥
CNRM-ESM2-1	r1i1p1f2	🟩	🟩	🟩	🟥	🟩
GFDL-CM4	r1i1p1f1	🟩	🟥	🟥	🟥	🟥
GFDL-ESM4	r1i1p1f1	🟩	🟥	🟩	🟥	🟥
HadGEM3-GC31-LL	r1i1p1f3	🟩	🟩	🟥	🟥	🟥
IITM-ESM	r1i1p1f1	🟩	🟩	🟩	🟩	🟩
INM-CM4-8	r1i1p1f1	🟩	🟥	🟥	🟥	🟥
INM-CM5-0	r1i1p1f1	🟩	🟥	🟥	🟥	🟥
IPSL-CM6A-LR	r1i1p1f1	🟩	🟩	🟩	🟩	🟩
MPI-ESM1-2-HR	r1i1p1f1	🟩	🟩	🟩	🟩	🟩
MPI-ESM1-2-LR	r1i1p1f1	🟩	🟥	🟩	🟥	🟥
MRI-ESM2-0	r1i1p1f1	🟩	🟥	🟥	🟩	🟩
NorESM2-LM	r1i1p1f1	🟩	🟩	🟥	🟥	🟥
NorESM2-MM	r1i1p1f1	🟩	🟥	🟥	🟥	🟥
TaiESM1	r1i1p1f1	🟩	🟩	🟥	🟥	🟥
UKESM1-0-LL	r1i1p1f2	🟩	🟥	🟩	🟩	🟩

Table 8.1: This table shows the availability of data in ESGF-VA that exactly matches the original NetCDF datasets from ESGF, ensuring reproducibility of the GSAT analysis.

Laboratory have been explored, recognizing that the ESGF operates in a *best effort* basis, unlike the commercial grade reliability of cloud service providers. The suitability of the ESGF-VA Data Laboratory for executing moderately complex tasks has been assessed. While it proves inadequate for data-intensive workflows, defined as those requiring hundreds of gigabytes of data, it remains effective for many climate analysis tasks. This effectiveness is largely due to the subsetting capabilities of remote data access, combined with the virtual, analysis-ready data sources provided by ESGF-VA, which enable simple and efficient analyses.

Remote data access has been provided by the ESGF-VA Data Laboratory through OPeNDAP, although other technologies also offer remote data access capabilities. In Chapter 7, Kerchunk was considered as an alternative to OPeNDAP; however, its use was not adopted because it requires inspecting the internals of NetCDF files, which is not currently feasible. In contrast, the ESGF-VA relies exclusively on information available in the ESGF index nodes, which can be queried at the scale of the entire CMIP6 archive. Nevertheless, the adoption of Kerchunk by modeling centers could, in the future, provide more efficient alternatives for remote data access within the federation.

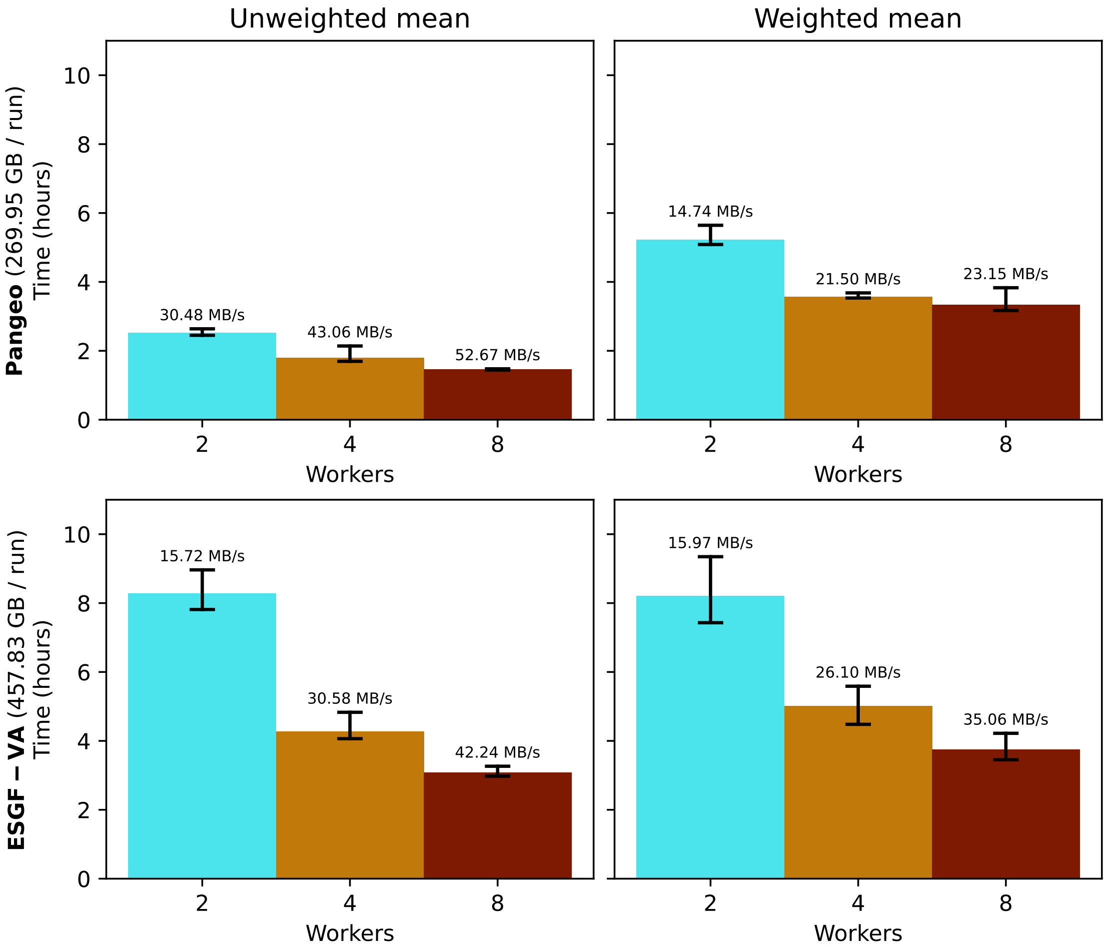


FIGURE 8.9: Results of the experimental retrieval of data required to compute GSAT, as shown in Figure 8.8. Where vertical bars are shown, they represent the minimum and maximum time taken from the available samples. Due to blocking tasks during the computation of the weighted mean, this operation cannot saturate bandwidth, as can be observed in comparison with the unweighted operation. Additionally, because of chunk compression, data retrieval from Pangeo requires nearly half the volume compared to ESGF.

The ESGF-VA Data Laboratory illustrates the characteristics of the ESGF infrastructure in terms of its technological dependencies, scalability, and continuous data availability. Its methodology relied on OPeNDAP because the information required to generate virtual aggregations was available in the ESGF index nodes, thereby solving the challenge of producing virtual aggregations without direct access to the data nodes. However, this methodology is technology-independent and can be applied to any remote data access system, provided that the necessary information for generating virtual aggregation granules is available in the index nodes.

The scalability of the ESGF-VA Data Laboratory is inherently limited by the scalability of the ESGF itself. In this regard, cloud repositories have demonstrated greater reliability and availability, albeit at the expense of data duplication. Such repositories mitigate the frequent downtimes observed in ESGF data node services. However, the sustainability of this approach remains uncertain and will need to be assessed in future phases of model intercomparison projects such as CMIP. Finally, reliable data availability within ESGF continues to be a major challenge. At present, model data may disappear without notification, which severely impacts reproducibility. These shortcomings are inherited by services built on top of ESGF, including the ESGF-VA Data Laboratory. In this sense, policies requiring modeling teams to commit to guaranteeing data access should be enforced at both data nodes and index nodes to ensure the reproducibility of products and services developed on top of ESGF.

The ESGF-VA Data Laboratory is the culminating contribution of this thesis. It advances the state of the art in climate data analysis for climate change assessments, moving beyond the traditional reliance on file downloads toward new infrastructures that support more advanced analysis techniques, primarily through the use of ARD and remote data access. Performance evaluations have been conducted to assess the limitations and strengths of the proposed technological solutions. The next chapter presents the main conclusions of this work and outlines potential future directions for climate data storage, analysis, and infrastructures.

Part III

Concluding Remarks

Chapter 9

Conclusions and Future Work

9.1 Main Conclusions

This thesis has examined the convergence of climate data science and the technological challenge of Big Data it faces, with a particular emphasis on how modern data infrastructures, tools, and technologies can enhance the storage, analysis, accessibility, and reproducibility of climate data. The research highlights the transformative potential of data-centric approaches in addressing the growing complexity and scale of climate datasets. This thesis has provided a comprehensive overview of the current state of climate data management in three key dimensions: climate data storage, climate data analysis, and climate data infrastructures. Thus, the first objective of the thesis has been met with the insiguits provided in these chapters. The existing technological landscape is predominantly centered on file downloads, a model that presents several limitations. These limitations hinder the potential of climate data science by introducing unnecessary complexity into analysis workflows. As a result, climate data analysts and researchers are burdened with managing these challenges, which in turn restricts the capabilities that climate data infrastructures can offer.

To address these issues, this work has proposed a forward-looking approach by introducing new concepts and identifying existing limitations. Central to this is the concept of the *climate data laboratory* 5.4. A data laboratory was interpreted as an environment equipped with the tools and infrastructure necessary to enable efficient data science. Drawing inspiration from experimental sciences, where laboratories are equipped with specialized instruments such as microscopes, spectrophotometers, and centrifuges, the data laboratory has been conceptualized as its digital counterpart. In this context, the instruments are more abstract in relation to their experimental counterparts. Data science laboratories include both hardware infrastructure and software tools that support various aspects of climate data management, such as storage formats, libraries, and analysis applications.

As a contribution of this thesis, Chapter 6 introduced the Interactive Atlas DataLab. The DataLab strikes a balance between advancing FAIR data principles and managing the practical costs of supporting the required infrastructure. It serves as both a testbed and a demonstration of technologies for climate data sharing, emphasizing their applicability in public climate services and the international exchange of climate information. By applying FAIR practices to the AR6 Interactive Atlas, the DataLab evaluates performance in relevant use cases, providing concrete evidence of how these principles can enhance reproducibility and accessibility. This contribution addresses the second objective of the thesis: enhancing climate data infrastructures beyond mere storage systems and Climate Information Products.

This thesis has employed the concept of *Analysis Ready Data* (ARD) as a means of providing users with a higher-level representation of climate data, thereby reducing the complexity of scripts that would otherwise need to handle low-level tasks such as NetCDF file location and file system management. Through logical aggregations, users can interact with data in the form of ARD, performing logical and statistical operations directly on the dimensions of these data cubes rather than on individual files in a file system. The specific contributions to ARD based on remote data access are described in Chapter 7, where the methodology for generating ARD for the entire CMIP6 database within the ESGF was presented.

Moreover, three modes of climate data analysis have been identified—download and analyze, remote data access, and next-to-data computing—each with its own advantages and disadvantages for different types of workflows. These contributions address the third objective of the thesis: enhancing climate data analysis. In particular, they demonstrate how the integration of virtual aggregations, remote data access, and analysis-ready data can substantially reduce the technical barriers typically faced by researchers working with climate datasets. All these techniques and technologies were combined in a unified environment, described in Chapter 8, where a data laboratory with access to virtual ARD and remote data services was presented. Furthermore, examples of climate data access that support climate change assessment, as presented in the IPCC AR6, were reproduced, accompanied by a dedicated performance study to objectively evaluate their viability in real-world production settings.

In summary, the concepts of the data laboratory and analysis-ready data within the ESGF framework have been integrated into a single environment: the *ESGF-VA Data Laboratory*. This laboratory represents the next evolutionary step in climate data infrastructures, moving beyond traditional systems focused on file downloads toward enhanced services such as remote data access and server-side computing. These capabilities reduce structural and technical burdens, allowing researchers to concentrate on the core aspects of climate data analysis. The ESGF-VA Data Laboratory extends the Interactive Atlas DataLab by strengthening the reproducibility and reusability of climate information products. It does so by enabling their generation directly through the infrastructures of ESGF and Pangeo, without the need to maintain parallel systems—a requirement that would otherwise impose significant costs. Finally, an objective performance evaluation has been conducted, enabling comparisons between the different technologies presented and fulfilling the fourth objective of the thesis.

All together, the contributions of this thesis have adhered to FAIR principles – ensuring that climate data are findable, accessible, interoperable, and reusable – which maximizes their usability across different research domains and facilitates reproducibility of analyses. The data laboratories that have been presented play a crucial role in advancing reproducibility and supporting the FAIR principles. By providing a controlled and interactive environment where datasets, analytical workflows, and computational resources are clearly documented, versioned, and consistently managed, data laboratories enable researchers to reliably replicate analyses and build upon previous work. In particular, such infrastructures are well-suited for large-scale assessments, such as those conducted by the IPCC, where multiple researchers must coordinate analyses, share results, and produce consistent and verifiable findings. By integrating reproducibility and FAIR-compliant practices at the core of data laboratories, climate research can achieve higher transparency, reliability, and long-term usability.

9.2 Future Work

There exists a current transition of climate data infrastructures to cloud-based environments. This shift introduces new capabilities, such as scalability, accessibility, and collaborative potential, while also presenting challenges that must be addressed. The analysis of this thesis offers a detailed exploration of both the opportunities and obstacles associated with this transition, laying the groundwork for future advancements in climate data science. Overall, this work underscores the importance of integrating computational and climate science expertise to develop interoperable, efficient, and user-friendly data systems. Such integration is essential for advancing global climate research and supporting informed decision-making in the face of climate change.

9.2.1 The Next Generation of the ESGF

The primary infrastructure for climate data in this work has been the Earth System Grid Federation (ESGF). Over the years, ESGF has proven to be a reliable platform for distributing climate data, supporting the storage and dissemination of climate model outputs used in the Intergovernmental Panel on Climate Change (IPCC) assessment reports. Therefore, ensuring its sustainability is a critical aspect of its continued evolution. Research centers may face challenges in maintaining the necessary infrastructure to participate as nodes within the federation. However, ESGF's support for data replication provides a mechanism to mitigate such failures.

ESGF continues to undergo significant technological evolution. At the time of writing, ESGF is in the midst of transitioning to ESGF2 - the so-called *next generation* of the federation - to continue providing access to climate model outputs. Unfortunately, this transition has led to a divergence in development efforts between the United States and Europe, informally referred to as *ESGF West* and *ESGF East*, respectively. This raises concerns about the extent of technological divergence between the two systems. Although the primary interface of ESGF remains a file distribution system, critical components such as data publication and user authentication must be carefully harmonized to ensure that the federation continues to deliver the high quality of service it has maintained over the years.

The main contributions of this thesis are based on the current ESGF architecture and will likely not be extended to ESGF2, as the architectural decisions underway suggest the discontinuation of existing remote data access services. Among the primary changes anticipated for the federation, remote data access via protocols such as OPeNDAP appears set to be phased out. The prevailing consensus is that ESGF will evolve into an infrastructure focused exclusively on NetCDF file distribution, while external initiatives such as Pangeo will assume the role of enhancing data accessibility across diverse platforms, including commercial cloud services. However, given the commercial nature of these infrastructures and the uncertainty regarding long-term commitments to guarantee open access, exclusive reliance on cloud providers seems difficult to sustain. Consequently, ESGF is expected to remain the principal archive for climate data.

It is clear that the ESGF has evolved and will continue to do so. Our work suggests that this ongoing evolution should address not only indexing and data download but also, where possible, the provision of direct data access suitable for a wide range of use cases. Such support could include providing modelling centres with clear guidance on how to chunk and organize their data, going beyond reliance on CMOR alone, as not all centres adopt it. Although this thesis has focused on OPeNDAP, NcML, and Kerchunk, future work should include the evaluation and assessment of other lightweight data servers and metadata file formats. These efforts will support more informed decision-making in enabling both remote data access and the generation of ARD. Examples include but are not limited to xpublish (https://github.com/xpublish-community/xpublish) and DMR++ (https://opendap.github.io/DMRpp-wiki/DMRpp.html).

This work has studied how to improve climate data access within the ESGF and to evaluate the feasibility of positioning the federation as an infrastructure that supports more than simple file downloads. Our findings suggest that these improvements are indeed realistic, although the operational administration and limited resources of ESGF nodes may constrain their ability to support the costs and requirements of such deployments. Consequently, alternative and more cost-effective mechanisms for climate data access, based on remote data access or server-side computing, may emerge in the near future.

9.2.2 Sparse Climate Data

This thesis has emphasized the need for the storage and analysis of dense climate data. Dense data has been contrasted with sparse data, as each poses different requirements for storage and analysis. However, climate data can also be sparse, as in the case of data produced by observational systems such as meteorological stations, or from trajectories such as those of planes and ships. The distribution and management of this type of data currently follow the same download-and-analyze workflow pattern commonly used in climate data management.

The concepts regarding storage, analysis, and infrastructure discussed in this work can also be applied to sparse meteorological data sources, although the specific storage requirements of sparse data must be carefully considered, as they may not be well supported by current climate data storage technologies based on multidimensional arrays. Building on the contributions of this thesis, the logical and statistical manipulation of sparse climate data can also be greatly simplified by overcoming the current challenges of their analysis, which closely resemble those associated with dense climate data.

9.2.3 Database Research

Current practices in model intercomparison projects (e.g., CMIP, CORDEX) involve publishing millions of NetCDF files to a federated infrastructure such as the ESGF. To analyze these datasets, users typically download the files to local workstations or high-performance computing (HPC) environments. At this stage, climate data users — often working in Python or R — begin their analysis. The first challenge is to locate and organize the relevant NetCDF files within a large and often inconsistently structured file system. This task is particularly daunting for scientists who may not have extensive programming experience. Ideally, the NetCDF files conform to community standards such as CF conventions and ESGF publication guidelines. In practice, however, this is frequently not the case. As a result, scripts become cluttered with numerous conditional statements to handle inconsistencies.

These conditions address issues such as incompatible units, inconsistent dimension names, and missing or non-standard metadata. Even basic assumptions — such as shared time coordinates across variables from the same model — often do not hold, requiring additional logic to reconcile discrepancies. The complexity of these preprocessing steps increases significantly as more models and variables are included in the analysis. Ultimately, constructing a script to compare just two time series becomes a substantial coding effort. In contrast, if a database management system (DBMS) handled these preprocessing tasks, the analysis could be expressed in a high-level query language [30].

It is important to clarify that the goal of ARD is not simply to merge files into a single *superset*, as sometimes described when generating NcMLs or Kerchunk aggregations. Rather, the objective is to enable logically valid inferences from data whose correctness is enforced by the system itself. In database terminology, what may appear as a superset corresponds instead to a persistent view, an inference derived from the underlying data that can be queried as if it were native. In the climate data community, this concept aligns with Analysis Ready Data (ARD). NcML files, or alternatives such as Kerchunk, can be seen as examples of ARD, but their correctness is not inherently guaranteed without a formal data model. By contrast, within the database theory, the logical soundness of such persistent views is assured [30].

The shortcomings of current climate data management systems largely arise from a widespread lack of understanding of core database principles in the climate data community. Addressing this issue requires a multi-step approach: users must first recognize where these deficiencies lie, understand why they matter, learn how to work around them, and ultimately advocate for improvements from storage technology developers. However, many users themselves lack familiarity with these foundational concepts, limiting their ability to contribute effectively to a collaborative solution. Moving toward a resilient, database-driven framework will require the integration of five decades of database research [30].

9.3 Publications and Contributions

The research carried out in this thesis has led to several scholarly contributions, including the publication of three peer-reviewed journal articles. The main contributions of this work are the following.

1. Cimadevilla E, Iturbide M, Cofiño AS, Fernández J, Sitz LE, Palacio A, et al. (2025). *The IPCC Interactive Atlas DataLab: Online reusability for regional climate change assessment.* PLOS Clim 4(6): e0000644. `https://doi.org/10.1371/journal.pclm.0000644`

   This contribution introduced the IPCC Interactive Atlas DataLab as an innovative step toward enhancing the reproducibility, transparency, and usability of climate change data for regional assessment. Beyond technical advances, the DataLab demonstrates clear applicability for informing national adaptation strategies, supporting scientific collaboration, and strengthening the link between climate research and policy, thereby laying groundwork for more dynamic and interactive approaches in future IPCC assessments.

2. Cimadevilla, E., Lawrence, B.N., & Cofiño, A.S. (2025). *The Earth System Grid Federation (ESGF) Virtual Aggregation (CMIP6 v20240125).* Geoscientific Model Development, 18(8), 2461–2478. `https://doi.org/10.5194/gmd-18-2461-2025`

   This contribution introduced the Earth System Grid Federation Virtual Aggregation (ESGF-VA) as a sustainable approach to providing analysis-ready climate data within the existing ESGF infrastructure. The originality of this contribution resides in that it offers a sustainable solution by creating virtual datasets with minimal storage costs, improving efficiency in handling the growing volume of climate model outputs while also highlighting best practices needed for data organization and accessibility.

3. Cimadevilla, E. (2025). *Why the relational data model matters for climate data management.* Computers & Geosciences, 201, 105931. `https://doi.org/10.1016/j.cageo.2025.105931`

   This contribution argues that the climate science community has overlooked the formal principles of the Relational Data Model, relying instead on libraries such as NetCDF that lack rigor for logical inference. The work contributes a strong case for adopting relational approaches in climate data management. This article has contributed to the theoretical background presented in Part I, with its use limited to aspects directly relevant to the scope of the thesis.

In addition, the work done here has also contributed to other publications.

1. Hoz, A.P., et al. (2025). *DataLab as a Service: Distributed Computing Framework for Multi-Interactive Analysis Environments.* IEEE Access, 13, 22566–22577. `https://doi.org/10.1109/ACCESS.2025.3536637`

2. Iturbide, M., et al. (2022). *Implementation of FAIR principles in the IPCC: the WGI AR6 Atlas repository.* Scientific Data, 9(1), 629. `https://doi.org/10.1038/s41597-022-01739-y`

Other contributions include oral and poster presentations:

1. Cimadevilla, E. and Cofiño, A.S. (2022) *Storage growth mitigation through data analysis ready climate datasets using HDF5 Virtual Datasets*, 28 March. Available at: `https://doi.org/10.5194/egusphere-egu22-7151`.

2. Cimadevilla, E., Iturbide, M. and Cofiño, A.S. (2023) *Virtual aggregations to improve scientific ETL and data analysis for datasets from the Earth System Grid Federation*, 15 May. Available at: `https://doi.org/10.5194/egusphere-egu23-16117`.

3. Cimadevilla, E. (2024) *A Science Gateway for climate data analysis based on Virtual Analysis Ready Data.* Proceedings of the 16th International Workshop on Science Gateways, Toulouse: Zenodo, 30 September. Available at: `https://doi.org/10.5281/ZENODO.13863563`.

Bibliography

[1] Anthony J. G. Hey and Microsoft Research, editors. *The fourth paradigm: data intensive scientific discovery.* Microsoft Research, Redmond, Wash, 2. printing, version 1.1 edition, 2009. ISBN 978-0-9825442-0-4.

[2] Wolfgang Pietsch. Aspects of Theory-Ladenness in Data-Intensive Science. *Philosophy of Science*, 82(5):905–916, December 2015. ISSN 0031-8248, 1539-767X. doi: 10.1086/683328. URL `https://www.cambridge.org/core/product/identifier/S003182480000831X/type/journal_article`.

[3] Wolfgang Pietsch. The Causal Nature of Modeling with Big Data. *Philosophy & Technology*, 29(2):137–171, June 2016. ISSN 2210-5433, 2210-5441. doi: 10.1007/s13347-015-0202-2. URL `http://link.springer.com/10.1007/s13347-015-0202-2`.

[4] Luciano Floridi. Big Data and Their Epistemological Challenge. *Philosophy & Technology*, 25(4):435–437, December 2012. ISSN 2210-5433, 2210-5441. doi: 10.1007/s13347-012-0093-4. URL `http://link.springer.com/10.1007/s13347-012-0093-4`.

[5] Danah Boyd and Kate Crawford. CRITICAL QUESTIONS FOR BIG DATA: Provocations for a cultural, technological, and scholarly phenomenon. *Information, Communication & Society*, 15(5):662–679, June 2012. ISSN 1369-118X, 1468-4462. doi: 10.1080/1369118X.2012.678878. URL `http://www.tandfonline.com/doi/abs/10.1080/1369118X.2012.678878`.

[6] Massimo Pigliucci. The end of theory in science? *EMBO reports*, 10(6):534–534, June 2009. ISSN 1469-221X, 1469-3178. doi: 10.1038/embor.2009.111. URL `https://www.embopress.org/doi/10.1038/embor.2009.111`.

[7] Tony Hey, Dennis Gannon, and Jim Pinkelman. The Future of Data-Intensive Science. *Computer*, 45(5):81–82, May 2012. ISSN 0018-9162. doi: 10.1109/MC.2012.181. URL `http://ieeexplore.ieee.org/document/6197782/`.

[8] D. Chen, M. Rojas, B. Samset, K. Cobb, A. Diongue Niang, P. Edwards, S. Emori, S. Faria, E. Hawkins, P. Hope, P. Huybrechts, M. Meinshausen, S. Mustafa, G.-K. Plattner, and A.-M. Tréguier. Framing, Context, and Methods. In V. Masson-Delmotte, P. Zhai, A. Pirani, S. Connors, C. Péan, S. Berger, N. Caud, Y. Chen, L. Goldfarb, M. Gomis, M. Huang, K. Leitzell, E. Lonnoy, J. Matthews, T. Maycock, T. Waterfield, O. Yelekçi, R. Yu, and B. Zhou, editors, *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*, pages 147–286. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2021.

[9] Paul N. Edwards. History of climate modeling. *WIREs Climate Change*, 2(1): 128–139, January 2011. ISSN 1757-7780, 1757-7799. doi: 10.1002/wcc.95. URL `https://wires.onlinelibrary.wiley.com/doi/10.1002/wcc.95`.

[10] J. T. Overpeck, G. A. Meehl, S. Bony, and D. R. Easterling. Climate Data Challenges in the 21st Century. *Science*, 331(6018):700–702, February 2011. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1197869. URL `https://www.sciencemag.org/lookup/doi/10.1126/science.1197869`.

[11] J. Salazar Loor and P. Fdez-Arroyabe. Aerial and Satellite Imagery and Big Data: Blending Old Technologies with New Trends. In Nilanjan Dey, Chintan Bhatt, and Amira S. Ashour, editors, *Big Data for Remote Sensing: Visualization, Analysis and Interpretation*, pages 39–59. Springer International Publishing, Cham, 2019. ISBN 978-3-319-89922-0 978-3-319-89923-7. doi: 10.1007/978-3-319-89923-7_2. URL `http://link.springer.com/10.1007/978-3-319-89923-7_2`.

[12] Venkatramani Balaji, Karl E. Taylor, Martin Juckes, Bryan N. Lawrence, Paul J. Durack, Michael Lautenschlager, Chris Blanton, Luca Cinquini, Sébastien Denvil, Mark Elkington, Francesca Guglielmo, Eric Guilyardi, David Hassell, Slava Kharin, Stefan Kindermann, Sergey Nikonov, Aparna Radhakrishnan, Martina Stockhause, Tobias Weigel, and Dean Williams. Requirements for a global data infrastructure in support of CMIP6. *Geoscientific Model Development*, 11(9):3659–3680, September 2018. ISSN 1991-9603. doi: 10.5194/gmd-11-3659-2018. URL `https://www.geosci-model-dev.net/11/3659/2018/`.

[13] Martin Juckes, Karl E. Taylor, Paul J. Durack, Bryan Lawrence, Matthew S. Mizielinski, Alison Pamment, Jean-Yves Peterschmitt, Michel Rixen, and Stéphane Sénési. The CMIP6 Data Request (DREQ, version 01.00.31). *Geoscientific Model Development*, 13(1):201–224, January 2020. ISSN 1991-9603. doi: 10.5194/gmd-13-201-2020. URL `https://gmd.copernicus.org/articles/13/201/2020/`.

[14] Veronika Eyring, Sandrine Bony, Gerald A. Meehl, Catherine A. Senior, Bjorn Stevens, Ronald J. Stouffer, and Karl E. Taylor. Overview of the Coupled Model Intercomparison Project Phase 6 (CMIP6) experimental design and organization. *Geoscientific Model Development*, 9(5):1937–1958, May 2016. ISSN 1991-9603. doi: 10.5194/gmd-9-1937-2016. URL https://www.geosci-model-dev.net/9/1937/2016/.

[15] Filippo Giorgi and Linda O. Mearns. Approaches to the simulation of regional climate change: A review. *Reviews of Geophysics*, 29(2):191–216, May 1991. ISSN 8755-1209, 1944-9208. doi: 10.1029/90RG02636. URL https://agupubs.onlinelibrary.wiley.com/doi/10.1029/90RG02636.

[16] Daniela Jacob, Juliane Petersen, Bastian Eggert, Antoinette Alias, Ole Bøssing Christensen, Laurens M. Bouwer, Alain Braun, Augustin Colette, Michel Déqué, Goran Georgievski, Elena Georgopoulou, Andreas Gobiet, Laurent Menut, Grigory Nikulin, Andreas Haensler, Nils Hempelmann, Colin Jones, Klaus Keuler, Sari Kovats, Nico Kröner, Sven Kotlarski, Arne Kriegsmann, Eric Martin, Erik Van Meijgaard, Christopher Moseley, Susanne Pfeifer, Swantje Preuschmann, Christine Radermacher, Kai Radtke, Diana Rechid, Mark Rounsevell, Patrick Samuelsson, Samuel Somot, Jean-Francois Soussana, Claas Teichmann, Riccardo Valentini, Robert Vautard, Björn Weber, and Pascal Yiou. EURO-CORDEX: new high-resolution climate change projections for European impact research. *Regional Environmental Change*, 14(2):563–578, April 2014. ISSN 1436-3798, 1436-378X. doi: 10.1007/s10113-013-0499-2. URL http://link.springer.com/10.1007/s10113-013-0499-2.

[17] Rasmus E Benestad, Inger Hanssen-Bauer, and Deliang Chen. *Empirical-Statistical Downscaling*. WORLD SCIENTIFIC, September 2008. ISBN 978-981-281-912-3 978-981-281-914-7. doi: 10.1142/6908. URL https://www.worldscientific.com/worldscibooks/10.1142/6908.

[18] Sebastian Scher and Gabriele Messori. Weather and climate forecasting with neural networks: using general circulation models (GCMs) with different complexity as a study ground. *Geoscientific Model Development*, 12(7):2797–2809, July 2019. ISSN 1991-9603. doi: 10.5194/gmd-12-2797-2019. URL https://gmd.copernicus.org/articles/12/2797/2019/.

[19] Leonardo Olivetti and Gabriele Messori. Advances and prospects of deep learning for medium-range extreme weather forecasting. *Geoscientific Model Development*, 17(6):2347–2358, March 2024. ISSN 1991-9603. doi: 10.5194/gmd-17-2347-2024. URL https://gmd.copernicus.org/articles/17/2347/2024/.

[20] Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia. GraphCast: Learning skillful medium-range global weather forecasting, 2022. URL `https://arxiv.org/abs/2212.12794`. Version Number: 2.

[21] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, Pedram Hassanzadeh, Karthik Kashinath, and Animashree Anandkumar. FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators, 2022. URL `https://arxiv.org/abs/2202.11214`. Version Number: 1.

[22] Ad Hoc Study Group on Carbon Dioxide and Climate, Climate Research Board, Assembly of Mathematical and Physical Sciences, and National Research Council. *Carbon Dioxide and Climate: A Scientific Assessment.* National Academies Press, Washington, D.C., March 1979. ISBN 978-0-309-11910-8. doi: 10.17226/12181. URL `https://www.nap.edu/catalog/12181`. Pages: 12181.

[23] Intergovernmental Panel On Climate Change (Ipcc). *Climate Change 2021 – The Physical Science Basis: Working Group I Contribution to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change.* Cambridge University Press, 1 edition, July 2023. ISBN 978-1-009-15789-6. doi: 10.1017/9781009157896. URL `https://www.cambridge.org/core/product/identifier/9781009157896/type/book`.

[24] J. Birkmann, E. Liwenga, R. Pandey, E. Boyd, R. Djalante, F. Gemenne, W. Leal Filho, P.F. Pinho, L. Stringer, and D. Wrathall. Poverty, Livelihoods and Sustainable Development. In H. O. Pörtner, D. C. Roberts, M. Tignor, E. S. Poloczanska, K. Mintenbeck, A. Alegría, M. Craig, S. Langsdorf, S. Löschke, V. Möller, A. Okem, and B. Rama, editors, *Climate Change 2022: Impacts, Adaptation and Vulnerability. Contribution of Working Group II to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*, pages 1171–1274. Cambridge University Press, Cambridge, UK and New York, USA, 2022. ISBN 978-1-009-32584-4. doi: 10.1017/9781009325844.010.1171. Type: Book Section.

[25] R. Bezner Kerr, T. Hasegawa, R. Lasco, I. Bhatt, D. Deryng, A. Farrell, H. Gurney-Smith, H. Ju, S. Lluch-Cota, F. Meza, G. Nelson, H. Neufeldt, and P. Thornton. Food, Fibre, and Other Ecosystem Products. In H. O. Pörtner, D. C. Roberts, M. Tignor, E. S. Poloczanska, K. Mintenbeck, A. Alegría, M. Craig, S. Langsdorf,

S. Löschke, V. Möller, A. Okem, and B. Rama, editors, *Climate Change 2022: Impacts, Adaptation and Vulnerability. Contribution of Working Group II to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*, pages 713–906. Cambridge University Press, Cambridge, UK and New York, USA, 2022. ISBN 978-1-009-32584-4. doi: 10.1017/9781009325844.007.714. Type: Book Section.

[26] J.M. Gutiérrez, R.G. Jones, G.T. Narisma, L.M. Alves, M. Amjad, I.V. Gorodetskaya, M. Grose, N.A.B. Klutse, S. Krakovska, J. Li, D. Martínez-Castro, L.O. Mearns, S.H. Mernild, T. Ngo-Duc, B. van den Hurk, and J.-H. Yoon. Atlas. In V. Masson-Delmotte, P. Zhai, A. Pirani, S.L. Connors, C. Péan, S. Berger, N. Caud, Y. Chen, L. Goldfarb, M.I. Gomis, M. Huang, K. Leitzell, E. Lonnoy, J.B.R. Matthews, T.K. Maycock, T. Waterfield, O. Yelekçi, R. Yu, and B. Zhou, editors, *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*, pages 1927–2058. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2021. doi: 10.1017/9781009157896.021. Type: Book Section.

[27] Maialen Iturbide, Jesús Fernández, José M. Gutiérrez, Anna Pirani, David Huard, Alaa Al Khourdajie, Jorge Baño-Medina, Joaquin Bedia, Ana Casanueva, Ezequiel Cimadevilla, Antonio S. Cofiño, Matteo De Felice, Javier Diez-Sierra, Markel García-Díez, James Goldie, Dimitris A. Herrera, Sixto Herrera, Rodrigo Manzanas, Josipa Milovac, Aparna Radhakrishnan, Daniel San-Martín, Alessandro Spinuso, Kristen M. Thyng, Claire Trenham, and Ozge Yelekçi. Implementation of FAIR principles in the IPCC: the WGI AR6 Atlas repository. *Scientific Data*, 9(1): 629, October 2022. ISSN 2052-4463. doi: 10.1038/s41597-022-01739-y. URL `https://www.nature.com/articles/s41597-022-01739-y`.

[28] Meng Lu, Marius Appel, and Edzer Pebesma. Multidimensional Arrays for Analysing Geoscientific Data. *ISPRS International Journal of Geo-Information*, 7(8):313, August 2018. ISSN 2220-9964. doi: 10.3390/ijgi7080313. URL `http://www.mdpi.com/2220-9964/7/8/313`.

[29] Charles R. Harris, K. Jarrod Millman, Stéfan J. Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. Van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández Del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E.

Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-020-2649-2. URL `https://www.nature.com/articles/s41586-020-2649-2`.

[30] Ezequiel Cimadevilla. Why the relational data model matters for climate data management. *Computers & Geosciences*, 201:105931, July 2025. ISSN 00983004. doi: 10.1016/j.cageo.2025.105931. URL `https://linkinghub.elsevier.com/retrieve/pii/S0098300425000810`.

[31] Stefano Nativi, John Caron, Ben Domenico, and Lorenzo Bigagli. Unidata's Common Data Model mapping to the ISO 19123 Data Model. *Earth Science Informatics*, 1(2):59–78, September 2008. ISSN 1865-0473, 1865-0481. doi: 10.1007/s12145-008-0011-6. URL `https://link.springer.com/10.1007/s12145-008-0011-6`.

[32] Brian Eaton, Jonathan Gregory, Bob Drach, Karl Taylor, Steve Hankin, John Caron, Rich Signell, Phil Bentley, Greg Rappa, Heinke Höck, Alison Pamment, Martin Juckes, Martin Raspaud, Randy Horne, Timothy Whiteaker, David Blodgett, Charlie Zender, and Daniel Lee. NetCDF Climate and Forecast (CF) Metadata Conventions v1.9, 2024. URL `https://cfconventions.org/`.

[33] Russ Rew, Ed Hartnett, and John Caron. NetCDF-4: Software implementing an enhanced data model for the geosciences. In *86th AMS Annual Meeting*, 2006. URL `https://www.scopus.com/inward/record.uri?eid=2-s2.0-77949308478&partnerID=40&md5=0903d47dc1ecf01f2472095fa3191055`. Type: Conference paper.

[34] HDF. HDF5 Dimension Scale Specification and Design Notes, March 2005. URL `https://support.hdfgroup.org/HDF5/doc/HL/H5DS_Spec.pdf`.

[35] Sriniket Ambatipudi and Suren Byna. A Comparison of HDF5, Zarr, and netCDF4 in Performing Common I/O Operations, February 2023. URL `http://arxiv.org/abs/2207.09503`. arXiv:2207.09503 [cs].

[36] Florin Rusu. Multidimensional Array Data Management. *Foundations and Trends® in Databases*, 12(2-3):69–220, 2023. ISSN 1931-7883, 1931-7891. doi: 10.1561/1900000069. URL `http://www.nowpublishers.com/article/Details/DBS-069`.

[37] Peter Lindstrom. Fixed-Rate Compressed Floating-Point Arrays. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2674–2683, December 2014. ISSN 1077-2626, 1941-0506, 2160-9306. doi: 10.1109/TVCG.2014.2346458. URL `https://ieeexplore.ieee.org/document/6876024/`.

[38] Milan Klöwer, Miha Razinger, Juan J. Dominguez, Peter D. Düben, and Tim N. Palmer. Compressing atmospheric data into its real information content. *Nature*

Computational Science, 1(11):713–724, November 2021. ISSN 2662-8457. doi: 10.1038/s43588-021-00156-2. URL https://www.nature.com/articles/s43588-021-00156-2.

[39] Carlo Buontempo, Samantha N. Burgess, Dick Dee, Bernard Pinty, Jean-Noël Thépaut, Michel Rixen, Samuel Almond, David Armstrong, Anca Brookshaw, Angel Lopez Alos, Bill Bell, Cedric Bergeron, Chiara Cagnazzo, Edward Comyn-Platt, Eduardo Damasio-Da-Costa, Anabelle Guillory, Hans Hersbach, András Horányi, Julien Nicolas, Andre Obregon, Eduardo Penabad Ramos, Baudouin Raoult, Joaquín Muñoz-Sabater, Adrian Simmons, Cornel Soci, Martin Suttie, Freja Vamborg, James Varndell, Stijn Vermoote, Xiaobo Yang, and Juan Garcés De Marcilla. The Copernicus Climate Change Service: Climate Science in Action. *Bulletin of the American Meteorological Society*, 103(12):E2669–E2687, December 2022. ISSN 0003-0007, 1520-0477. doi: 10.1175/BAMS-D-21-0315.1. URL https://journals.ametsoc.org/view/journals/bams/103/12/BAMS-D-21-0315.1.xml.

[40] R. Bayer and E. M. McCreight. Organization and maintenance of large ordered indexes. *Acta Informatica*, 1(3):173–189, 1972. ISSN 0001-5903, 1432-0525. doi: 10.1007/BF00288683. URL http://link.springer.com/10.1007/BF00288683.

[41] Douglas Comer. Ubiquitous B-Tree. *ACM Computing Surveys*, 11(2):121–137, June 1979. ISSN 0360-0300, 1557-7341. doi: 10.1145/356770.356776. URL https://dl.acm.org/doi/10.1145/356770.356776.

[42] Philip L. Lehman and S. Bing Yao. Efficient locking for concurrent operations on B-trees. *ACM Transactions on Database Systems*, 6(4):650–670, December 1981. ISSN 0362-5915, 1557-4644. doi: 10.1145/319628.319663. URL https://dl.acm.org/doi/10.1145/319628.319663.

[43] Alistair Miles, Jakirkham, Matthias Bussonnier, Josh Moore, Andrew Fulton, James Bourbeau, Tarik Onalan, Joe Hamman, Zain Patel, Matthew Rocklin, Gregory R. Lee, Davis Bennett, Elliott Sales De Andrade, Ryan Abernathey, Martin Durant, Vincent Schut, Raphael Dussin, Chris Barnes, Ben Williams, Boaz Mohar, Charles Noyes, Shikharsg, Juan Nunez-Iglesias, Aleksandar Jelenak, Anderson Banihirwe, David Baddeley, Eric Younkin, George Sakkis, and Ian Hunt-Isaak. zarr-developers/zarr-python: v2.10.3, November 2021. URL https://zenodo.org/record/5712786.

[44] Ryan P. Abernathey, Tom Augspurger, Anderson Banihirwe, Charles C. Blackmon-Luca, Timothy J. Crone, Chelle L. Gentemann, Joseph J. Hamman, Naomi Henderson, Chiara Lepore, Theo A. McCaie, Niall H. Robinson, and Richard P. Signell.

Cloud-Native Repositories for Big Scientific Data. *Computing in Science & Engineering*, 23(2):26–35, March 2021. ISSN 1521-9615, 1558-366X. doi: 10.1109/MCSE.2021.3059437. URL `https://ieeexplore.ieee.org/document/9354557/`.

[45] Mohan Ramamurthy. Geoscience Cyberinfrastructure in the Cloud: Data-Proximate Computing to Address Big Data and Open Science Challenges. In *2017 IEEE 13th International Conference on e-Science (e-Science)*, pages 444–445, Auckland, October 2017. IEEE. ISBN 978-1-5386-2686-3. doi: 10.1109/eScience.2017.63. URL `http://ieeexplore.ieee.org/document/8109168/`.

[46] R. Rew and G. Davis. NetCDF: an interface for scientific data access. *IEEE Computer Graphics and Applications*, 10(4):76–82, 1990. doi: 10.1109/38.56302.

[47] John Caron. HDF5 Dimension Scales, July 2012. URL `https://www.unidata.ucar.edu/blogs/developer/en/entry/dimensions_scales`.

[48] John Caron. HDF5 Dimension Scales - Part 2, July 2012. URL `https://www.unidata.ucar.edu/blogs/developer/en/entry/dimension_scale2`.

[49] John Caron. HDF5 Dimension Scales - Part 3, August 2012. URL `https://www.unidata.ucar.edu/blogs/developer/en/entry/dimension_scales_part_3`.

[50] John Caron. NetCDF-4 Dimensions and HDF5 Dimension Scales, August 2012. URL `https://www.unidata.ucar.edu/blogs/developer/en/entry/netcdf4_shared_dimensions`.

[51] J.-Y. Lee, J. Marotzke, G. Bala, L. Cao, S. Corti, J.P. Dunne, F. Engelbrecht, E. Fischer, J.C. Fyfe, C. Jones, A. Maycock, J. Mutemi, O. Ndiaye, S. Panickal, and T. Zhou. Future Global Climate: Scenario-Based Projections and Near-Term Information. In V. Masson-Delmotte, P. Zhai, A. Pirani, S.L. Connors, C. Péan, S. Berger, N. Caud, Y. Chen, L. Goldfarb, M.I. Gomis, M. Huang, K. Leitzell, E. Lonnoy, J.B.R. Matthews, T.K. Maycock, T. Waterfield, O. Yelekçi, R. Yu, and B. Zhou, editors, *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*, pages 553–672. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2021. doi: 10.1017/9781009157896.006. Type: Book Section.

[52] Elisabeth Lloyd, Greg Lusk, Stuart Gluck, and Seth McGinnis. Varieties of Data-Centric Science: Regional Climate Modeling and Model Organism Research. *Philosophy of Science*, 89(4):802–823, October 2022. ISSN 0031-8248, 1539-767X. doi: 10.1017/psa.2021.50. URL `https://www.cambridge.org/core/product/identifier/S0031824821000507/type/journal_article`.

[53] Ishwarappa and J. Anuradha. A Brief Introduction on Big Data 5Vs Characteristics and Hadoop Technology. *Procedia Computer Science*, 48:319–324, 2015. ISSN 18770509. doi: 10.1016/j.procs.2015.04.188. URL https://linkinghub.elsevier.com/retrieve/pii/S1877050915006973.

[54] Martin Sudmanns, Dirk Tiede, Stefan Lang, Helena Bergstedt, Georg Trost, Hannah Augustin, Andrea Baraldi, and Thomas Blaschke. Big Earth data: disruptive changes in Earth observation data management and analysis? *International Journal of Digital Earth*, 13(7):832–850, July 2020. ISSN 1753-8947, 1753-8955. doi: 10.1080/17538947.2019.1585976. URL https://www.tandfonline.com/doi/full/10.1080/17538947.2019.1585976.

[55] Miguel D. Mahecha, Fabian Gans, Gunnar Brandt, Rune Christiansen, Sarah E. Cornell, Normann Fomferra, Guido Kraemer, Jonas Peters, Paul Bodesheim, Gustau Camps-Valls, Jonathan F. Donges, Wouter Dorigo, Lina M. Estupinan-Suarez, Victor H. Gutierrez-Velez, Martin Gutwin, Martin Jung, Maria C. Londoño, Diego G. Miralles, Phillip Papastefanou, and Markus Reichstein. Earth system data cubes unravel global multivariate dynamics. *Earth System Dynamics*, 11(1): 201–234, February 2020. ISSN 2190-4987. doi: 10.5194/esd-11-201-2020. URL https://esd.copernicus.org/articles/11/201/2020/.

[56] John L. Dwyer, David P. Roy, Brian Sauer, Calli B. Jenkerson, Hankui K. Zhang, and Leo Lymburner. Analysis Ready Data: Enabling Analysis of the Landsat Archive. *Remote Sensing*, 10(9):1363, August 2018. ISSN 2072-4292. doi: 10.3390/rs10091363. URL https://www.mdpi.com/2072-4292/10/9/1363.

[57] Jose Garcia, Peter Fox, Patrick West, and Stephan Zednik. Developing service-oriented applications in a grid environment: Experiences using the OPeNDAP back-end-server. *Earth Science Informatics*, 2(1-2):133–139, June 2009. ISSN 1865-0473, 1865-0481. doi: 10.1007/s12145-008-0017-0. URL https://link.springer.com/10.1007/s12145-008-0017-0.

[58] John Caron, Ethan Davis, Marcos Hermida, Dennis Heimbigner, Sean Arms, Christian Ward-Garrison, Ryan May, Lansing Madry, Robb Kambic, and Hailey Johnson. Unidata THREDDS Data Server, 1997. URL http://www.unidata.ucar.edu/software/tds/. Language: en Medium: application/java-archive.

[59] OGC. WPS 2.0.2 Interface Standard, March 2015. URL http://docs.opengeospatial.org/is/14-065/14-065.html.

[60] Stephan Hoyer and Joe Hamman. xarray: N-D labeled Arrays and Datasets in Python. *Journal of Open Research Software*, 5(1):10, April 2017. ISSN 2049-9647.

doi: 10.5334/jors.148. URL `https://openresearchsoftware.metajnl.com/article/10.5334/jors.148/`.

[61] M. Iturbide, J. Bedia, S. Herrera, J. Baño-Medina, J. Fernández, M.D. Frías, R. Manzanas, D. San-Martín, E. Cimadevilla, A.S. Cofiño, and J.M. Gutiérrez. The R-based climate4R open framework for reproducible climate data access and post-processing. *Environmental Modelling & Software*, 111:42–54, January 2019. ISSN 13648152. doi: 10.1016/j.envsoft.2018.09.009. URL `https://linkinghub.elsevier.com/retrieve/pii/S1364815218303049`.

[62] David Hassell, Jonathan Gregory, Jon Blower, Bryan N. Lawrence, and Karl E. Taylor. A data model of the Climate and Forecast metadata conventions (CF-1.6) with a software implementation (cf-python v2.1). *Geoscientific Model Development*, 10(12):4619–4646, December 2017. ISSN 1991-9603. doi: 10.5194/gmd-10-4619-2017. URL `https://gmd.copernicus.org/articles/10/4619/2017/`.

[63] Julius Busecke, Markus Ritschel, Elizabeth Maroon, Tom Nicholas, and Readthedocs-Assistant. jbusecke/xMIP: v0.7.1, January 2023. URL `https://zenodo.org/record/3678662`.

[64] Anderson Banihirwe, Matthew Long, Max Grover, bonnland, Julia Kent, Pascal Bourgault, Dougie Squire, Julius Busecke, Aaron Spring, Hauke Schulz, Kevin Paul, RondeauG, and Tobias Kölling. intake/intake-esm: intake-esm v2023.11.10, November 2023. URL `https://zenodo.org/doi/10.5281/zenodo.3491062`.

[65] Nathan Collier, Max Grover, and Jemma Stachelek. esgf2-us/intake-esgf, April 2024. URL `https://github.com/esgf2-us/intake-esgf`.

[66] John Caron, Ethan Davis, Marcos Hermida, Dennis Heimbigner, Sean Arms, Christian Ward-Garrison, Ryan May, Lansing Madry, Robb Kambic, Howard Van Dam II, and Hailey Johnson. Unidata NetCDF-Java Library, 2009. URL `https://www.unidata.ucar.edu/software/netcdf-java/`.

[67] Martin Durant. fsspec/kerchunk, 2024. URL `https://github.com/fsspec/kerchunk`.

[68] David Hassell, Jonathan Gregory, Neil R. Massey, Bryan N. Lawrence, and Sadie L. Bartholomew. NetCDF Climate and Forecast Aggregation (CFA) Conventions, 2023. URL `https://github.com/NCAS-CMS/cfa-conventions/blob/main/source/cfa.md`.

[69] H.K. Ramapriyan, P.J.T. Leonard, E.M. Armstrong, S.J.S. Khalsa, D.K. Smith, L.F. Iredell, D.M. Wright, G.J. Huffman, and T.R. Walker. Data Product Development

Guide (DPDG) for Data Producers, Verion 2, 2024. URL https://www.earthdata.nasa.gov/engage/data-producer-resources/data-product-development-standards. Publisher: NASA Earth Science Data and Information System Standards Coordination Office.

[70] Monya Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604): 452–454, May 2016. ISSN 0028-0836, 1476-4687. doi: 10.1038/533452a. URL https://www.nature.com/articles/533452a.

[71] Jiawei Wang, Tzu-yang Kuo, Li Li, and Andreas Zeller. Assessing and restoring reproducibility of Jupyter notebooks. In *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering*, pages 138–149, Virtual Event Australia, December 2020. ACM. ISBN 978-1-4503-6768-4. doi: 10.1145/3324884.3416585. URL https://dl.acm.org/doi/10.1145/3324884.3416585.

[72] Jialin Liu, Quincey Koziol, Gregory F. Butler, Neil Fortner, Mohamad Chaarawi, Houjun Tang, Suren Byna, Glenn K. Lockwood, Ravi Cheema, Kristy A. Kallback-Rose, Damian Hazen, and Mr Prabhat. Evaluation of HPC Application I/O on Object Storage Systems. In *2018 IEEE/ACM 3rd International Workshop on Parallel Data Storage & Data Intensive Scalable Computing Systems (PDSW-DISCS)*, pages 24–34, Dallas, TX, USA, November 2018. IEEE. ISBN 978-1-7281-0192-7. doi: 10.1109/PDSW-DISCS.2018.00005. URL https://ieeexplore.ieee.org/document/8638426/.

[73] Katie Baynes, Rahul Ramachandran, Dan Pilone, Patrick Quinn, Jason Gilman, Ian Schuler, and Alireza Jazayeri. NASA's EOSDIS Cumulus: Ingesting, Archiving, Managing, and Distributing Earth Science Data from the Commercial Cloud, December 2017. URL https://ntrs.nasa.gov/citations/20180000548.

[74] Bregt Saenen and Lidia Borrell-Damian. Federating research infrastructures in Europe for FAIR access to data: Science Europe Briefing on EOSC. Technical report, European Open Science Cloud (EOSC), November 2022. URL https://zenodo.org/record/7346887. Publisher: Zenodo.

[75] C. L. Gentemann, C. Holdgraf, R. Abernathey, D. Crichton, J. Colliander, E. J. Kearns, Y. Panda, and R. P. Signell. Science Storms the Cloud. *AGU Advances*, 2 (2), June 2021. ISSN 2576-604X, 2576-604X. doi: 10.1029/2020AV000354. URL https://onlinelibrary.wiley.com/doi/10.1029/2020AV000354.

[76] Dean N. Williams, V. Balaji, Luca Cinquini, Sébastien Denvil, Daniel Duffy, Ben Evans, Robert Ferraro, Rose Hansen, Michael Lautenschlager, and Claire Trenham.

A Global Repository for Planet-Sized Experiments and Observations. *Bulletin of the American Meteorological Society*, 97(5):803–816, May 2016. ISSN 0003-0007, 1520-0477. doi: 10.1175/BAMS-D-15-00132.1. URL https://journals.ametsoc.org/doi/10.1175/BAMS-D-15-00132.1.

[77] Luca Cinquini, Daniel Crichton, Chris Mattmann, John Harney, Galen Shipman, Feiyi Wang, Rachana Ananthakrishnan, Neill Miller, Sebastian Denvil, Mark Morgan, Zed Pobre, Gavin M. Bell, Bob Drach, Dean Williams, Philip Kershaw, Stephen Pascoe, Estanislao Gonzalez, Sandro Fiore, and Roland Schweitzer. The Earth System Grid Federation: An open infrastructure for access to distributed geospatial data. In *2012 IEEE 8th International Conference on E-Science*, pages 1–10, Chicago, IL, USA, October 2012. IEEE. ISBN 978-1-4673-4466-1 978-1-4673-4467-8 978-1-4673-4465-4. doi: 10.1109/eScience.2012.6404471. URL http://ieeexplore.ieee.org/document/6404471/.

[78] Ruth Petrie, Sébastien Denvil, Sasha Ames, Guillaume Levavasseur, Sandro Fiore, Chris Allen, Fabrizio Antonio, Katharina Berger, Pierre-Antoine Bretonnière, Luca Cinquini, Eli Dart, Prashanth Dwarakanath, Kelsey Druken, Ben Evans, Laurent Franchistéguy, Sébastien Gardoll, Eric Gerbier, Mark Greenslade, David Hassell, Alan Iwi, Martin Juckes, Stephan Kindermann, Lukasz Lacinski, Maria Mirto, Atef Ben Nasser, Paola Nassisi, Eric Nienhouse, Sergey Nikonov, Alessandra Nuzzo, Clare Richards, Syazwan Ridzwan, Michel Rixen, Kim Serradell, Kate Snow, Ag Stephens, Martina Stockhause, Hans Vahlenkamp, and Rick Wagner. Coordinating an operational data distribution network for CMIP6 data. *Geoscientific Model Development*, 14(1):629–644, January 2021. ISSN 1991-9603. doi: 10.5194/gmd-14-629-2021. URL https://gmd.copernicus.org/articles/14/629/2021/.

[79] William J. Gutowski Jr., Filippo Giorgi, Bertrand Timbal, Anne Frigon, Daniela Jacob, Hyun-Suk Kang, Krishnan Raghavan, Boram Lee, Christopher Lennard, Grigory Nikulin, Eleanor O'Rourke, Michel Rixen, Silvina Solman, Tannecia Stephenson, and Fredolin Tangang. WCRP COordinated Regional Downscaling EXperiment (CORDEX): a diagnostic MIP for CMIP6. *Geoscientific Model Development*, 9(11): 4087–4095, November 2016. ISSN 1991-9603. doi: 10.5194/gmd-9-4087-2016. URL https://gmd.copernicus.org/articles/9/4087/2016/.

[80] S. Fiore, P. Nassisi, A. Nuzzo, M. Mirto, L. Cinquini, D. Williams, and G. Aloisio. A climate change community gateway for data usage & data archive metrics across the earth system grid federation. In *CEUR Workshop Proceedings*, volume 2975, 2021. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85117857366&partnerID=40&md5=4882870b6cda97c5595337cb15c624b2.

[81] Majid Asadnabizadeh. Critical findings of the sixth assessment report (AR6) of working Group I of the intergovernmental panel on climate change (IPCC) for global climate change policymaking a summary for policymakers (SPM) analysis. *International Journal of Climate Change Strategies and Management*, 15(5):652–670, November 2023. ISSN 1756-8692, 1756-8692. doi: 10.1108/IJCCSM-04-2022-0049. URL `https://www.emerald.com/insight/content/doi/10.1108/IJCCSM-04-2022-0049/full/html`.

[82] Tommaso Venturini, Kari De Pryck, and Robert Ackland. Bridging in network organisations. The case of the Intergovernmental Panel on Climate Change (IPCC). *Social Networks*, 75:137–147, October 2023. ISSN 03788733. doi: 10.1016/j.socnet.2022.01.015. URL `https://linkinghub.elsevier.com/retrieve/pii/S0378873322000156`.

[83] Martina Stockhause and Michael Lautenschlager. CMIP6 Data Citation of Evolving Data. *Data Science Journal*, 16:30, June 2017. ISSN 1683-1470. doi: 10.5334/dsj-2017-030. URL `http://datascience.codata.org/articles/10.5334/dsj-2017-030/`.

[84] Charles Stern, Ryan Abernathey, Joseph Hamman, Rachel Wegener, Chiara Lepore, Sean Harkins, and Alexander Merose. Pangeo Forge: Crowdsourcing Analysis-Ready, Cloud Optimized Data Production. *Frontiers in Climate*, 3:782909, February 2022. ISSN 2624-9553. doi: 10.3389/fclim.2021.782909. URL `https://www.frontiersin.org/articles/10.3389/fclim.2021.782909/full`.

[85] Kieran Findlater, Sophie Webber, Milind Kandlikar, and Simon Donner. Climate services promise better decisions but mainly focus on better data. *Nature Climate Change*, 11(9):731–737, September 2021. ISSN 1758-678X, 1758-6798. doi: 10.1038/s41558-021-01125-3. URL `https://www.nature.com/articles/s41558-021-01125-3`.

[86] Karl E. Taylor, Ronald J. Stouffer, and Gerald A. Meehl. An Overview of CMIP5 and the Experiment Design. *Bulletin of the American Meteorological Society*, 93(4):485–498, April 2012. ISSN 1520-0477. doi: 10.1175/BAMS-D-11-00094.1. URL `https://journals.ametsoc.org/doi/10.1175/BAMS-D-11-00094.1`.

[87] Javier Diez-Sierra, Maialen Iturbide, José M. Gutiérrez, Jesús Fernández, Josipa Milovac, Antonio S. Cofiño, Ezequiel Cimadevilla, Grigory Nikulin, Guillaume Levavasseur, Erik Kjellström, Katharina Bülow, András Horányi, Anca Brookshaw, Markel García-Díez, Antonio Pérez, Jorge Baño-Medina, Bodo Ahrens, Antoinette Alias, Moetasim Ashfaq, Melissa Bukovsky, Erasmo Buonomo, Steven Caluwaerts, Sin Chan Chou, Ole B. Christensen, James M. Ciarlò, Erika Coppola,

Lola Corre, Marie-Estelle Demory, Vladimir Djurdjevic, Jason P. Evans, Rowan Fealy, Hendrik Feldmann, Daniela Jacob, Sanjay Jayanarayanan, Jack Katzfey, Klaus Keuler, Christoph Kittel, Mehmet Levent Kurnaz, René Laprise, Piero Lionello, Seth McGinnis, Paola Mercogliano, Pierre Nabat, Barış Önol, Tugba Ozturk, Hans-Jürgen Panitz, Dominique Paquin, Ildikó Pieczka, Francesca Raffaele, Armelle Reca Remedio, John Scinocca, Florence Sevault, Samuel Somot, Christian Steger, Fredolin Tangang, Claas Teichmann, Piet Termonia, Marcus Thatcher, Csaba Torma, Erik Van Meijgaard, Robert Vautard, Kirsten Warrach-Sagi, Katja Winger, and George Zittis. The Worldwide C3S CORDEX Grand Ensemble: A Major Contribution to Assess Regional Climate Change in the IPCC AR6 Atlas. *Bulletin of the American Meteorological Society*, 103(12):E2804–E2826, December 2022. ISSN 0003-0007, 1520-0477. doi: 10.1175/BAMS-D-22-0111.1. URL `https://journals.ametsoc.org/view/journals/bams/103/12/BAMS-D-22-0111.1.xml`.

[88] Anna Pirani, Andrés Alegria, Alaa Al Khourdajie, Wawan Gunawan, José Manuel Gutiérrez, Kirstin Holsman, David Huard, Martin Juckes, Michio Kawamiya, Nana Klutse, Volker Krey, Robin Matthews, Adam Milward, Charlotte Pascoe, Gerard van der Shrier, Alessandro Spinuso, Martina Stockhause, and Xiaoshi Xing. The implementation of FAIR data principles in the IPCC AR6 assessment process. Task Group on Data Support for Climate Change Assessments (TG-Data) guidance document. *Zenodo*, 2022. doi: 10.5281/zenodo.6504468. 00000.

[89] Martina Stockhause, David Huard, Alaa Al Khourdajie, José M. Gutiérrez, Michio Kawamiya, Nana Ama Browne Klutse, Volker Krey, David Milward, Andrew E. Okem, Anna Pirani, Lina E. Sitz, Silvina A. Solman, Alessandro Spinuso, and Xiaoshi Xing. Implementing FAIR data principles in the IPCC seventh assessment cycle: Lessons learned and future prospects. *PLOS Climate*, 3(12):e0000533, December 2024. ISSN 2767-3200. doi: 10.1371/journal.pclm.0000533. URL `https://journals.plos.org/climate/article?id=10.1371/journal.pclm.0000533`. Publisher: Public Library of Science.

[90] Anna Pirani, Diego Cammarano, Ellie Fisher, Beate Krüss, Robin Matthews, Charlotte Pascoe, Lina Sitz, and Martina Stockhause. Experience in the Implementation of FAIR Data Principles in the WGI AR6 Assessment. Technical report, Zenodo, August 2022. URL `https://zenodo.org/record/6992173`. Version Number: 1.

[91] Anna Pirani, Robin Matthews, and Lina Sitz. AR6 Working Group I FAIR Supplementary Material. Technical report, Task Group on Data Support for Climate Change Assessments, April 2022. URL `https://zenodo.org/record/6451137`. Publisher: Zenodo Version Number: 1.

[92] Jim Gray, David T. Liu, Maria Nieto-Santisteban, Alex Szalay, David J. DeWitt, and Gerd Heber. Scientific data management in the coming decade. *ACM SIGMOD Record*, 34(4):34–41, December 2005. ISSN 0163-5808. doi: 10.1145/1107499.1107503. URL `https://dl.acm.org/doi/10.1145/1107499.1107503`.

[93] John L. Schnase, Tsengdar J. Lee, Chris A. Mattmann, Christopher S. Lynnes, Luca Cinquini, Paul M. Ramirez, Andrew F. Hart, Dean N. Williams, Duane Waliser, Pamela Rinsland, W. Phillip Webster, Daniel Q. Duffy, Mark A. McInerney, Glenn S. Tamkin, Gerald L. Potter, and Laura Carriere. Big Data Challenges in Climate Science: Improving the next-generation cyberinfrastructure. *IEEE Geoscience and Remote Sensing Magazine*, 4(3):10–22, September 2016. ISSN 2168-6831, 2473-2397. doi: 10.1109/MGRS.2015.2514192. URL `http://ieeexplore.ieee.org/document/7570342/`.

[94] Wm. A. Wulf. The Collaboratory Opportunity. *Science*, 261(5123):854–855, August 1993. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.8346438. URL `https://www.science.org/doi/10.1126/science.8346438`.

[95] Dave Snowdon, Elizabeth F. Churchill, and Alan J. Munro. Collaborative Virtual Environments: Digital Spaces and Places for CSCW: An Introduction. In Dan Diaper, Colston Sanger, Elizabeth F. Churchill, David N. Snowdon, and Alan J. Munro, editors, *Collaborative Virtual Environments*, pages 3–17. Springer London, London, 2001. ISBN 978-1-85233-244-0 978-1-4471-0685-2. doi: 10.1007/978-1-4471-0685-2_1. URL `http://link.springer.com/10.1007/978-1-4471-0685-2_1`. Series Title: Computer Supported Cooperative Work.

[96] Brian E. Granger and Fernando Pérez. Jupyter: Thinking and Storytelling With Code and Data. *Computing in Science & Engineering*, 23(2):7–14, 2021. doi: 10.1109/MCSE.2021.3059263.

[97] Fernando Pérez and Brian E. Granger. IPython: a System for Interactive Scientific Computing. *Computing in Science and Engineering*, 9(3):21–29, May 2007. ISSN 1521-9615. doi: 10.1109/MCSE.2007.53. URL `https://ipython.org`. Publisher: IEEE Computer Society.

[98] Project Jupyter, Matthias Bussonnier, Jessica Forde, Jeremy Freeman, Brian Granger, Tim Head, Chris Holdgraf, Kyle Kelley, Gladys Nalvarte, Andrew Osheroff, M. Pacer, Yuvi Panda, Fernando Perez, Benjamin Ragan-Kelley, and Carol Willing. Binder 2.0 - Reproducible, interactive, sharable environments for science at scale. In Fatih Akici, David Lippa, Dillon Niederhut, and M. Pacer, editors, *Proceedings of the 17th Python in Science Conference*, pages 113 – 120, 2018. doi: 10.25080/Majora-4af1f417-011.

[99] Aida Palacio Hoz, Andrés Heredia Canales, Ezequiel Cimadevilla Álvarez, Marta Obregón Ruiz, and Alvaro López García. DataLab as a Service: Distributed computing framework for multi-interactive analysis environments. *IEEE Access*, pages 1–1, 2025. ISSN 2169-3536. doi: 10.1109/ACCESS.2025.3536637. URL `https://ieeexplore.ieee.org/document/10858125/`.

[100] P.A. Arias, N. Bellouin, E. Coppola, R.G. Jones, G. Krinner, J. Marotzke, V. Naik, M.D. Palmer, G.-K. Plattner, J. Rogelj, M. Rojas, J. Sillmann, T. Storelvmo, P.W. Thorne, B. Trewin, K. Achuta Rao, B. Adhikary, R.P. Allan, K. Armour, G. Bala, R. Barimalala, S. Berger, J.G. Canadell, C. Cassou, A. Cherchi, W. Collins, W.D. Collins, S.L. Connors, S. Corti, F. Cruz, F.J. Dentener, C. Dereczynski, A. Di Luca, A. Diongue Niang, F.J. Doblas-Reyes, A. Dosio, H. Douville, F. Engelbrecht, V. Eyring, E. Fischer, P. Forster, B. Fox-Kemper, J.S. Fuglestvedt, J.C. Fyfe, N.P. Gillett, L. Goldfarb, I. Gorodetskaya, J.M. Gutierrez, R. Hamdi, E. Hawkins, H.T. Hewitt, P. Hope, A.S. Islam, C. Jones, D.S. Kaufman, R.E. Kopp, Y. Kosaka, J. Kossin, S. Krakovska, J.-Y. Lee, J. Li, T. Mauritsen, T.K. Maycock, M. Meinshausen, S.-K. Min, P.M.S. Monteiro, T. Ngo-Duc, F. Otto, I. Pinto, A. Pirani, K. Raghavan, R. Ranasinghe, A.C. Ruane, L. Ruiz, J.-B. Sallée, B.H. Samset, S. Sathyendranath, S.I. Seneviratne, A.A. Sörensson, S. Szopa, I. Takayabu, A.-M. Tréguier, B. van den Hurk, R. Vautard, K. von Schuckmann, S. Zaehle, X. Zhang, and K. Zickfeld. Technical Summary. In V. Masson-Delmotte, P. Zhai, A. Pirani, S.L. Connors, C. Péan, S. Berger, N. Caud, Y. Chen, L. Goldfarb, M.I. Gomis, M. Huang, K. Leitzell, E. Lonnoy, J.B.R. Matthews, T.K. Maycock, T. Waterfield, O. Yelekçi, R. Yu, and B. Zhou, editors, *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*, pages 33–144. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2021. doi: 10.1017/9781009157896.002. Type: Book Section.

[101] H.-O. Pörtner, D.C. Roberts, M. Tignor, E.S. Poloczanska, K. Mintenbeck, A. Alegría, M. Craig, S. Langsdorf, S. Löschke, V. Möller, A. Okem, and B. Rama. *Climate Change 2022: Impacts, Adaptation, and Vulnerability. Contribution of Working Group II to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change.* Cambridge University Press, Cambridge, UK and New York, NY, USA, 2022. doi: 10.1017/9781009325844.

[102] IPCC. Annex VIII: Acronyms. In V. Masson-Delmotte, P. Zhai, A. Pirani, S.L. Connors, C. Péan, S. Berger, N. Caud, Y. Chen, L. Goldfarb, M.I. Gomis, M. Huang, K. Leitzell, E. Lonnoy, J.B.R. Matthews, T.K. Maycock, T. Waterfield, O. Yelekçi, R. Yu, and B. Zhou, editors, *Climate Change 2021: The Physical*

Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change, pages 2257–2266. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2021. Type: Book Section.

[103] Copernicus Climate Change Service. Gridded monthly climate projection dataset underpinning the IPCC AR6 Interactive Atlas, 2023. URL `https://cds.climate.copernicus.eu/doi/10.24381/cds.5292a2b0`.

[104] Ezequiel Cimadevilla, Maialen Iturbide, Antonio S. Cofiño, Jesús Fernández, Lina E. Sitz, Aida Palacio, Andrés Heredia, and José M. Gutiérrez. The IPCC Interactive Atlas DataLab: Online reusability for regional climate change assessment. *PLOS Climate*, 4(6):e0000644, June 2025. ISSN 2767-3200. doi: 10.1371/journal.pclm.0000644. URL `https://dx.plos.org/10.1371/journal.pclm.0000644`.

[105] Stefan Lange. Trend-preserving bias adjustment and statistical downscaling with ISIMIP3BASD (v1.0). *Geoscientific Model Development*, 12(7):3055–3070, July 2019. ISSN 1991-9603. doi: 10.5194/gmd-12-3055-2019. URL `https://gmd.copernicus.org/articles/12/3055/2019/`.

[106] Maialen Iturbide, Ana Casanueva, Joaquín Bedia, Sixto Herrera, Josipa Milovac, and José Manuel Gutiérrez. On the need of bias adjustment for more plausible climate change projections of extreme heat. *Atmospheric Science Letters*, 23(2): e1072, February 2022. ISSN 1530-261X, 1530-261X. doi: 10.1002/asl.1072. URL `https://rmets.onlinelibrary.wiley.com/doi/10.1002/asl.1072`.

[107] Ezequiel Cimadevilla and Maialen Iturbide. SantanderMetGroup/IPCC-Atlas-Datalab: PLOSv3, January 2025. URL `https://zenodo.org/doi/10.5281/zenodo.14646227`.

[108] Antonio Lima, Luca Rossi, and Mirco Musolesi. Coding Together at Scale: GitHub as a Collaborative Social Network. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1):295–304, May 2014. ISSN 2334-0770, 2162-3449. doi: 10.1609/icwsm.v8i1.14552. URL `https://ojs.aaai.org/index.php/ICWSM/article/view/14552`.

[109] Dirk Merkel. Docker: lightweight linux containers for consistent development and deployment. *Linux journal*, 2014(239):2, 2014.

[110] Maialen Iturbide, José Manuel Gutiérrez, Lincoln Muniz Alves, Joaquín Bedia, Ezequiel Cimadevilla, Antonio S. Cofiño, Ruth Cerezo-Mota, Alejandro Di Luca, Sergio Henrique Faria, Irina Gorodetskaya, Mathias Hauser, Sixto Herrera, Helene T.

Hewitt, Kevin J. Hennessy, Richard G. Jones, Svitlana Krakovska, Rodrigo Manzanas, Daniel Marínez-Castro, Gemma Teressa Narisma, Intan S. Nurhati, Izidine Pinto, Sonia I. Seneviratne, Bart van den Hurk, and Carolina S. Vera. An update of IPCC climate reference regions for subcontinental analysis of climate model data: Definition and aggregated datasets. *Earth System Science Data Discussions*, pages 1–16, April 2020. ISSN 1866-3508. doi: https://doi.org/10.5194/essd-2019-258. URL `https://essd.copernicus.org/preprints/essd-2019-258/`. Publisher: Copernicus GmbH.

[111] Javier Diez-Sierra, Maialen Iturbide, Jesús Fernández, José M. Gutiérrez, Josipa Milovac, and Antonio S. Cofiño. Consistency of the regional response to global warming levels from CMIP5 and CORDEX projections. *Climate Dynamics*, 61 (7-8):4047–4060, October 2023. ISSN 0930-7575, 1432-0894. doi: 10.1007/s00382-023-06790-y. URL `https://link.springer.com/10.1007/s00382-023-06790-y`.

[112] Ezequiel Cimadevilla Alvarez and Maialen Iturbide. SantanderMetGroup/IPCC-Atlas-Datalab: PLOSv1, December 2024. URL `https://zenodo.org/doi/10.5281/zenodo.14524856`.

[113] Ezequiel Cimadevilla, Bryan N. Lawrence, and Antonio S. Cofiño. The Earth System Grid Federation (ESGF) Virtual Aggregation (CMIP6 v20240125). *Geoscientific Model Development*, 18(8):2461–2478, April 2025. ISSN 1991-9603. doi: 10.5194/gmd-18-2461-2025. URL `https://gmd.copernicus.org/articles/18/2461/2025/`.

[114] Stefano Nativi, Paolo Mazzetti, and Max Craglia. A view-based model of data-cube to support big earth data systems interoperability. *Big Earth Data*, 1(1-2):75–99, December 2017. ISSN 2096-4471, 2574-5417. doi: 10.1080/20964471.2017.1404232. URL `https://www.tandfonline.com/doi/full/10.1080/20964471.2017.1404232`.

[115] James Gallagher, Nathan Potter, Rmorris2342, Captain James Tiberius Kirk, Kodi Neumiller, The Robot Travis, Tsgouros, Blackone-Sudo, Kyang2014, Slav Korolev, Dan Horák, Ethan Davis, H. Joe Lee, Yuanho, Orion Poplawski, Ryan Schmidt, and Sam Lloyd. OPENDAP/libdap4: libdap 3.20.11 for Hyrax 1.16.8, July 2022. URL `https://zenodo.org/record/6878992`.

[116] Brian E. J. Rose, John Clyne, Ryan May, James Munroe, Amelia Snyder, Orhan Eroglu, and Kevin Tyle. Collaborative Research: GEO OSE TRACK 2: Project Pythia and Pangeo: Building an inclusive geoscience community through accessible, reusable, and reproducible workflows, July 2023. URL `https://zenodo.org/record/8184298`. Publisher: Zenodo.

[117] Neil C. Swart, Jason N. S. Cole, Viatcheslav V. Kharin, Mike Lazare, John F. Scinocca, Nathan P. Gillett, James Anstey, Vivek Arora, James R. Christian, Sarah Hanna, Yanjun Jiao, Warren G. Lee, Fouad Majaess, Oleg A. Saenko, Christian Seiler, Clint Seinen, Andrew Shao, Michael Sigmond, Larry Solheim, Knut Von Salzen, Duo Yang, and Barbara Winter. The Canadian Earth System Model version 5 (CanESM5.0.3). *Geoscientific Model Development*, 12(11):4823–4873, November 2019. ISSN 1991-9603. doi: 10.5194/gmd-12-4823-2019. URL `https://gmd.copernicus.org/articles/12/4823/2019/`.