GreenDIGIT

Greener Future Digital Research Infrastructures

Deliverable 4.1 State of the Art in RI and digital infrastructure sustainability and technologies assessment for energy efficiency and impact

GRANT AGREEMENT NUMBER: 101131207

This project has received funding from the European Union's HE research and innovation programme under the grant agreement No. 101131207

Lead Beneficiary:	PSNC
Type of Deliverable:	Report
Dissemination Level:	Public
Submission Date:	17.03.2025
Version:	2.0 Revised version

Versioning and contribution history

Version	Description	Contributions
0.1	Initial version	Krzysztof Dombek
0.2	Sections structure	Gergely Sipos
0.2a	Contributions to all sections	All Authors
0.3	Integrate all contributions and formatting	Krzysztof Dombek
0.3a	Deliverable review and references	Yuri Demchenko, Gergely Sipos
0.4a	Pre-final working draft	Krzysztof Dombek
0.5	Pre-final draft	Krzysztof Dombek, Gergely Sipos, Yuri Demchenko
0.6	Final Draft	Krzysztof Dombek
1.0	Final Version	Ineke Brouwer, Naomi van der Most
2.0	Revised version – corrected author list	Ineke Brouwer

Authors

Author	Partner	Contribution to Section
Gergely Sipos	EGI	1, 2, 3.1
Catalin Condurache	EGI	3.2, 3.4
Álvaro López García	CSIC	3.3
Zdenek Sustr	CESNET	3.2, 4
Jiří Sitera	CESNET	3.5, 4
Edouard Guegain	Greenspector	3.6, 5.3
Jozsef Kovacs	SZTAKI	4.3
Tamas Maray	SZTAKI	4.3
Andrei Tsaregorodtsev	CNRS	4
Kostas Chounos	UTH	5.1
Nicole Simone Martinelli	CNIT	5.2
Alessandro Musumeci	CNIT	5.2
Krzysztof Dombek	PSNC	5.4, 6.3
Shashikant Ilager	UvA	6.1
Gonçalo Teixeira de Pinho Ferreira	UvA	6.2
Sebastian Gallenmüller	TUM	6.3
Kilian Holzinger	TUM	6.3
Roberto Trasarti	CNR	6.2
Andrea Passarella	CNR	6.2
Yuri Demchenko	UvA	7, 8
Adnan Tahir	UvA	7.3

Reviewers

Name	Organisation
Yuri Demchenko	UvA
Gergely Sipos	EGI

Disclaimer

The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Commission. The Commission does not guarantee the accuracy of the data included in this study. Neither the Commission nor any person acting on the Commission's behalf may be held responsible for the use, which may be made of the information contained therein.

Table of contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Description
API	Application Programming Interface
AKS	Azure Kubernetes Services
AWS	Amazon Web Services
BESS	Battery Energy Storage System
CDN	Content Delivery Network
DER	Distributed Energy Resource
DMI	Data Management Infrastructure
EDC	Total Energy Consumption (kWh)
EED	European Energy Efficiency Directive
ENTSO-E	European Association of TSOs
EOSC	European Open Science Cloud
EPC	Energy Proportionality Coefficient
FAIR	Findable, Accessible, Interoperable, and Reusable
GenAI	Generative AI
GSF	Green Software Foundation
GSMM	Green Software Measurement Model
GW	Gateway
HTC	High-throughput Computing
HPC	High-performance Computing
KSM	Kube State Metrics
LLM	Large Language Model
ML	Machine Learning
NCDD	Nederlands Coalitie Duurzame Digitalisering
NEEE	Network Equipment Energy Efficiency
MDAF	Management Data Analytics Function
NWDAF	Network Data Analytics Function
PCDR	Power Consumption per Data Rate
PDU	Power Distribution Unit
PUE	Power Usage Effectiveness
RAPL	Running Average Power Limit
REST	Representational State Transfer
RI	Research Infrastructure
SCI	Software Carbon Intensity
SEMD	Smart Energy Management Device
SGX	Software Guard Extension
SoC	System-on-Chip
SotA	State of the Art
STC	Standardized Testing Conditions
TRL	Technology Readiness Level
TSDB	Time-Series Database

TSO	Transmission System Operators
UDM	Unified Data Management
VM	Virtual Machine
VRE	Virtual Research Environment
WMS	Workload Management System

Executive Summary

The GreenDIGIT project is a European initiative aimed at reducing the environmental footprint of Research Infrastructures (RIs) and digital services by enhancing energy efficiency and sustainability. As digital infrastructures continue to grow in scale and complexity, their energy consumption and carbon emissions have become pressing concerns. This report, Deliverable D4.1, provides a comprehensive State of the Art assessment of current technologies, methodologies, and policies that influence the sustainability of RIs, with a particular focus on metrics, optimization strategies, and software solutions for energy-efficient operations.

The presented **State of the Art** (SotA) analysis provides a comprehensive overview of existing technologies, methodologies, and best practices for improving energy efficiency and enhancing the sustainability of scientific computing and digital RIs. It serves as a foundational study for WP4, WP5, and WP6, supporting the design, development, and deployment of GreenDIGIT's sustainability-oriented frameworks and tools addressing aspects such as energy efficiency, carbon-aware computing, sustainable data management, and research reproducibility.

The deliverable assesses technological approaches and best practices across multiple infrastructure layers and the data collection and processing continuum, including cloud computing, high-throughput computing (HTC), 5G networks, and IoT environments, providing insights into how digital RIs can optimize their operations while minimizing energy consumption and carbon emissions.

The document is structured around major technology domains that play a crucial role in reducing the environmental impact of digital infrastructures: metrics and monitoring, scientific workflow scheduling and optimisation, energy efficient networks, research data management, experimental research reproducibility, and overall sustainable software design practices. Each chapter provides an overview of existing technologies, ongoing research, and potential improvements that will shape the next-generation sustainable RIs.

The deliverable provides an extended analysis of existing practices for data centre energy efficiency and carbon footprint metrics collection and monitoring. The report outlines the European Energy Efficiency Directive (EED) reporting requirements for data centres, highlighting metrics such as Power Usage Effectiveness (PUE), renewable energy usage, and water efficiency. Existing monitoring infrastructures like EGI, Euroean Open Science Cloud (EOSC), and Prometheus are assessed for their applicability in federated research environments. The study identifies gaps in carbon emission tracking and energy efficiency reporting, proposing an energy data model for harmonized monitoring across different RIs.

A key finding of this study is that RIs often lack the necessary tools and frameworks to effectively measure, monitor, and optimize their environmental impact. While significant advances have been made in energy-efficient computing, carbon-aware workload scheduling, and optimized data storage, their adoption remains fragmented. Scientific workflows are often executed without consideration of energy efficiency or carbon intensity, leading to excessive energy consumption and inefficiencies across HTC, high-performance computing (HPC), and cloud environments. Similarly, scientific data management strategies are not yet optimized for sustainability, with redundant data storage and inefficient replication practices contributing to unnecessary energy usage.

The presented analysis identified a need for GreenDIGIT to work on the development of federated environmental monitoring infrastructures, enabling RIs to track, report, and optimize their sustainability metrics in real-time. The project is also exploring energy-aware scheduling mechanisms, which will help align computational workloads with low-carbon energy availability. These advancements will significantly improve energy efficiency and carbon footprint management for scientific computing environments.

The overview and analysis of sustainable software design principles and practices will help the project to implement these practices in scientific applications design and efficient use. By enhancing Virtual Research Environments (VREs) and integrating intelligent experiment reproducibility frameworks, the project ensures that research workflows and experimental data collection can be reused and optimized, reducing redundant computations and minimizing energy waste.

Based on the presented State of the Art analysis, the deliverable provides key recommendations for energy efficiency and environmental sustainability of the future RIs:

- Standardized environmental metrics reporting across all European RIs, leveraging federated monitoring platforms.
- Adoption of energy-aware workload scheduling, integrating real-time carbon intensity data for optimized task execution.
- Implementation of sustainable software design practices, including energy-efficient programming, hardware-aware optimization, and carbon-aware software execution.
- Enhancing scientific experiment reproducibility through low-impact digital experiment frameworks, reducing redundant computational efforts and improving efficiency of research data management, in particular compliance with the Findable, Accessible, Interoperable, and Reusable (FAIR) data principles.

The report also touches on the need to address energy efficiency of the growing use of Generative AI (GenAI) in future scientific research.

The presented analysis and identified gaps and research and development topics will provide a basis for GreenDIGIT to develop and validate prototype necessary solutions and services, helping European RIs transition to greener, more energy-efficient operations. These efforts will contribute to reducing the environmental impact of scientific computing while maintaining high-performance research capabilities.

1 Introduction

Lowering the environmental impact of digital services and technologies has to become a priority for both the operation of existing digital services and the design of future digital infrastructures. Indeed, digital infrastructures are responsible for 5 to 9 % of the world's total electricity use and more than 2 % of global emissions. In 2018, data centres accounted for 2.7 % of the electricity demand in the EU-28. Within the Union, data centres accounted for 2.7 % of electricity demand in 2018 and will reach 3.21 % by 2030 if development continues on the current trajectory[1].

The GreenDIGIT project[2], funded in the Horizon Europe RI programme runs between 2024-2026 to research, prototype and validate various technical and non-technical approaches and tools that digital service providers within RI communities can adopt to lower their environmental impact.

GreenDIGIT will deliver contributions and will move beyond State of the Art in 3 areas:

1. Policy and Governance: Establishing frameworks for measuring and managing the environmental impact of digital services through the full RI lifecycle.
2. Resource Efficiency: Develop software tools, frameworks and infrastructures that can support digital service providers in measuring and lowering their environmental impact by optimising operation.
3. Skills and Collaboration: Run community building and training programmes for RI provider and user communities to increase their understanding and skills about environmental impact lowering techniques and to foster cooperation among RIs around such topics.

This deliverable is a SotA study covering the 2nd area from this list. A separate milestone document, titled "MS8.1 Study of existing policies and regulations" provides SotA for area 1, and another deliverable, titled "D10.2 Definition of the competences and skills for sustainability and environmental impact awareness and a set of training modules" already offer solutions for area 3.

Chapter 2 of the document details the GreenDIGIT contributions to area 2, by positioning the various software tools/frameworks/infrastructures of the project into a single architecture. This architecture, and the various areas it covers serves as a basis for the chapters of this SotA.

The remainder of the deliverable is structured in the following way. Chapter 3 provides an extensive overview of the metrics for data centre environmental sustainability. Chapter 4 is focused on the scheduling approaches for green optimisation of scientific workflows. Chapter 5 provides existing practices, approaches and used metrics for monitoring the energy efficiency of networks, 5G infrastructure, IoT and mobile environment. Chapter 6 is covering sustainability aspects in research infrastructure for experimental research including sustainability in Virtual RIs and experiment reproducibility metadata formats. Chapter 7 provides an overview of the sustainable software design practices that are important for scientific software and applications. This section also provides an overview of the best practices recommended by the leading cloud providers Amazon Web Services (AWS) and Microsoft Azure that are defined as a part of their Well-Architected Framework. Section 8 analyses data management aspects for both supporting environmental research and ensuring energy

[1] Directive (EU) 2023/1791 of the European Parliament and of the Council of 13 September 2023 on energy efficiency and amending Regulation: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ%3AJOL_2023_231_R_0001&qid=1695186598766

[2] GreenDIGIT Project: https://greendigit-project.eu/

efficiency of the data management infrastructure; the latter is important for research and for industry that both face challenges with collecting, storing and processing huge amounts of data. Chapter 9 touches on the need to address energy efficiency of the growing use of GenAI in future scientific research. The final section provides a summary and vision for future implementation of recommendations that come out of the presented State of the Art analysis.

2 GreenDIGIT software solutions

In the second half of 2024, GreenDIGIT ran a comprehensive landscape analysis[3] about digital RI practices and identified needs and gaps in existing practices, approaches, metrics, and tools in addressing environmental sustainability and impact lowering. The landscape study revealed a gap in the existence and consistent adoption of software tools within IT centres and user communities to track their environmental footprint and to optimise their operations to lower such impacts.

The GreenDIGIT project aims to fill this gap by providing solutions to energy-related impact minimisation. The focus is on energy, because the landscape analysis showed that digital service providers consider energy consumption as their biggest polluting factor. The list below details the various technical solutions that GreenDIGIT started working on, and will deliver and validate in 2025-26. These solutions are also presented in Figure 1 in an architecture that positions them towards.

1. Metrics collection infrastructure to support RIs in the collection, aggregation and analysis of environmental impact-related metrics of their digital services. The infrastructure should be capable to work across the spectrum of equipment used within digital RIs (network-edge-IoT-cloud-HTC-HPC). (Lead: EGI, contributions from EGI, SLICES, SoBigData, EBRAINS)
2. Federated data management infrastructure (DMI) to support RIs in the optimal use of their storage and network equipment by optimising data transfer, data replication and data storage instructions within federated environments. (Lead: CNR)
3. Digital scientific experiment[4] reproducibility infrastructure to support RIs in the registration, lookup and re-play of previously conducted digital experiments, eliminating/minimising the need of re-executing scientific calculations that have been already performed in the past. (CNR, TUM)
4. Environments and frameworks that provide integrated support for users to the lowering of the environmental footprint of their activities on digital RIs, by building on, and expanding the solutions from the previous 1, 2, 3 points. Particularly: Making informed decisions for resource allocation using data from the metrics infrastructure (area 1); Minimising data transfers and storage with the use of the federated data infrastructure (area 2); Offering scientific reproducibility with minimised computational impact (area 3). The implementations foreseen within this area are:
    4.1. An extension to the AI4OS AI platform that can optimise the execution of AI model training and inference tasks on federated cloud-HPC infrastructures. (IFCA-CSIC)
    4.2. An extension to the DIRAC middleware that can run massively parallel applications on federated High Throughput Cluster resources. (CNRS)
    4.3. A JupyterHub environment that integrates with the rest of the ecosystem and can serve as a front-end towards users and user communities to perform computational analysis on digital RI services, with minimal impact on the environmental. (TUM, UvA)
    4.4. A data centre automation service that monitors the use of computers, the trend of incoming user requests and based on these switches HW on/off to lower energy consumption without causing Quality-of-Service failures towards the users. (SZTAKI)
    4.5. Workload manager (resource broker) that can choose the most suitable type of HW resources based on job characteristics inside one compute centre or among multiple centres, considering the environmental impact and expected time of completion of compute tasks. (E.g. for compute-heavy, I/O heavy, memory heavy tasks) (CESNET)

[3] Deliverable 3.1 RIs Landscape review, best practices analysis and identification of needs within the ESFRI RIs

[4] The word ‘experiment’ in this context is used for any type/form of data analysis application that runs on digital infrastructures. Experiments can exist in the form of workflows, VMs, containers, compute jobs, etc.

4.6. Batch scheduler that can delay the execution of jobs to periods when the energy used by the centre is lower in carbon content, making the best compromise between time of completion and environmental impact. (CESNET)

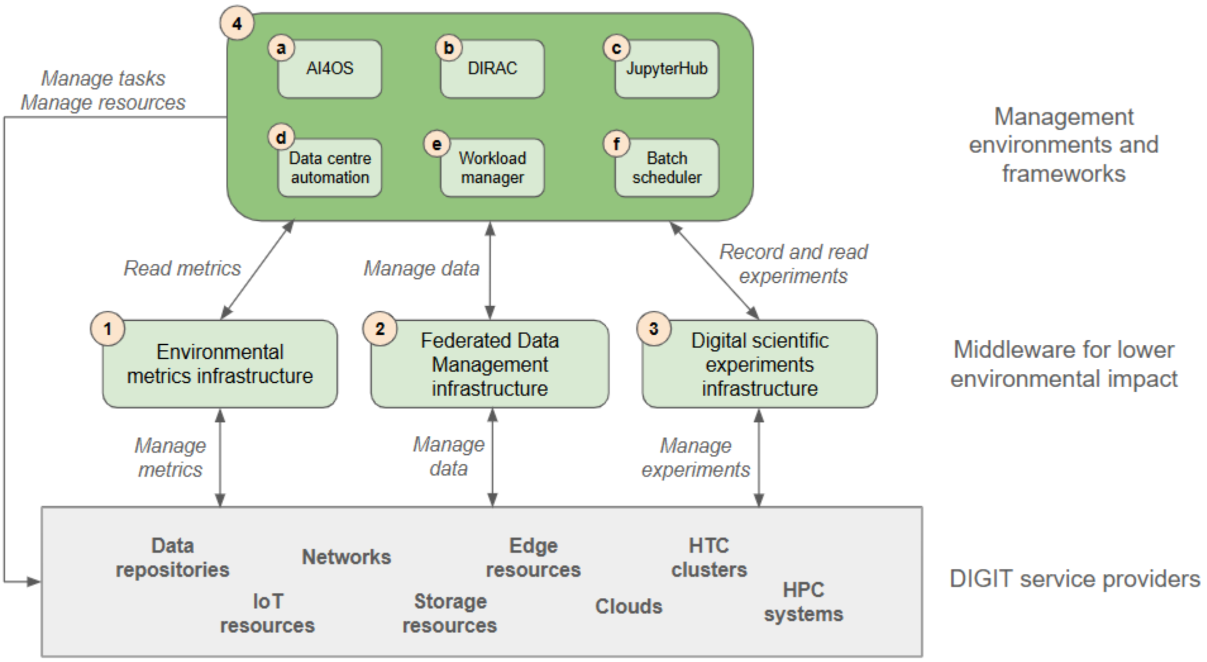


Figure 1. GreenDIGIT software solutions for lower environmental impact (green boxes with orange labels). See the description of each box above.

The next sections of this document go into the SotA review of the context of these GreenDIGIT developments.

3 Metrics for data centre environmental sustainability

3.1 European Regulation for Data Centre Environmental Sustainability Reporting

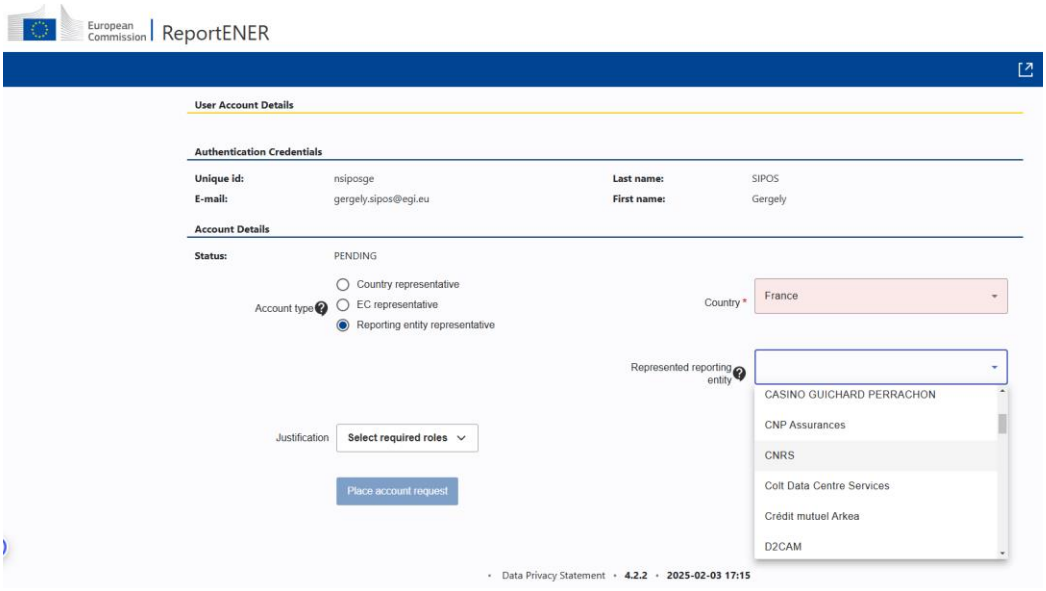
In the evolving landscape of data centre management, an increasing emphasis is placed on the intersection of sustainability and compliance with government regulations. As the pace of digital transformation continues to accelerate, it fuels an increasing demand for data centres, culminating in heightened energy consumption, carbon emissions, and water usage. However, in the face of these challenges, various government directives around the world have come to the forefront as key regulatory frameworks designed to manage these environmental impacts. The most relevant for GreenDIGIT in this respect is the EED[5]. According to the EED:

- By 15th of May 2024 and annually thereafter, Member States shall require owners and operators of data centres with a non-redundant rated electrical load of at least 500 kW to (1) monitor and report various energy performance metrics. These metrics include energy consumption, power utilization, temperature, heat utilization, and the use of renewable energy AND (2) to implement energy recovery facilities, aiming for up to 20 % energy recovery, which can be achieved by connecting to district heating networks.
- Newly constructed data centres are expected to achieve a PUE value of 1.2. Additionally, from 2026, data centres must source all their energy from renewable sources, either physically or virtually.

The EC database where the data centres have to report is already available[6]. Some of the countries already have data centres listed, and this includes some academic data centres too (e.g. CNRS from France as shown in the screenshot in Figure 2). This reporting requirement will gradually broaden to additional countries and data centres too, and will penetrate more deeply into the academic data centre community (at least to the large centres). Therefore studying the metrics that the EED demands can serve as a good baseline for SotA metrics.

[5] Directive (EU) 2023/1791 of the European Parliament and of the Council of 13 September 2023 on energy efficiency and amending Regulation (EU) 2023/955 (recast) (Text with EEA relevance): https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ%3AJOL_2023_231_R_0001&qid=1695186598766

[6] European database on data centres: https://ec.europa.eu/energy-climate-plans-reporting/ePlatform/reportENER/screen/home



User Account Details

Authentication Credentials

Unique id: nsiposge
Last name: SIPOS
E-mail: gergely.sipos@egi.eu
First name: Gergely

Account Details

Status: PENDING

Account type
○ Country representative
○ EC representative
● Reporting entity representative

Country * France

Represented reporting entity

CASINO GUICHARD PERRACHON
CNP Assurances
CNRS
Colt Data Centre Services
Crédit mutuel Arkea
D2CAM
Justification
Select required roles
Place account request
Data Privacy Statement • 4.2.2 • 2025-02-03 17:15

Figure 2 The EC reporting portal

The EED reporting obligations encompass a total of 24 performance indicators, which can be categorized into three main areas:

1. **AREA: Energy and sustainability indicators:**
    1. Installed information technology power demand (PDIT, kW)
    2. Data centre total floor area (SDC, m²)
    3. Data centre computer room floor area (SCR, m²)
    4. Total energy consumption (EDC, kWh)
    5. Total energy consumption of IT equipment (EIT, kWh)
    6. Electrical grid functions
    7. Average battery capacity (‘CBtG’, in kW)
    8. Total water input (‘WIN’, in cubic metres)
    9. Total potable water input (‘WIN-POT’, in cubic metres)
    10. Waste heat reused (‘EREUSE’, in kWh)
    11. Average waste heat temperature (‘TWH’, in degree Celsius)
    12. Average setpoint information technology equipment intake air temperature (‘TIN’, in degree Celsius)
    13. Types of refrigerants used in the cooling and air conditioning equipment
    14. Cooling degree days (‘CDD’, in degree-days)
    15. Total renewable energy consumption (‘ERES-TOT’, in kWh)
    16. Total renewable energy consumption from Guarantees of Origin (‘ERES-GOO’, in kWh)
    17. Total renewable energy consumption from Power Purchasing Agreements (‘ERES-PPA’, in kWh)
    18. Total renewable energy consumption from on-site renewables (‘ERES-OS’, in kWh)

2. **AREA: ICT capacity indicators**
    1. ICT capacity for servers (‘CSERV’)
    2. ICT capacity for storage equipment (‘CSTOR’, in petabytes)
3. **AREA: Data traffic indicators**
    1. Incoming traffic bandwidth (‘BIN’, in gigabytes per second)
    2. Outgoing traffic bandwidth (‘BOUT’, in gigabytes per second)
    3. Incoming data traffic (‘TIN’, in exabytes)
    4. Outgoing data traffic (‘TOUT’, in exabytes)

Notice that while the regulation emphasizes energy-related metrics, it does not explicitly mandate the reporting of total CO_2 emissions. However, by reporting energy consumption and the proportion of renewable energy used, data centres provide data that can be used to estimate their carbon footprint. Therefore, while direct reporting of total CO_2 emissions is not explicitly required, the reported energy metrics contribute to assessing the environmental impact of data centre operations. Annex III of EED specifies calculation methodologies for the following **sustainability indicators** that shall be calculated based on the information and key performance indicators communicated to the European database on data centres:

- PUE
- Water Usage Effectiveness (WUE)
- Energy Reuse Factor (ERF)
- Renewable Energy Factor (REF)

3.2 Metrics infrastructures for federated environments

This section provides an overview of those existing software infrastructures that can be considered a baseline for the ‘GreenDIGIT Environmental Metrics Infrastructure’.

3.2.1 EGI Accounting system for metrics

The EGI Accounting Repository and Portal is a service designed to collect, store, aggregate, and display information about the consumption of resources for High Throughput Compute jobs, VMs, and online storage providers. Providers supporting these types of services can connect their different service endpoints to the Accounting Service, which is centrally managed and will provide collected data about the service usage.

Probes and sensors that gather accounting information according to certain data formats are deployed locally at service providers, and subsequently a network of message brokers is forwarding data to a central Accounting Repository where the data are processed to generate various summaries and views for display in the Accounting Portal[7].

Current metrics are ‘*Number of VMs*’ (exemplified in Figure 3), ‘*Disk Used*’, ‘*Memory Used*’, ‘*Number of ProcessorHours*’ (for Cloud), ‘*Number of Jobs*’, ‘*CPU Efficiency*’, ’*Total CPU time used*’, ‘*Number of ProcessorHours*’ (HTC) and they can be displayed by Resource Centres and various time units (from

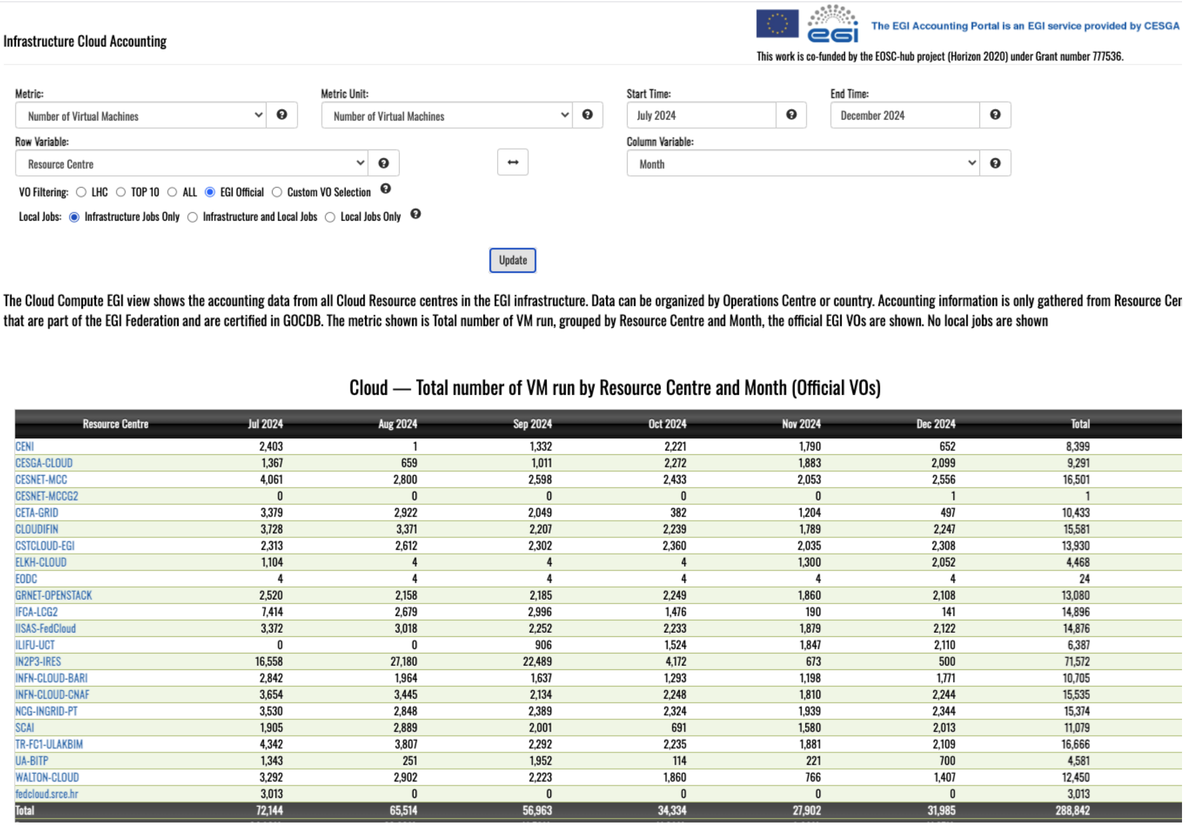
[7] EGI Accounting Portal: https://accounting.egi.eu/

’*Month*’ to ‘*Year*’). A common format to report GPU use and utilization in infrastructure clouds has also been developed, and different components of the ecosystem are being extended to support it.

From the GreenDIGIT perspective, specific probes and sensors that are implemented within the project partner premises could gather relevant environmental impact related metrics and feed them in the metrics framework at the Accounting Portal level to be consumed via API by a broad category of readers (platforms, tools, brokers, web pages).

Features useful for the project and provided by the EGI Accounting framework are:

- scalable for O(100) providers
- scalable for O(10) consumers
- can store historical values for the stored metrics
- support for time-series queries
- available GUI and read-write API
- support for cumulative reporting of consumption of long-running workflows



Infrastructure Cloud Accounting

The EGI Accounting Portal is an EGI service provided by CESGA

This work is co-funded by the EOSC-hub project (Horizon 2020) under Grant number 777536.

Metric: Number of Virtual Machines

Metric Unit: Number of Virtual Machines

Start Time: July 2024

End Time: December 2024

Row Variable: Resource Centre

Column Variable: Month

VO Filtering: ○ LHC ○ TOP 10 ○ ALL ● EGI Official ○ Custom VO Selection

Local Jobs: ● Infrastructure Jobs Only ○ Infrastructure and Local Jobs ○ Local Jobs Only

The Cloud Compute EGI view shows the accounting data from all Cloud Resource centres in the EGI infrastructure. Data can be organized by Operations Centre or country. Accounting information is only gathered from Resource Cer
that are part of the EGI Federation and are certified in GOCDB. The metric shown is Total number of VM run, grouped by Resource Centre and Month, the official EGI VOs are shown. No local jobs are shown

Cloud — Total number of VM run by Resource Centre and Month (Official VOs)

Resource Centre	Jul 2024	Aug 2024	Sep 2024	Oct 2024	Nov 2024	Dec 2024	Total
CENI	2,403	1	1,332	2,221	1,790	652	8,399
CESGA-CLOUD	1,367	659	1,011	2,272	1,883	2,099	9,291
CESNET-MCC	4,061	2,800	2,598	2,433	2,053	2,556	16,501
CESNET-MCCG2	0	0	0	0	0	1	1
CETA-GRID	3,379	2,922	2,049	382	1,204	497	10,433
CLOUDIFIN	3,728	3,371	2,207	2,239	1,789	2,247	15,581
CSTCLOUD-EGI	2,313	2,612	2,302	2,360	2,035	2,308	13,930
ELKH-CLOUD	1,104	4	4	4	1,300	2,052	4,468
EODC	4	4	4	4	4	4	24
GRNET-OPENSTACK	2,520	2,158	2,185	2,249	1,860	2,108	13,080
IFCA-LCG2	7,414	2,679	2,996	1,476	190	141	14,896
IISAS-FedCloud	3,372	3,018	2,252	2,233	1,879	2,122	14,876
ILIFU-UCT	0	0	906	1,524	1,847	2,110	6,387
IN2P3-IRES	16,558	27,180	22,489	4,172	673	500	71,572
INFN-CLOUD-BARI	2,842	1,964	1,637	1,293	1,198	1,771	10,705
INFN-CLOUD-CNAF	3,654	3,445	2,134	2,248	1,810	2,244	15,535
NCG-INGRID-PT	3,530	2,848	2,389	2,324	1,939	2,344	15,374
SCAI	1,905	2,889	2,001	691	1,580	2,013	11,079
TR-FC1-ULAKBIM	4,342	3,807	2,292	2,235	1,881	2,109	16,666
UA-BITP	1,343	251	1,952	114	221	700	4,581
WALTON-CLOUD	3,292	2,902	2,223	1,860	766	1,407	12,450
fedcloud.srce.hr	3,013	0	0	0	0	0	3,013
Total	72,144	65,514	56,963	34,334	27,902	31,985	288,842

Figure 3: Total number of VMs run by each EGI Federation Cloud centre during H2 2024

The framework is already used by GreenDIGIT partners that are also members of the EGI Federation and are contributing to the EGI infrastructure with HTC and Cloud resources.

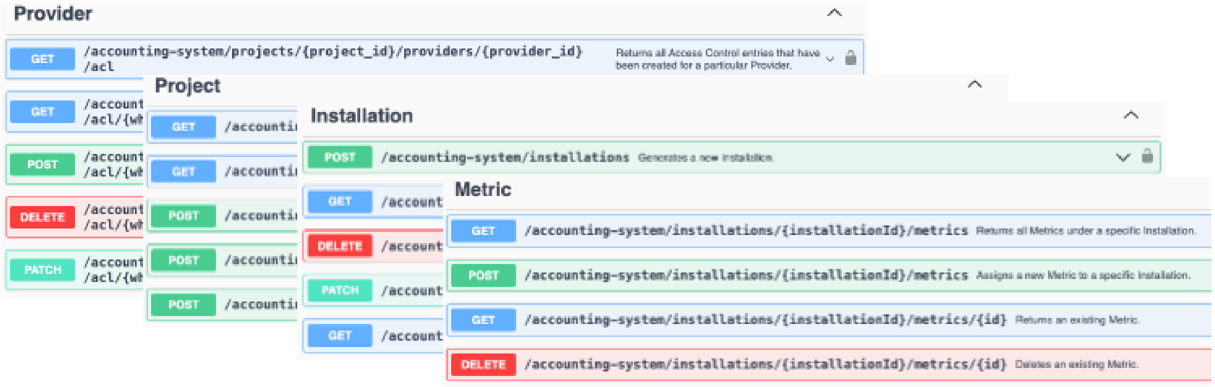
The Grid Usage Record and Cloud Usage Record are very similar to each other, and easily extensible. The consumer is not sensitive to extra attributes added by producers, making it forward-compatible with producers publishing additional consumption attributes, e.g. power, utilization rate, or carbon cost.

3.2.2 EOSC Accounting system for metrics

The EOSC Accounting Service is a platform designed to efficiently collect, aggregate and exchange metrics across various infrastructures, providers and projects, all part of the EOSC EU Node[8].

The main functions of the platform are expressed by a REST API (Figure 4):

- Accept input from several different resources
- Define and support metrics
- Database secure storage and intelligent aggregation of incoming data
- Make aggregated data available to various clients
- Search, filter, offer data for a specific time period.



Provider

GET /accounting-system/projects/{project_id}/providers/{provider_id}/acl Returns all Access Control entries that have been created for a particular Provider.

GET /account /acl/{wh

POST /account /acl/{wh

DELETE /account /acl/{wh

PATCH /account /acl/{wh

Project

GET /accounti

GET /accounti

POST /accounti

POST /accounti

POST /accounti

Installation

POST /accounting-system/installations Generates a new installation.

GET /account

DELETE /account

PATCH /account

GET /account

Metric

GET /accounting-system/installations/{installationId}/metrics Returns all Metrics under a specific Installation.

POST /accounting-system/installations/{installationId}/metrics Assigns a new Metric to a specific Installation.

GET /accounting-system/installations/{installationId}/metrics/{id} Returns an existing Metric.

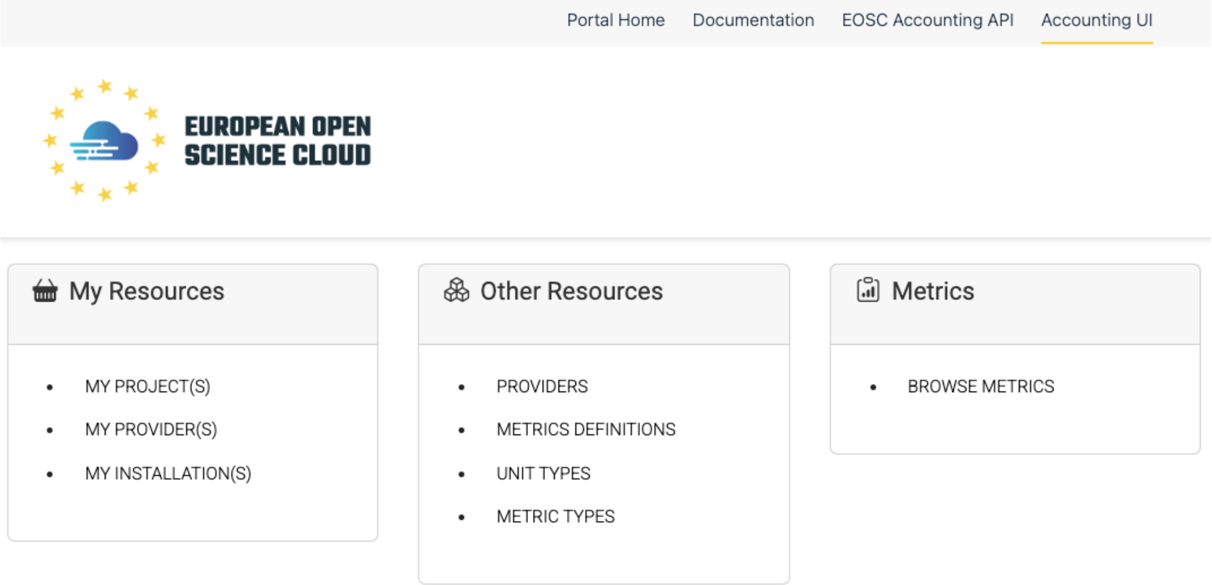
DELETE /accounting-system/installations/{installationId}/metrics/{id} Deletes an existing Metric.

Figure 4: Snapshot of EOSC Accounting system API structure

In addition, API resources must only be obtainable by authenticated clients. For this reason, every client who wants access API resources should be authenticated.

[8] EOSC EU Node: https://open-science-cloud.ec.europa.eu/

The platform also offers an intuitive User Interface[9] (Figure 5) that allows clients to interact effortlessly with the Accounting Service. Users can access and manage accounting data for specific time periods through a user-friendly dashboard.



From the GreenDIGIT perspective, probes and sensors that are deployed at project partner premises could gather the relevant environmental impact related metrics and feed them in the metrics framework at the EOSC Accounting Service level to be consumed via API by a broad category of readers (platforms, tools, brokers, web pages).

The framework is used by the GreenDIGIT partners that are also contributing with services to the EOSC EU Node platform.

[9] EOSC Accounting service: https://ui-acc.open-science-cloud.ec.europa.eu/

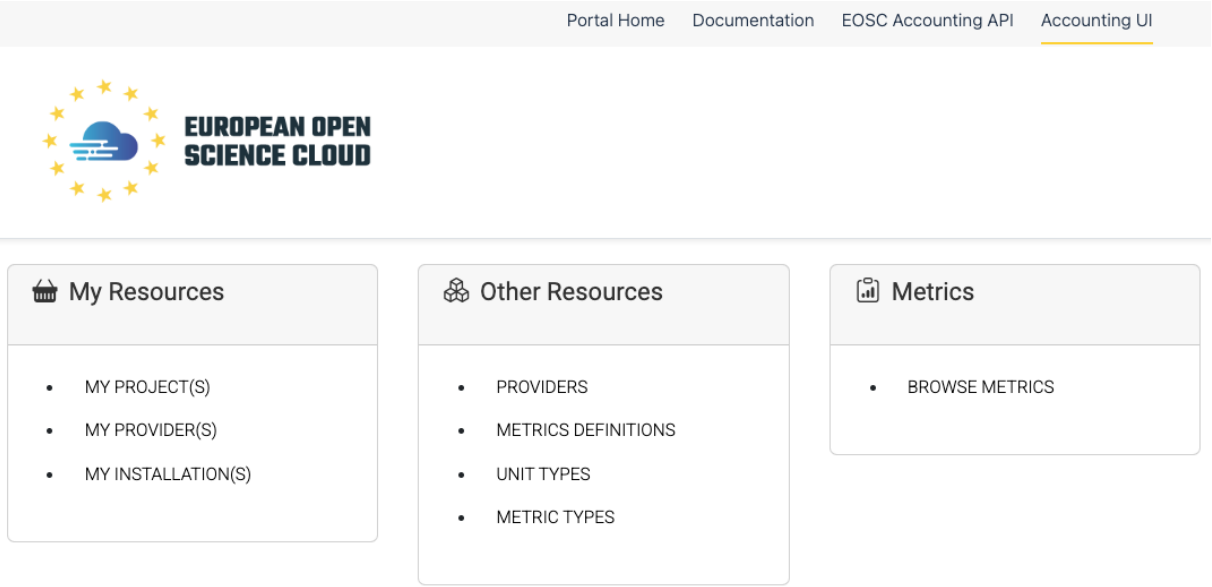


Figure 5: EOSC Accounting service UI

3.2.3 Prometheus

Prometheus[10] is an open-source monitoring and alerting system designed for time-series data. It is widely used for observability in cloud-native environments, especially with Kubernetes. Prometheus is already used for local metrics collection purposes at some of the GreenDIGIT members, including CSIC and SZTAKI.

Key features of Prometheus are:

- Time-Series Database (TSDB) – Stores numerical data over time, indexed by labels
- Pull-Based Model – Prometheus scrapes metrics from instrumented services instead of waiting for them to push data
- PromQL (Prometheus Query Language) – Enables powerful queries and analysis of time-series data
- Multi-Dimensional Data Model – Uses labels (key-value pairs) to categorize and filter metrics
- Alerting & Visualization – Works with Alertmanager for notifications and integrates with Grafana for graphical dashboards
- Service Discovery – Dynamically detects targets using Kubernetes, Consul, or static configurations

The typical use cases where Prometheus is applied are:

- Infrastructure monitoring (CPU, memory, disk usage)
- Tracking application performance (API response times, request counts)
- Observability in microservices and Kubernetes clusters
- Generating alerts for anomalies (e.g. high error rates)

Some of the most common Prometheus metrics are:

- Counter – A cumulative metric (e.g. HTTP requests count)
- Gauge – A metric that can go up or down (e.g. memory usage)

[10] Prometheus: https://prometheus.io/

- Histogram – Distributes values into buckets (e.g. response time)
- Summary – Similar to Histogram but includes quantiles

Points of convergence already exist between Prometheus as the consumer service and the accounting system in EGI. There is an exporter service[11], which accepts Cloud Accounting Records in the format used by EGI, and exports them as metrics into Prometheus. It can be operated either at site-level (before the message queue) or at consumer side (after the record is delivered through the message queue).

3.3 Metrics collection and management of Cloud (OpenStack) and AI environments

Since energy consumption is one of the main causes of the environmental impact of RIs, it is necessary to measure and control its use in these facilities. Beyond VMs, in a cloud computing environment, there is a large number of different hardware equipment that needs to be monitored. In addition, to optimize their use, it is necessary to measure them at different levels of abstraction, from the technical room to the VM deployed in the cloud.

3.3.1 Metric collection at site level

At this level of detail, which monitors the energy consumption of the entire infrastructure as a whole, it is necessary to instrument the electrical panels with power meters, and then monitor these sensors.

3.3.2 Metric collection at node level

Servers are the most energy-consuming hardware equipment. Broadly speaking, these have a set of sensors on the motherboard, which are managed by the Baseboard Management Controller (BMC), and can be queried through the Intelligent Platform Management Interface (IPMI). It should be noted that although the communication protocols are standard, the implementation of the BMC and the available sensors vary with the vendor and device model, based on the tests we have performed on a wide variety of hardware and in different data centres. Through this source, we can obtain the power consumption of the entire server, and in most cases broken down by power supply (PSU), fans, CPU, and memory.

Using this approach, it is possible to obtain the total energy consumption per server, but in order to use this consumption to get a realistic carbon footprint, we need to correct this value with some correction factor, since for that server to work, more systems are needed, such as cooling, which also consumes a significant percentage of energy. To correct this measure, we can use a very well-known metric such as PUE[12], since it relates the total consumption of the data centre with what is consumed by the IT equipment, which we can consider practically the sum of the power consumption of the compute servers.

Another common way to measure power consumption, used by most existing software tools for this purpose, is to use Running Average Power Limit (RAPL)[13]. This is a feature of Intel/AMD's x86 CPUs,

[11] https://github.com/goat-project/exporter
[12] https://datacenters.lbl.gov/sites/default/files/isc13_tuepaper.pdf
[13] https://hubblo-org.github.io/scaphandre-documentation/explanations/rapl-domains.html

manufactured after 2012, which allows setting limits on the power used by the CPU and other nearby components. Also allows, as feedback, to report the cumulative power consumption of various system-on-chip (SoC) power domains. According to existing documentation, there are the following domains: Core/PP0, which encompasses the power consumed by the aggregate of CPU cores. Uncore/PP1: power consumed by components close to the CPU cores, which in most cases refers to the integrated GPU chipset. DRAM: power consumed by the memories. Package/PKG: includes “Core” and “Uncore”. In some documentation and some of our experiments, it seems to include “DRAM”, but this does not seem to be true in all cases. Finally, the specification also establishes a last domain, PSys, which measures the power consumption of the whole SoC, but in our tests, we have not found any CPU model that supports this domain, nor does the documentation detail what other components are taken into account, other than those already mentioned above in other domains. The RAPL power data is exposed to the operating system through the model-specific registers (MSR), read, in a Linux environment, through the power cap sensor, available in a kernel module.

Due to security issues (CVE-2020-8694, CVE-2020-8695 / INTEL-SA-00389) it is not always possible in some cases to read these registers, or the values are not entirely accurate. The vulnerability exploits what are known as power-side channel attacks, which exploit the unprivileged access to RAPL Interface [1]. An attack called Platypus [2] can recover cryptographic keys from Intel Software Guard Extension (SGX) and the Linux Kernel, inferring such data from the CPU power consumption when computing them. To protect against such attacks there are several fixes, both software and hardware. First, the Linux kernel was updated as of v5.10 to make the registers accessible only to privileged users, but this update does not protect against a privileged attacker. Therefore, CPU manufacturers also acted: On the one hand, Intel released directly a microcode update[14], so that when SGX is enabled, the measurement read from the CPU voltage regulators is filtered before being written to the registers. On the other hand, AMD disabled support for the power cap sensor in the Linux kernel, and after some patches by Google to the power cap sensor[15], the registers became available again for the 17h family of AMD processors as of kernel version 5.8, and as of version 5.11 for the 19h family[16].

Finally, it should be considered that this last measurement method does not include all the equipment of the machine, and consequently, other hardware that is highly demanding in terms of power consumption, such as external GPUs, will not be taken into account. Therefore, when calculating the carbon footprint of a machine using this approach, it will be necessary to adjust the measurement value obtained by a corrective factor, to obtain a realistic power consumption and get a real carbon footprint at the end. This correction factor can be the ratio between the energy consumed by the server and the energy consumed by the CPU, or as defined by Patterson: IT Usage Effectiveness (ITUE) [3], or at data centre level, the energy consumed by the data centre divided by the energy consumed by the CPU, defined by Patterson as: Total Usage Effectiveness (TUE).

3.3.3 Metric collection at VM level

Finally, at this level of detail, we focus on the energy consumption of the main asset used by users and the platforms deployed on them in a cloud computing environment: VMs. This level of technicality is

[14] https://www.intel.com/content/www/us/en/developer/articles/technical/software-security-guidance/advisory-guidance/running-average-power-limit-energy-reporting.html
[15] https://www.phoronix.com/news/Google-Zen-RAPL-PowerCap
[16] https://www.phoronix.com/news/AMD-RAPL-Linux-Now-19h

more complex than the one described above since it is a virtual resource, which runs on a physical resource but is shared due to virtualization. Furthermore, there is no single alternative since we are introducing a new element into the system stack, such as hypervisors. The two major open-source virtualization platforms, KVM/QEMU and Xen Project take different approaches to address this problem. Starting with the latter, the Xen middleware exposes RAPL model-specific registers directly to the dom0 (host), and then this is replicated to all domUs (VMs), but based on our tests, it does not proportionally adjust the value based on the usage a VM makes of the hardware. This also leads to a security and isolation problem. Some publications like "Enabling power-awareness for the Xen hypervisor" [4] propose adding power-awareness capabilities to the Xen dom0 to share the energy consumption of each machine proportionally to the hardware usage. On the other hand, when virtualizing with KVM/QEMU, the VM does not have access to the RAPL registers, since this VM is just another process in the set of processes of the host operating system. For this purpose, there are already tools that allow the energy consumption per process to be calculated from the host machine. These types of tools are based on the fact that a multitasking CPU does timesharing, being able to execute only one process in each jiffy. Therefore, we can measure CPU consumption only when the process is running, counting the jiffies that said process uses. A tool that is based on this methodology is Scaphandre[17], an open-source metrology agent dedicated to electric power and energy consumption metrics created by Hubblo, which has to be deployed on every host machine. In addition, it has been specifically designed to monitor the power consumption of VMs and containers running on a host machine, since it is capable of sharing said measurements with the VM or container itself[18], through a hypervisor-shared file system in memory (virtiofs[19]), and this measurement can be read by a Scaphandre agent deployed inside the VM as if it were a physical machine. Virtiofs is a shared file system purposed for virtualization, it is specifically designed to take advantage of the locality of VMs and the hypervisor, using memory-shared pages.

With Scaphandre, obtaining the power consumption of a VM is easy, but sharing that measurement across a variable multi-tenant computing environment (i.e., a cloud infrastructure operated by OpenStack) is non-trivial. First, since virtiofs is included in libvirt v6.2.0, it requires that all operating systems on both host machines and VMs be upgraded to at least version 21.04 (Hirsute Hippo) in the case of Ubuntu. Also, libvirt v6.2.0 requires at least version 22.0.0 of OpenStack Nova. On the management side, libvirt v6.2.0 requires at least OpenStack Nova version 22.0.0[20]. Second, there is a lack of automation support in OpenStack to provide a feature in OpenStack Nova to manage virtiofs mounts, between VM and host. To create the shared filesystem, it is necessary to modify the XML definition of the KVM/QEMU VM. Nova fully manages this file, and only Nova can change it. There are some nova-specs proposing changes to support this feature [21].

As previously mentioned, and since Scaphandre in its 1.0.0 version only obtains measurements from RAPL registers[22], the VM's energy consumption does not include consumption from other specific hardware such as network cards or GPUs. Therefore, it is also necessary to apply a correction factor,

[17] https://hubblo-org.github.io/scaphandre-documentation/

[18] https://hubblo-org.github.io/scaphandre-documentation/explanations/how-scaph-computes-per-process-power-consumption.html

[19] https://libvirt.org/kbase/virtiofs.html

[20] https://docs.openstack.org/nova/latest/reference/libvirt-distro-support-matrix.html

[21] https://specs.openstack.org/openstack/nova-specs/specs/2023.1/approved/virtiofs-scaphandre.html

[22] https://hubblo-org.github.io/scaphandre-documentation/explanations/host_metrics.html

the same as explained above for servers using RAPL as a measurement source, to obtain a real carbon footprint.

Finally, it is also possible to obtain the energy consumption and utilization of an NVIDIA GPU, since these measurements are available in the card driver, and can be read with CLI tools such as nvidia-smi or through an API using one of the management services oriented to data centres such as NVIDIA DCGM[23]

3.4 Metrics collection and management of High Throughput Compute environments

Environmental sustainability is a strategic area of priority for many RIs that provide High Throughput Compute (HTC) environments and for their funding agencies. WLCG[24] is one such international collaboration that is driving several activities aiming to estimate the energy consumption needed by the computing resources used by the LHC[25] experiments and also to help partners reduce the carbon footprint of their computing needs. Several GreenDIGIT partners are also WLCG participants, therefore involved in the above mentioned activities (e.g. CNRS, CSIC, CESNET).

The survey on RIs landscape review run by GreenDIGIT in 2024 highlighted that the energy has been considered by the respondents to have the primary environment impact, with water coming second. Therefore, the initial focus will be the capability to record and track the energy related environmental impact of HTC services and applications over time by choosing the right metrics that capture these aspects.

While the possible frameworks to collect and aggregate these metrics are discussed in Chapter 5, this section briefly describes the methodology on metrics implementation and reporting suitable for the HTC computing resources.

3.4.1 Metrics at site level

Sites providing HTC services have similar hardware configurations to sites providing cloud services. Therefore, similar metrics are expected to be collected (see Chapter 4 for a comprehensive list).

One common metric is the **PUE** used to determine the energy efficiency of a data centre. While many sites are reporting PUE values, for missing sites a default value could be assumed until sites can update it to demonstrate improvement. The idea is to reward sites that make an effort to improve the infrastructure to be more environmentally friendly, while not penalising the ones that have no immediate means to do that. This is not just for PUE but also and more generally for carbon emissions. The more important aspect is to measure improvements, rather than absolute values that might be difficult to interpret.

Along with power consumption, HTC sites are also measuring or expected to measure their carbon footprint (***CFP***). This is a more complex endeavour since it has to encompass all Scope 1, 2 and 3

[23] https://developer.nvidia.com/dcgm
[24] https://home.cern/science/computing/grid
[25] https://home.cern/science/accelerators/large-hadron-collider

emissions[26], but initially only scope 2 (mainly the operations) will be part of the measurement and monitoring.

With this simplification in place an initial formula for ***CFP*** is ***E * U * C***, where:

- ***E*** is the total energy used (kWh) by site
- ***U*** is the share of the HW (%) used for a specific workflow (parallel workflows at a time are assumed)
- ***C*** is the Carbon content (kg/kWh) in energy provided.

3.4.2 Metrics at worker node level

A worker node is the basic computing unit responsible for executing individual tasks or jobs submitted to the HTC service.

Historically the HTC specific payloads were submitted as single core workloads and while nowadays this changed, with several cores allocated per job, the most useful related metric is the ***power consumption per CPU core***. Sites are expected to report this metric either as a single value, or as a list depending on the hardware involved. Alternatively, for non-reporting sites a default value could be provided, and this could act as an incentive for sites to update the numbers every time they improve.

Ongoing work and investigations are taking place on how job-based measurements can be done and validated. For example the average ***power consumption per HS23***[27] can be estimated based on the hardware inventory at sites and the workload reported. Another possibility would consist of using the batch system capabilities to report back energy consumption and report it further upstream in the accounting tools, but this approach is in its early stages.

All these metrics that characterise consumption by particular payloads can be evaluated by tools described in Section 6.2 and then collected by pilot jobs submitted by DIRAC workload manager to report consumption of power and other resources per executed user task (see Section 14.4).

3.4.3 Metrics at service level

The systems and services used for scheduling jobs to the HTC clusters can be also subject of power consumption monitoring and reporting. For example the EGI Workload Manager[28] service (based on DIRAC WMS – section 14.4) consists of a number of VMs (10 mid-range virtual servers) running various components of the service. The service uses also database servers hosted in the same computing centre. The load of the service can vary in a large range depending on the user's activity. This load and the corresponding power consumption are subject to a dedicated monitoring and optimization using tools described in section 6.2. In particular, the next generation of the DIRAC services that will be used for the GreenDIGIT orchestration demonstrator will be deployed using a Kubernetes infrastructure. This will allow to allocate resources dynamically as required by the current loads and not assuming peak loads as it is done now. This will require measurements of the system loads and elaborating algorithms for load prediction and the corresponding resources allocation.

[26] https://www.carbonneutral.com/news/scope-1-2-3-emissions-explained
[27] HEPSCore: https://cds.cern.ch/record/2879939/files/document.pdf
[28] https://www.egi.eu/service/workload-manager/

3.5 Energy carbon content collection

3.5.1 Carbon intensity of electricity

In addition to basic parameters such as consumption and price, carbon intensity is essential for data centres and their power consumption. This parameter is driven by both grid regulation (production vs consumption) and the fact that renewable energy is highly dependent on weather conditions. To help with the regulation requirements on the consumption side, it can be used flexibly, with the ability to control or shift load. The power grids are operated by Transmission System Operators (TSOs), which are responsible for the stable and efficient use of currently available resources.

Carbon intensity of electricity is referred to as Operating Emissions Rate which can be in general a metric formed from mix of pollutants (Green House Gases (GHG) intensity) not only the CO_2. We distinguish two types of rates:

- Marginal rate – based on the emissions of power plant needed in response to a change in consumption, serving as a signal for load shifting and indicates how much can be avoided.
- Average rate – an average of emissions of all power plants needed to cover current demand. It can describe the current carbon impact of a particular data centre or workload.

The use of marginal carbon intensity as a signal for load shifting and avoided emissions reporting is attractive, but controversial due to issues of accountability, accuracy, and reliability. Therefore, more complex life cycle tracking methods are proposed and simple and easy to understand load shifting signals are used (solar + wind ratio in the current power mix).

3.5.2 Carbon intensity data availability

The current carbon intensity of electricity is not a value generally available, compared to price or production mix. Although in some countries the data is available directly from the TSO, in most it must be calculated from the actual production mix. To the calculation enters typically estimates (carbon intensity of individual fuels and embedded emissions) and methodical decisions (reflecting imports/exports and counting batteries and pumped-storage hydroelectricity). There are 3 levels of data sources. Individual TSOs, European Association of TSOs (ENTSO-E) and aggregation platforms focused on providing data for reporting emissions and load shifting.

Current generation mix is available through APIs of individual TSOs or through the ENTSO-E, where all data is available in full granularity and frequency. In addition, the ENTSO-E provides historical data and even forecasts, all free of charge. To calculate the carbon intensity of the generation mix, we need emission factors for each source. A freely available reference table can be found in the contributions section of the well-known aggregator platform, ElectricityMaps.[29] Well curated data (current and forecasted carbon intensity) are available directly from aggregator platforms (Electricity Maps, WattTime[30], OpenClimateFix), but are typically commercially available or incomplete.

In terms of data granularity and frequency, the generally available data described above covers regions in Europe that are typically equivalent to countries and is provided on an hourly basis. Datasets are

[29] https://github.com/electricitymaps/electricitymaps-contrib/wiki/Default-emission-factors

[30] https://watttime.org/wp-content/uploads/2023/11/WattTime-MOER-modeling-20221004.pdf

available, for example from the European Environment Agency (EEA), that cover historical data with granularity such as years.

Standardization efforts are underway, in particular by the Linux Foundation's Carbon Data Specification Consortium (CDSC). The Green Software Foundation (GSF) provides the Carbon Aware SDK, an abstraction layer that decouples applications from data sources.

3.6 Energy data model and format

The existing energy monitoring tools, such as Scaphandre [31], PowerAPI [32], or Kepler [33], generally implement a standard approach. First, the data is collected by a software or hardware probe, second, the collected metrics are stored in an energy database, and third, the stored data is reported to a user through monitoring tools. To facilitate industrial adoption, such tools are designed to integrate into existing monitoring systems, such as Prometheus for storage and Grafana for reporting.

To keep track of usage history, energy metrics are stored as time-series of the power usage in Watts, or in Ampere-hours per second if the device is battery-powered. The energy data of a system can be implemented as a set of time-series rather than a single one, to provide a finer granularity. For instance, each core of a CPU can monitored independently. As a consequence, simultaneously monitoring different granularity levels of a given system may lead to redundancy in measures. For instance, the energy used by a given core of a CPU will appear in the time-series of this core, of the CPU it belongs to, of its server, and of its infrastructure. Thus, the exact scope that the time-series represent must also be stored, to clarify how such energy data can be aggregated. In addition, details such as the methodology used to obtain such data are relevant to store the data, and to quantify the level of confidence in said data. For instance, modelled power values are not as trustworthy as measured power values, and may not be considered as a ground truth for further research.

In addition to energy metrics, the Green Software Measurement Model (GSMM) [5] recommends to monitor usage activity, such as the CPU, GPU or RAM usage, expressed as a percentage, or the amount of data permanently stored or transmitted over network. Additional metrics, such as the amount of useful work produced, allow for quantifying the efficiency of the monitored system as the useful work produced per unit of energy used. While GSMM proposes a framework to unify the setup, tools, and procedures regarding energy measure, it does not propose a standardized data model for energy measures.

[31] https://github.com/hubblo-org/scaphandre
[32] https://powerapi.org/
[33] https://github.com/sustainable-computing-io/kepler

4 Scheduling Approaches for Green Optimisation of Scientific Workflow

4.1 Motivation and previous work

Power-grid aware scheduling is based on a long-term effort to increase efficiency in computing infrastructures. In resource-aware job scheduling, an optimal match between job characteristics (software, data) and available resources (CPU, RAM, GPU) is searched. The basic idea is to add an element describing the consumption view to this concept. Existing work is typically based on power capping, optimising workload placement between nodes (keeping below the level at which modern CPUs are most effective), and automatically powering down nodes [6].

4.2 Carbon-aware optimisation

An important consideration is the criteria used to optimize. In addition to power consumption and its price, the current carbon intensity of the regional electricity and the demand for power-grid regulation are added. In the case of waste heat reuse, the current heat demand can also be an important input. Flexibility is the ability to manage consumption over time based on demand. It is cited as one of the solutions to energy and climate challenges and is based on smart grid concepts that have been developed over the years [7]. An emerging term to describe these concepts in software design and implementation is carbon-aware. The GSF represents an industry effort, led by Microsoft, focused on software that runs on personal computers. Notable examples include carbon-aware distribution of updates and carbon-aware scheduling of operating system tasks (including a carbon-aware browser) [8].

For large distributed infrastructures (both public clouds and RIs), time and geographic load planning according to current conditions with a focus on reliability and availability is an essential function. In 2023, Google published its carbon-based load management platform [9], CERN held a workshop focused on the topic in 2024 [5]. Google is active in popularizing the benefits in case of extreme events (weather, energy crisis) [6]. Existing works in scheduling are typically focused on use cases related to Machine Learning (ML) loads as [10] . A fundamental factor is the design of scientific workflows, which is usually not in the hands of the research infrastructure. Involvement of scientific users can have a major impact on the efficiency of optimization, specifically for scheduling this can be information about the nature of the tasks in terms of energy consumption (CPU vs IO bound) and priority. Work towards addressing this issue focuses not only on reporting and feedback, but also on the division of responsibilities between infrastructure and scientific users [11] .

4.3 Energy-Efficient Resource Management and Workflow Scheduling in Cloud and Data Centres

A number of strategies have been investigated in both industry and academia to lower energy consumption in cloud and data centre environments. These methods fall under the following general categories:

1. Resource Scheduling using Heuristics and Rules; The application of rule-based and heuristic-based resource allocation is one of the classic resource allocation methods. In determining when to up or downscale resources, such systems apply predetermined rules, i.e., CPU utilization over a given

percentage threshold. Example: By consolidating workloads onto fewer active servers and favouring energy conservation, Power-Aware Best-Fit Decreasing (PABFD) is more sophisticated than classic bin-packing approaches.
Limitations: These methods are easy to use and quick to apply but not easily adaptable to different workloads.

2. Optimization Based on Reinforcement Learning (RL), in which an AI agent learns an appropriate scaling policy through interaction with the system, has been investigated recently for cloud resource management. Agents that dynamically distribute resources in response to real-time workload fluctuations have been trained using Deep Q-Networks (DQN) and Proximal Policy Optimization (PPO).
Benefits: By continuously adjusting to demand and reducing wasteful energy use, it can outperform rule-based approaches.
Difficulties: Needs a lot of processing power and training data to achieve convergence.
3. Consolidation of Energy-Aware VMs; By strategically placing VMs on fewer real machines, VM consolidation strategies seek to reduce the number of active servers. These approaches are based on moving workloads from unproductive servers to other servers and shutting down the idle ones is known as "live migration of VMs." Dynamic voltage and frequency scaling (DVFS) is a technique for dynamically modifying processor frequency in order to lower power consumption. One such method is Google's Borg System, which effectively distributes workloads throughout data centres while implementing energy-conscious regulations.
4. Using Edge Computing and Fog Computing to Reduce Power Consumption Edge computing delegates particular workloads to devices that are in proximity to users (for example, local edge nodes or IoT gateways (GWs) rather than keeping all processing in large data centres. This saves energy expenses by keeping use of the cloud to a minimum and reducing distant transport of data. As a case in point, fog computing reduces use of large data centres by shifting workloads to neighbouring resources.
Challenges: Effective allocation of tasks demands strong orchestration strategies.
5. Function-as-a-Service (FaaS) and Serverless Models Only when a function is called is serverless computing dynamically provisioning resources. This realizes huge energy savings in that it avoids having to constantly maintain servers in a live status. Examples of these are Google Cloud Functions and AWS Lambda.
Benefits: Workloads scale automatically without VM management needs. Cons: Unsuitable for all applications, especially those that need to process in a stateful manner.

4.4 Task scheduling for HTC/Cloud/HPC computing resources

The most resource consuming part of computations in large RIs is performed using HTC (computational grids), Cloud and HPC resources. Workflows typically consist of producing modelling data (Monte-Carlo simulations, digital twins) as well as processing large volumes of modelling data and data acquired in scientific experiments. The processed data is usually geographically distributed over multiple data centres to optimize the use of computing resources and avoid potential data losses. Treatment of large data volumes is done by execution of a large number of tasks (jobs) that are usually grouped together to form production campaigns. One production is processing a large portion of data of the same type applying the same processing algorithms.

The task schedulers are the services that are responsible for the massive task placement and execution in the distributed heterogeneous computing environments. The primary goal is to optimize the task placement for the fastest turnaround and to make the most efficient usage of various computing resources. However, other optimization criteria can be included into scheduling algorithms.

The DIRAC Workload Management System (WMS) is a smart task scheduler in a distributed computing environment [12]. It was developed for the LHCb experiment at CERN [10] and is now used by multiple scientific communities around the world. In particular the EGI Workload Manager service is based on the DIRAC software [13]. The scheduler architecture is based on the concept of pilot jobs and consists of the following components:

- a registry of available sites where their metrics and connection details are collected;
- a central Task Queue service to which the user jobs are submitted;
- pilot factories specific for each computing resource type;
- pilot jobs running on worker nodes and forming together with the central Task Queue a scalable distributed WMS.

Pilot jobs are sent to various computing centres using appropriate protocols and credentials. They make reservations of computing resources (slots in batch systems or VMs in the cloud sites) that will be available for the scheduler service. Pilot jobs collect a precise description of their respective computing resources and present it to the Matcher service of the central Task Queue which is choosing one of the waiting jobs suitable for execution on the worker node. The pilot job receives the user task and steers its execution on the worker node, reports its status and uploads results to storage systems where they are made available for the users.

The DIRAC WMS is well suited for the optimization of massive computational workflows to minimize the overall power consumption and to take into account other metrics characterizing computing resources at various sites:

1. The pilot jobs make resource reservations with knowledge of user task properties currently available in the central Task Queue. Therefore, the reservations are made in sites which have currently a spare capacity and are most optimal for task execution. For example, data processing tasks trigger resource reservations on sites close to storage systems hosting the required data. This minimizes the use of network resources for data transfers. Other site metrics defined by the GreenDIGIT project as well as more detailed job characteristics can be taken into account. The central Task Queue offers a global view of all the user payloads and allows for general optimization of the resource reservation for multiple user communities and multiple heterogeneous sites served by the WMS services.
2. The pilot jobs prepare and validate the execution environment before the actual job starts. This considerably reduces the user job failure rate preventing the waste of resources and energy.
3. The pilot jobs steer the user job execution on the worker nodes and monitor the exact consumption of CPU, memory, input/output traffic of the job processes. These data are communicated back to users which allows them to precisely describe the properties of subsequent similar jobs which in turn will result in a more precise scheduling decision.
4. Pilots can detect malfunctioning jobs, e.g. stalled or overusing memory and CPU allocations, and prevents the waste of resources and reducing useless power consumption.
5. The pilots can communicate with the hosting environment and collect detailed information about the worker nodes which can be used later while making resource reservations with better accuracy.
6. In cases where pilots can interact with the hosting environment, they can trigger certain actions to minimize power consumption, for example, by reducing the CPU frequency while the user is performing input/output operations.

DIRAC WMS registers resources consumption data, e.g. CPU, data transfers and many others in its Accounting subsystem. Other accountable data can be defined and registered, e.g. power-aware resources consumption metrics. This allows for monitoring and evaluating the effect of power-awareness augmented scheduling algorithms.

5 Metrics for Energy Efficiency of Network and 5G Infrastructure

5.1 Metrics collection and management of IoT environments

5.1.1 Trends in Energy Efficiency and Metrics Collection in IoT Environment

This section contains an executive summary of the trends, available solutions and current considerations, that are given in terms of IoT energy consumption optimisation.

During the past years, the need for interconnected IoT devices was strongly intensified, which also triggered exponential increments in the user data demands. [14] In such a way, escalated interest in IoT energy efficiency increased in the research community as well. With IoT devices deployed across diverse sectors such as smart cities, healthcare, vehicular networks and industrial automation, the cumulative energy footprint is becoming a critical concern. It is worth noting that there is a huge variation in the characteristics and capabilities of IoT devices, indicatively ranging from simplistic sensors (e.g. temperature, humidity), smart devices (e.g. watches) to high performance telecommunication devices (e.g. IEEE 802.11ah, LoRa, NB-IoT). Thus, there are already approaches for IoT devices / networks, that intend to lower the power consumption and improve the energy efficiency along different parts of the continuum (IoT Sensor Network, Far Edge, Near Edge and Cloud), as depicted in Figure 6: GreenDIGIT high-level architecture for optimisation.

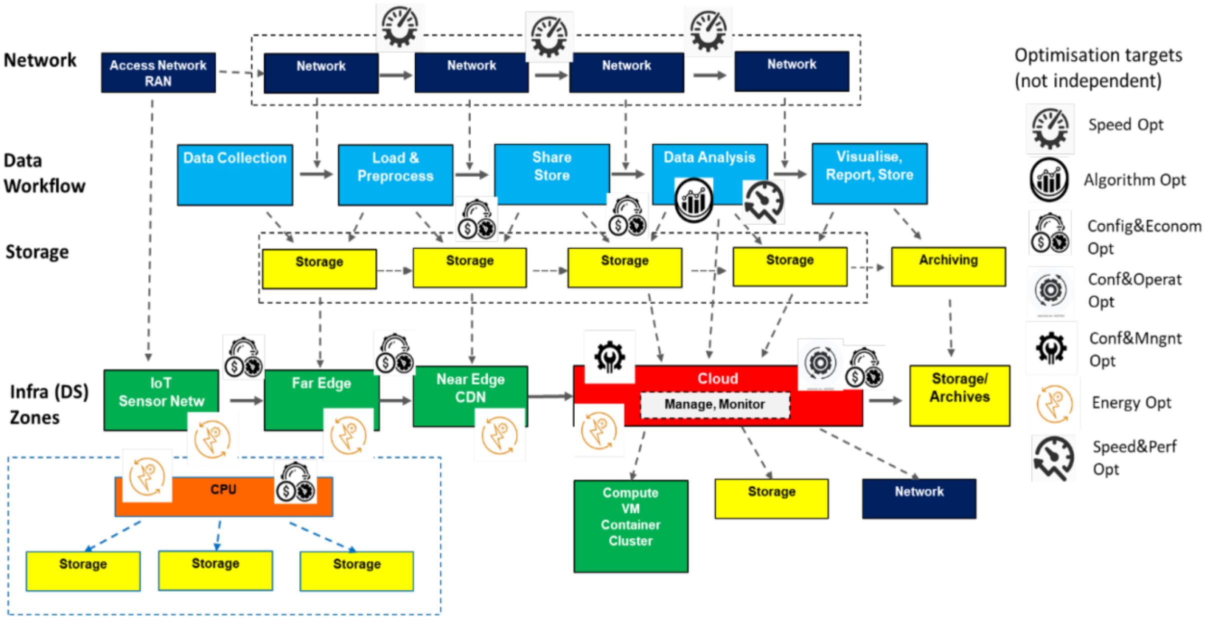


Figure 6: GreenDIGIT high-level architecture for optimisation

Energy-efficient communication protocols like Low Power Wide Area Network (LPWAN), e.g. IEEE 802.11ah, LoRa and NB-IoT, are gaining traction for long-range, low-power connectivity. Specifically, the potential optimisations in such networks can be divided into 4 major categories across the continuum mentioned above.

IoT Sensor Network (Device Layer - Things Layer)

This layer consists of the actual IoT hardware, which is deployed to create the device/ sensor networks. As mentioned above, IoT devices span multiple categories from simplistic sensors (e.g. environmental, motion, industrial sensors and smart wearables) to advanced communication & network devices (e.g. short/long range telecommunication transceivers). The former hardware category is mainly utilized to collect measurements, and the latter to deploy networks and exchange information (data, applications etc.). Additionally, this layer contains the hardware that is used to monitor useful metrics of the IoT ecosystem (e.g. smart meters for energy consumption).

Far Edge (Edge Computing Layer)

This part of the continuum typically contains companion devices like Single-Board Computers (Raspberry PI, Nvidia Jetson, Intel NUC etcetera), that are utilized to host IoT network devices and/or sensors mentioned in the previous category. Companion devices offer in many cases, enough computational power to locally process measurements gathered and exchanged information among the IoT sensor network. This way, reduced latency and energy consumption optimisations may accrue. Of course, based on the IoT network architecture (Infrastructure, mesh etcetera), which is deployed at each application scenario, these devices may also have different networking roles, such as Access Points (APs), Stations (STAs), Relay nodes and GWs.

Near Edge / CDN (Network Layer)

Following the previous two layers, Near Edge (often via content delivery networks (CDNs) mainly consists of edge GWs, servers and fog computing nodes. These entities are typically placed geographically closer to the user compared to the cloud and provide distributed computing and storage capabilities. The main scope of this layer is to offload computational tasks that can't be executed from Far Edge devices (like Raspberry Pi), while in parallel larger round-trip times to/from Cloud are eliminated. Typical processes that are ported for execution at this point, may include but are not limited to more demanding data analysis compared to those performed at the Far Edge. This layer enhances the performance, scalability, and security of IoT applications by caching data closer to end devices and minimizing in many cases unnecessary communication between IoT endpoints and cloud infrastructure.

Cloud

Finally, this layer involves centralized Cloud infrastructures that intend to execute big data processing, Artificial Intelligence (AI) model training and host long-term data storage. Through the training of complex ML models, Cloud-based orchestration overall IoT continuum can be achieved. This includes techniques such as workload allocation, virtualization/containerization and dynamic resource scaling.

Starting from IoT sensor network design considerations (utilizing low-power IoT devices, harvesting solar energy and reducing transceiver operational times etc), up to sophisticated ML correlated decisions (workload allocation, virtualization etc) on the Cloud side. Below, there is an executive summary for some research approaches correlated with the topics mentioned.

5.1.2 EELAS

Initially, the authors of "EELAS: Energy Efficient and Latency Aware Scheduling of Cloud-Native ML Workloads" [15], aim to optimize the deployment and execution of cloud-native ML workloads across the cloud-to-things continuum by balancing energy efficiency and latency constraints. The key contributions of this research approach can be summarized as follows:

- **Multi-Level Scheduling Algorithm:** Optimally distributes ML workloads across IoT, edge, and cloud resources to reduce energy usage while meeting latency constraints.
- **Minimizing energy consumption:** Achieves cloud-to-things continuum energy consumption minimization when deploying workloads.
- **QoS-Aware Resource Allocation:** Dynamically allocates CPU resources based on real-time demand, ensuring energy-efficient operation without sacrificing performance.

It is worth noting that the scheduling problem is formulated as Integer Linear Programming (ILP) and a less complex heuristic algorithm that allows the efficient allocation of resources within the continuum is being developed. The developed approach is being evaluated through several extensive testbed scenarios. The experimental results highlight performance improvements in energy efficiency of over 41.8 % and compared to the off-the-shelf Kubernetes scheduler.

5.1.3 Study on Energy Consumption Optimization Scheduling for IoT

Following, the authors at "Study on Energy Consumption Optimization Scheduling for Internet of Things" [16] deal with the IoT scheduling energy costs themselves (communication costs). The list of this work's contributions is given below:

- **Energy Loss Optimization Scheduling / Multi-Objective Model:** Develops a mathematical model to balance energy consumption and scheduling efficiency in IoT devices. This is achieved through a multi-objective fuzzy optimization algorithm which simplifies the decision-making process.
- **Idle Time Optimization for IoT Device Scheduling:** The algorithm searches for the idle time of the device and optimizes the device scheduling energy consumption model to reduce the overall energy consumption of the device scheduling in the IoT environment.

The authors consider and formulate the mathematical problem as a constrained multi-objective optimization problem, and they use fuzzy mathematics to solve it. Specifically, they set goals, initially to minimize the total energy consumption in IoT scheduling, by simultaneously minimizing the scheduling time as well. Through Matlab simulations, the authors showcase higher modelling accuracy and better energy saving effects, compared to "traditional" methods examined.

5.1.4 IoT-enabled Smart Energy Management Device (SEMD) for Optimal Scheduling of Distributed Energy Resources (DERs)

Additionally, the authors of "IoT-enabled Smart Energy Management Device for Optimal Scheduling of Distributed Energy Resources" [17] propose the development of a SEMD designed for real-time optimization of DERs in consumer-end energy management. The device operates through three key modules: data preprocessing, forecasting, and optimization. By leveraging ML models (Linear Regression, XGBoost, and LSTM), SEMD provides accurate load demand and energy generation predictions, enabling efficient scheduling of DERs such as rooftop PV, Battery Energy Storage Systems (BESS), and Vehicle-to-Grid (V2G) enabled Electric Vehicles (EVs). The key contribution of this approach can be summarised as follows:

- **Development of SEMD:** A portable IoT-based SEMD with robust software and hardware architecture for real-time energy management is being developed.
- **ML Forecasting:** Integration of ML-based models (LR, XGBoost, LSTM) to enhance energy demand, PV generation, and price forecasting accuracy.
- **Experimental Validation:** Proof-of-Concept testing on real-world energy consumers, demonstrating economic benefits and grid reliability improvements.

The authors consider and formulate the mathematical problem as a Mixed Integer Non-Linear Programming (MINLP) model with the objective of minimizing energy costs for consumers while maximizing the utilization of DERs, such as rooftop PV, BESS, and EVs.

5.1.5 Outline – IoT Correlated Metrics

The current survey on energy consumption trends and practices, reveals that the energy efficiency / optimisation, can be characterized as a complex problem to tackle on the IoT environments. There are available approaches that consider some, or all the parts of the IoT continuum, and by providing different outcomes / monitoring solutions. Some common techniques found on the literature include but are not limited to energy-aware task scheduling, adaptive power management, optimised data transmission strategies and network optimisations. Focusing now more on the pure IoT part of the continuum (IoT Sensor Network & Far Edge), some power and network correlated metrics that can be taken into consideration are the following.

- **IoT Device / Sensor Power Consumption Metrics:** This includes the *average power consumption for idle, receive and transmit states* of the wireless adapter (where applicable to be measured). As the IoT Device / Sensor is attached to the Far Edge Device, through short-range local connections like serial (USB, pins etc.), in most of the cases there is no need to measure the data transmission and processing overhead for transferring the measurements. However, the large variety on the connection types given for the IoT devices (USB, pins, Hat etc.), increases the complexity or makes it impossible in some cases to measure the energy consumption on the device itself with appropriate hardware. In some cases, energy consumption measurement through software, or based on reference values, are the only available solutions.
- **Far Edge Device Power Consumption:** This part includes the power consumed during local data processing and before deciding to port them to the next parts of the continuum (Near Edge and Cloud). Thus, the *local data processing energy consumption (e.g. CPU)* can be measured and noted either through software or hardware components (depending on the device). Additionally, depending on the network connection (wired / wireless), it may be also useful to measure the *energy consumption required to transfer data to Near Edge and/or at the Cloud*, which may help in decision making to forward the data for further processing.

Additionally, some further metrics that are not directly relevant to energy consumption, but may affect it in some ways are the following:

- **Wireless Network Efficiency Metrics:** Specifically, for the IoT protocols, most of them operate under the use of the unlicensed spectrum (some exceptions may include protocols like NB-IoT). Thus, in these wireless environments there is always the case of performance uncertainty, as there is no control both number and the type (same or different protocol coexisting in the same frequencies) of contending wireless devices. Thus, in some cases it is useful to retrieve and capture networking metrics, such as *latency and available throughput* with the neighbour nodes. These measurements may also be useful for the decisioning / scheduling, load balancing, or to be compared with potential user constraints for served applications.

- **Energy harvesting:** There are also cases for the IoT and Far Edge device, in which the use of batteries and solar panels are taken into consideration and applied. Therefore, in such cases, it may be useful to consider and note the amount of *energy that can be collected* at each given time and utilized.

5.2 Metrics collection and management of 5G environments

5.2.1 5G observability framework

This part describes the overall 5G state of the art and possible observability framework to be included in GreenDIGIT Digital Infrastructure.

5G state of the art has shifted its architecture towards containerization, where network services are being decoupled from the infrastructure and are divided into smaller elementary models called microservices or network functions. On one hand, this approach has led to a radically new technological framework offering multi-tenant flexibility well beyond classical software-defined environments; on the other side, it posed more than a concern on environmental sustainability.

Studies have shown that the energy consumption of the infrastructure deployed for 5G is much higher with respect to its initial design [1]-[4]. Also, the energy consumption is a performance index of only the infrastructure layer, and it is not propagated to the upper 5G network providers and vertical applications. This means that the vertical stakeholder does not understand the impact on the resource and energy consumption, while infrastructure providers can easily adopt sophisticated resource and power management strategies. Given this issue, it is essential to integrate observability and monitoring frameworks that propagate Key Performance Indexes to all the stakeholders.

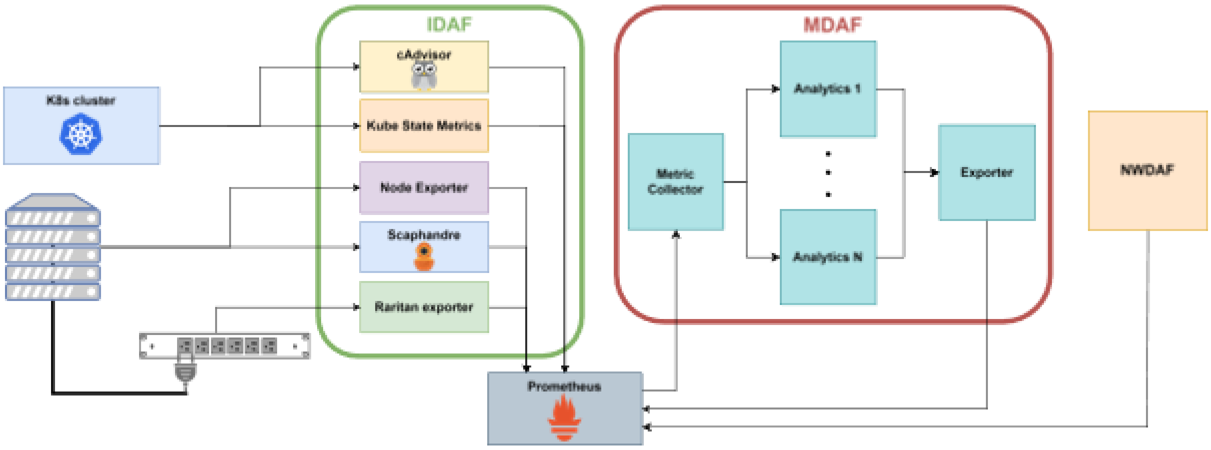


Figure 7:Internal architecture of the observability framework [5][6]

The observability framework is shown in Figure 7. As it can be seen, it is divided into three components:

5.2.1.1 (Infrastructure Data Analytics Function (IDAF)

It’s the module responsible for exposing infrastructure related metrics. As illustrated in Figure 1, the IDAF is composed of five elements that provide metrics in a Prometheus-like format. cAdvisor and Kube State Metrics (KSM) offer insights into the Kubernetes cluster. cAdvisor, which is integrated into the kubelet (the main K8s agent responsible for managing pod deployments and ensuring the health of the K8s cluster), reports on the utilization of computing resources (e.g. CPU, memory, networking, etc.) for each container. KSM, on the other hand, is an additional service that provides information on the state of various K8s objects (e.g. pods, nodes, replica sets, stateful sets, etc.).

NodeExporter and Scaphandre [8] gather metrics related to the bare-metal servers. NodeExporter collects a wide range of hardware and kernel-related metrics, including server resource utilization (e.g. CPU, memory, networking, etc.). Scaphandre, on the other hand, reports on the power consumption of individual processes running on the server, including containers and VMs. It uses Intel's RAPL [9] counters and CPU time spent on each process to calculate power usage. Currently, Scaphandre only tracks CPU power consumption, and in some cases, DRAM controller usage, making it particularly reliable for processes that do not require GPUs, as the CPU is typically the primary power consumer.

Lastly, the Raritan exporter is responsible for reporting the power consumption of Raritan Power Distribution Units (PDUs).

All the above metrics are stored in a Prometheus database, which is accessible by the three functions (i.e., IDAF, MDAF, NWDAF). Prometheus serves as a time-series database (TSDB) that collects metrics, each consisting of a timestamp, value, and optional key-value labels.

5.2.1.2 Management Data Analytics Function (MDAF)

Combines metrics coming from both the infrastructure and the management, allowing to estimate the current and the future carbon/energy footprint induced to the computing and offloading resources in the edge-cloud continuum, and even the availability and the use of green energy sources and hardware offloading engines.

The MDAF is primarily composed of three main components: the metric collector, the analytics blocks (which can include more than one), and the exporter.

The metric collector uses the prometheus-api-client library to query the Prometheus instance and retrieve metrics from Scaphandre and cAdvisor. Its primary purpose is to collect data on the power consumption of each process and the resource utilization of each container. Additionally, it can identify metrics that are not associated with virtualized components—namely, kernel processes essential for the proper functioning of the server and the layers above it, such as the hypervisor and container orchestration platforms. We assume that without these virtualized components (e.g. a containerized 5G network running on K8s), the servers would be inactive. As a result, the power consumption of these kernel processes is considered the "embodied power consumption."

The analytics blocks consist of a set of modular algorithms designed for data processing. These include calculating the embodied power consumption of the virtualized components, enriching the metrics to address Scaphandre's limitation in acquiring container labels for unique identification, and, importantly, mapping the kernel-level power metrics sourced from the IDAF.

Finally, the exporter's purpose is to expose the newly generated metrics in a Prometheus-compatible format. It relies once again on the prometheus-api-client library to ensure compliance with the Prometheus format.

5.2.1.3 Network Data Analytics Function (NWDAF)

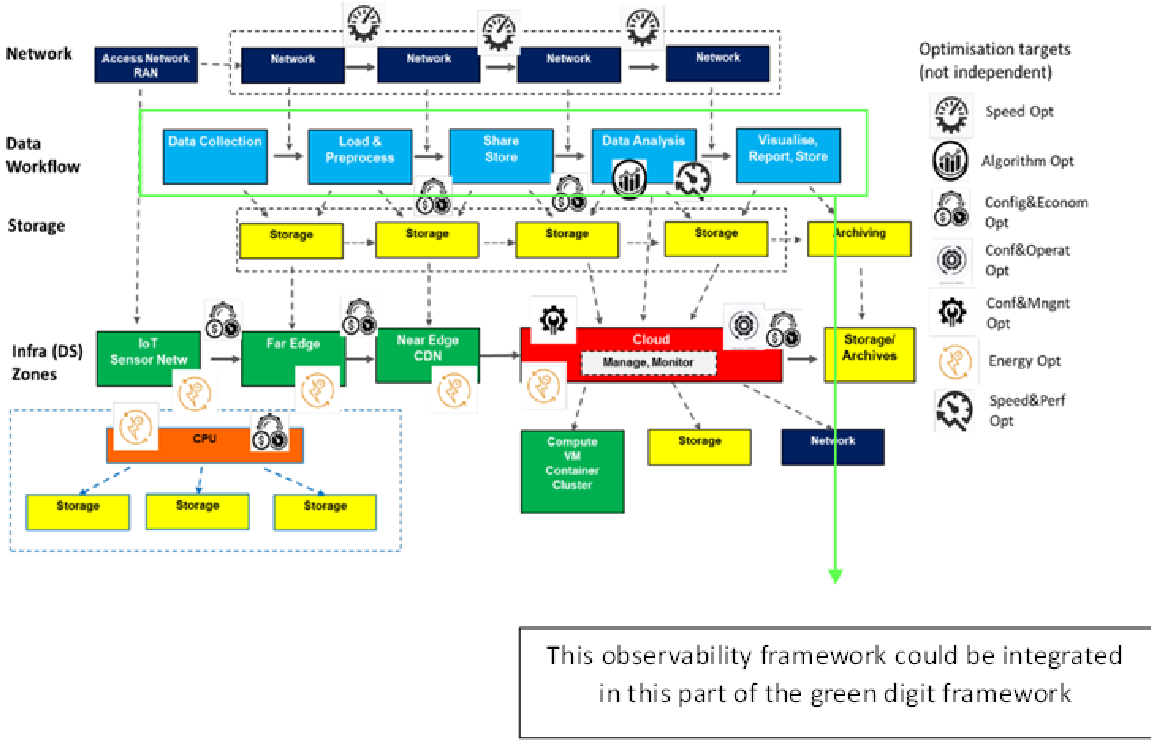
Conceived to acquire monitored data at the edge and the cloud part of the infrastructure and to produce and analyse added-value Key Performance Indicators (KPIs), like forecasts of NF workloads.

5.2.2 Connection to GreenDIGIT architecture

A possible solution is to integrate an observability framework to GreenDIGIT that is similar to the one proposed by 6Green Project [5][6]. This framework allows to:

- retrieve energy consumption of hardware infrastructures
- break down the different contributions from any software instance/ microservice in the 5G infrastructure
- recombine these contributions to estimate the consumption levels ascribable to the virtual resources and artefacts consumed by upper level 5G stakeholders.

Figure 8 provide an illustration where the 6G observability framework can be integrated in GreenDIGIT multi-dimensional infrastructure defining the main data processing, data communication and workflow processing streams.



This observability framework could be integrated in this part of the green digit framework

Figure 8:5G observability framework and potential connection to GreenDIGIT architecture

In conclusion, there are different available practices, approaches and solutions, that are given in the literature and consider the energy consumption monitoring for the IoT to Cloud continuum. The ideal characteristics for the system under development, is to be hardware agnostic, providing thus support and measurements for different IoT vendors.

5.3 Metrics collection and management of mobile environments

The power consumption of mobile software has become a critical research area, as excessive power consumption directly impacts end-users by draining batteries, shortening their lifespan over time, and reducing device usability. Thus, energy inefficiencies can lead users to replace hardware more frequently, causing environmental impacts from both the produced e-waste, and the manufacturing of new hardware. This limitation affects both the personal devices of private individuals, and the fleets of devices managed by institutions, such as devices provided to employees performing field work. Research infrastructure and scientists relying on mobile devices, for instance to study mobile software engineering or wireless networking, are also affected, as their device may be under high workload. This creates a compelling incentive for developers to understand and minimize the power usage of their software.

Improving software power efficiency can be achieved by adopting new practices and habits as part of the development processes [18].

Those new routines include:

- Understanding the requirements and goals of the software before the development phase. This involves anticipating optimal software development options [19][20] and carefully selecting power-efficient third-party libraries, as their power consumption can vary significantly, based on their design [21].
- Testing with environmental impact in mind, on a broader variety of devices to reduce the risk of software obsolescence [22] and identify abnormal consumption patterns during execution.
- Monitoring post-release applications to detect performance variations across environments and address disparities.

However, these power optimizations may conflict with other objectives of the application, such as maintaining performance [23] [24] and security [25], making power efficiency a secondary priority in some cases.

Evaluating the relevance of a power modelling solution requires knowledge of the device's actual power consumption, which serves as the ground truth for validating power model estimations. However, while hardware performance counters are exposed by traditional computer processors [26], they are not included in smartphones' SoC. Thus, the power usage of mobile devices is often assessed at the battery level, either by integrating a power meter between the battery and the device, or by relying on programmatically available battery charge indicators [27]. However, physical measurement of smartphone power consumption is often impractical due to the increasing trend of non-dismountable design, such as batteries being glued to the casing. Therefore, the overall battery charge is largely used to assess or predict the power efficiency of software, such as mobile applications [28], code snippets [29][30], or software design decisions [31]. However, such a metric encompasses the whole power usage of the device and offers only a coarse-grained granularity.

To tackle this limitation, since 2021, some Android devices have been equipped with a per-component power meter, the *On-Device Power Rails Monitor* (ODPM). These power meters are hardware probes placed downstream of the battery, directly before the component. ODPM can monitor the power usage of each of the twelve power rails of the device, with each rail powering one or many components. Available power rails include, for instance, each group of CPU cores, the network components, or the screen. However, this tool is still limited as few devices adopted it, and accurate date are only available in test environment, and not in production monitoring scenarios. For devices not equipped with ODPM, it remains possible to model per-component power-usage.

While developers may be interested in the energy efficiency of their software for optimization purpose, this indicator does not convert directly to environmental impact. The environmental impact of software running on mobile devices must account for its power usage and the electricity-mix of its users, but also on the power usage of networking infrastructure and servers used by this software. To comply with a life-cycle assessment approach, a share of the embodied impact of terminals, the networking infrastructures, and hosting infrastructures used by this software must also be imputed to this software. Such a quantification of environmental impacts allows for reporting results or arbitrating between energy and network optimizations that may shift impacts between the terminals, the network, and the servers, or between different impact categories.[34]

5.4 Metrics collection and management of networks

The draft specification "Energy Metrics For Data Networks"[35] introduces a framework for evaluating and enhancing the energy efficiency of data networks. It emphasizes the growing importance of energy efficiency due to rising energy costs and environmental concerns, highlighting the significant impact of network operations on overall energy consumption. The document proposes standardized metrics to facilitate benchmarking and improvements in network energy performance. Key metrics include:

- **Power Consumption per Data Rate (PCDR):** This metric measures the power consumed relative to the data rate of network equipment, evaluating the energy efficiency of data transmission. The formula for PCDR is:

$$\mathrm{PCDR} = \frac{\text{Power Consumption (W)}}{\text{Transmission Rate (Gbps)}}$$

  Where:

  - **Power Consumption (W):** The average power in watts consumed by the specified link, group of links, or the network during the measurement period.
  - **Transmission Rate (Gbps):** The average data rate in gigabits per second transmitted over the link, group of links, or the network during the same period.

[34] European Commission, Directorate-General for Environment. (2021). Commission Recommendation (EU) 2021/2279 of 15 December 2021 on the Use of the Environmental Footprint Methods to Measure and Communicate the Life Cycle Environmental Performance of Products and Organisations.

[35] https://datatracker.ietf.org/doc/draft-bogdanovic-green-energy-metrics/

For example, if a network link consumes 500 watts of power and has a transmission rate of 10 Gbps, the PCDR would be:

$$\text{PCDR} = \frac{500\,\text{W}}{10\,\text{Gbps}} = 50\,\text{W}/\text{Gbps}$$

- **PUE:** PUE assesses the overall energy efficiency of a data centre or facility, with a value closer to 1 indicating higher efficiency. The formula for PUE is:

$$\text{PUE} = \frac{\text{Total Facility Power}}{\text{IT Equipment Power}}$$

Where:

- **Total Facility Power:** The total power consumed by the data centre facility (including cooling, lighting, and other overhead).
- **IT Equipment Power:** The power consumed by the IT equipment (servers, storage, and network devices).

For instance, if a data centre has a total facility power consumption of 1,000 kW and the IT equipment consumes 700 kW, the PUE would be:

$$\text{PUE} = \frac{1{,}000\,\text{kW}}{700\,\text{kW}} \approx 1.43$$

- **Network Equipment Energy Efficiency (NEEE):** NEEE evaluates the energy efficiency of network equipment based on work output per energy input, benchmarking devices to compare energy performance under specific workloads. The formula for NEEE is:

$$\text{NEEE} = \frac{\text{Useful Work Output}}{\text{Energy Input}}$$

Where:

- **Useful Work Output:** The amount of work performed by the network equipment, such as data processed or transmitted.
- **Energy Input:** The total energy consumed by the network equipment during the measurement period.

For example, if a network device processes 5,000 gigabytes of data while consuming 250 kWh of energy, the NEEE would be:

$$\text{NEEE} = \frac{5{,}000\,\text{GB}}{250\,\text{kWh}} = 20\,\text{GB}/\text{kWh}$$

- **Energy Proportionality Coefficient (EPC):** EPC measures how closely a system's energy consumption scales with its workload, where a value close to 1 indicates power consumption is highly proportional to workload. The formula for EPC is:

$$\text{EPC} = \frac{\text{Power Consumption at Idle}}{\text{Power Consumption at Full Load}}$$

Where:

- **Power Consumption at Idle:** The power consumed by the system when it is idle.
- **Power Consumption at Full Load:** The power consumed by the system when operating at full capacity.

For instance, if a server consumes 100 W at idle and 200 W at full load, the EPC would be:

$$\mathrm{EPC} = \frac{100\,\mathrm{W}}{200\,\mathrm{W}} = 0.5$$

The draft also underscores the importance of standardized testing conditions (STC) to achieve reliable and consistent results. Adhering to STC ensures fairness, objectivity, valid comparisons, and minimizes confounding variables, thereby supporting test validity and reliability. By defining these metrics and guidelines, the document aims to promote energy efficiency in network design and operation, enabling administrators and designers to identify opportunities for energy savings, optimize performance, and to reduce the environmental impact of network operations.

6 Sustainability Aspects in RIs for Experimental Research

6.1 Integrated monitoring systems and resource management

RIs are complex cyber-physical systems composed of multiple interconnected subsystems, including cooling, computing, networking, PDUs, and facility management systems [32]. Each of these subsystems plays a critical role in ensuring the seamless operation of data centres. However, they are typically monitored and managed independently using different tools and teams, leading to significant challenges in achieving integrated optimization and sustainability by addressing energy efficiency aspects in a holistic and multi-factor/multidimensional way.

For instance, tools like Prometheus are used for collecting node-level metrics from physical servers [33], while entirely separate tools manage and monitor VMs, and cooling systems. This siloed approach creates difficulties in aggregating and synchronizing monitoring data across subsystems due to the heterogeneity of data sources and platforms. Key challenges in individual monitoring include:

- Heterogeneous Systems: The diversity of tools and platforms used for monitoring different subsystems creates interoperability issues. These systems often lack standardized protocols for data exchange, complicating efforts to integrate them into a cohesive monitoring framework.
- Data Synchronization: Aggregating real-time metrics from disparate sources is challenging due to variations in data formats, sampling rates, and communication protocols. This lack of synchronization can lead to delays or inaccuracies in decision-making.
- Scalability and Complexity: As RIs grow in size and complexity, managing individual monitoring systems becomes increasingly resource-intensive. Scaling up monitoring solutions without compromising performance or accuracy remains a significant challenge.

To address these challenges, there is a growing need for integrated monitoring overlay systems that aggregate metrics from all subsystems in real-time. Such systems would enable:

- Integrated/multi-factor Optimization: By providing a unified view of all subsystems, operators can identify inefficiencies and optimize resource allocation across the entire infrastructure.
- Real-Time Decision-Making: Integrated systems allow for immediate responses to anomalies or changes in operational conditions, improving resilience. For instance, such systems allow dynamic workload distribution and optimized cooling strategies [34].

6.2 VREs for experiment reproducibility

Reproducibility is a fundamental principle of Scientific Research, ensuring that experiments can be validated, extended, and built upon by other researchers. However, challenges such as inconsistent data management, lack of computational resources, and difficulties in sharing workflows hinder reproducibility. A VRE provides a comprehensive solution by integrating computational tools, data management frameworks, and collaborative platforms into a unified digital workspace.

6.2.1 The role of VREs for experiment reproducibility

A VRE enables scientists to carry out, document, and share their research in a standardised and controlled environment. It offers:

- **Data Management and Provenance management**: VREs ensure that all input data, intermediate results, and final outputs are stored securely and linked through metadata standards like PROV-O (Provenance Ontology), ensuring transparency in data lineage.
- **Standardised Workflows**: throughout systems such as Jupyter Notebooks, Galaxy Workflows, or RStudio integrated into a VRE, users can define and execute analytical pipelines that can be replicated and reused across studies.
- **Collaboration and Version Control**: VREs support versioned repositories for code and datasets, allowing teams to track changes, reproduce analyses, and collaborate effectively through tools like Git and D4Science's workspace versioning.
- **FAIR Compliance**: VREs align with the FAIR (Findable, Accessible, Interoperable, Reusable) principles by providing persistent identifiers, standardised metadata schemas, and access control mechanisms to promote data sharing and reproducibility.
- **Computational Reproducibility**: By providing access to containerised computing resources (e.g. via Docker or Kubernetes), VREs allow researchers to execute analytical workflows in controlled environments, reducing dependency on specific local machine configurations.

6.2.1.1 Notable implementations

D4Science VRE

D4Science is an advanced VRE platform that provides researchers with collaborative environments for data analysis, computational modelling, and experiment reproducibility. It supports various scientific domains, including social sciences, environmental sciences, marine research, and biodiversity studies.

Key features:

- **Cloud-based Execution:** Supports computational workflows using Jupyter, RStudio, and ShinyApps and external Cloud Providers such as Google Cloud Platform.
- **Provenance and Versioning:** Tracks execution history, ensuring that analyses can be reproduced.
- **FAIR-Compliant Data Management:** Integrates structured storage and metadata annotation for transparent data reuse.
- **Virtual Laboratories (VLabs):** Customisable environments tailored to specific research needs.

Galaxy Project VRE

Galaxy is a widely used VRE designed to enable reproducible computational research in life sciences and bioinformatics.

Key features:

- **Workflow Management:** Provides a user-friendly interface for creating and executing complex analysis pipelines.
- **Integrated Data Repositories:** Ensures persistent data storage and structured metadata management.
- **Transparent Reproducibility:** Every computational step is logged, and workflows can be easily re-run by other researchers.
- **Containerized Execution:** Uses Docker and Kubernetes to provide scalable and standardized computational environments.

6.2.1.2 Challenges and future directions

Despite significant advancements, challenges remain in ensuring long-term reproducibility, interoperability, and accessibility across VREs. Future efforts should focus on:

- **Interoperability across platforms:** enhancing standards for cross-platform workflow execution and integrating VREs with EOSC and other global RIs.
- **Sustainable Computing and Energy Efficiency**: as VREs increasingly rely on cloud computing and distributed infrastructures, reducing the carbon footprint of computational workflows is becoming a critical priority.
- **Scalability and Performance**: Optimizing Cloud and HPC resources to efficiently handle large-scale datasets while maintaining responsiveness for real-time collaboration.
- **Persistent Identifiers and Digital Object Linking**: Strengthening reproducibility by ensuring datasets, workflows, and results are persistently citable, retrievable, and reusable across different research environments.

6.2.2 Tools for VRE Energy Efficiency Optimisation and Environmental Sustainability

6.2.2.1 Addressing Energy Efficiency and Environmental Sustainability in VRE Operation

VREs are digital platforms that enable researchers to collaborate, share data, and conduct experiments in a virtualised environment [35]. VREs are increasingly critical for large-scale RIs, although they pose environmental challenges—which GreenDIGIT should address. This section focuses on sustainable VREs, focusing on energy-efficient design approaches, resource optimisation, and the integrations to be made in GreenDIGIT's architecture. The main features of VREs are:

- **Experimentation:** VREs support the design, execution, and analysis of experiments.
- **Sustainability resource optimisation:** VREs can be designed and operated in an energy-efficient manner to minimise their environmental impact.
- **Customisation:** VREs can be customised to meet the specific needs of different research communities.
- **Data Management:** VREs provide tools for storing, managing, and analysing large datasets.

The EGI Foundation [36] is one of the prime examples that have taken steps in this direction with the EGI Notebooks tool[36]. To address the environmental challenges posed by VREs, a range of tools and frameworks have emerged. These can be broadly categorised into metrics and KPI frameworks, which provide standardised ways to measure and understand the carbon intensity of software and infrastructure, and dashboard-like tools, which offer visualisations and real-time insights into energy consumption.

6.2.2.2 Metrics and KPI frameworks and tools

Green Metrics Tool[37] & SCI Specification[38]: This framework defines Software Carbon Intensity (SCI) as a standardised metric (gCO_2eq per function unit) to quantify software carbon emissions. The Green Metrics Tool is a developer-focused tool that implements the SCI specification. It functions by

[36] https://notebooks.egi.eu/hub/
[37] https://sci.greensoftware.foundation/
[38] https://www.green-coding.io/products/green-metrics-tool/

measuring the energy consumed by software during execution and converting it to carbon emissions using location-based or grid-average emission factors. Key components include energy meters, CPU utilisation trackers, and emission factor databases. Its unique characteristic is the focus on standardising software carbon measurement through the SCI specification, enabling comparability across different software systems.

EcoVisor [37]**:** EcoVisor operates as a software layer that virtualises energy systems, allowing applications to dynamically adapt to fluctuating clean energy availability. Functionally, it intercepts application energy requests and optimises them based on real-time clean energy supply data. Key components include energy supply forecasting modules, application energy demand models, and control algorithms that adjust application behaviour. Uniquely, EcoVisor enables carbon-aware application design by providing a software-defined interface to manage energy consumption in response to clean energy variability.

Cloud Carbon Footprint[39]**:** This tool estimates cloud emissions by analysing cloud provider billing and usage data. It functions by connecting to cloud accounts (AWS, GCP, Azure) and processing billing exports to attribute energy consumption and associated emissions to specific cloud services. Key components include data connectors for each cloud provider, emission factor datasets, and aggregation/reporting modules. Its unique aspect is its focus on cloud environments and its provider-agnostic approach, offering a consolidated view of cloud carbon footprints across different platforms.

6.2.2.3 Dashboard tools

CodeCarbon[40]**:** CodeCarbon is a Python library that instruments code to measure energy consumption and estimate carbon emissions. It functions by using software-based power meters to track CPU and GPU energy usage during code execution. Key components include runtime libraries, emission factor databases, and logging/visualisation capabilities. Its unique feature is its code-level integration and ease of use, allowing developers to directly measure and visualise the carbon impact of their code.

Zeus[41]**:** Zeus is a library specifically designed for deep learning energy measurement and optimisation. Functionally, it provides tools to profile the energy consumption of DL workloads and apply optimisation techniques. Key components include energy profilers, optimisation algorithms (e.g. model pruning, quantisation), and reporting dashboards. Uniquely, Zeus focuses on the specific energy challenges of AI/ML, offering targeted solutions for this domain.

Keit[42] (from Aknostic)**:** KEIT monitors Kubernetes clusters to provide insights into the environmental impact of containerised applications. It functions by collecting data from Kubernetes APIs and infrastructure metrics to estimate energy consumption and carbon emissions of workloads running in the cluster. Key components include data collectors, a metrics processing engine, and visualisation dashboards. Its unique aspect is its focus on Kubernetes environments, offering visibility and management of carbon emissions in modern containerised deployments.

[39] https://www.cloudcarbonfootprint.org/
[40] https://codecarbon.io/
[41] https://github.com/ml-energy/zeus
[42] https://aknostic.com/keit/

6.2.3 Outcomes and Recommendations for GreenDIGIT and connection with its architecture

Metrics and KPI Frameworks and Tools

- **Green Metrics Tool & SCI Specification**
    - Outcomes for GreenDIGIT: Standardised measurement of SCI within GreenDIGIT services.
    - Recommendations for GreenDIGIT: Adopt SCI as a core KPI; integrate Green Metrics Tool into software development - Into Notebooks
    - Connection to GreenDIGIT Architecture: Provides metrics for sustainability within the computing layer.
- **EcoVisor**
    - Outcomes for GreenDIGIT: Carbon footprint reduction through dynamic workload scheduling based on renewable energy availability.
    - Recommendations for GreenDIGIT: Investigate integration as an energy management layer; configure for GreenDIGIT energy data and renewable energy feeds. - Into DIRAC
    - Connection to GreenDIGIT Architecture: Energy management within computing and networking layers.
- **Cloud Carbon Footprint**
    - Outcomes for GreenDIGIT: Clear understanding of carbon emissions from GreenDIGIT's cloud resource usage.
    - Recommendations for GreenDIGIT: Deploy to monitor cloud emissions; set up cost centre tagging for accurate attribution.
    - Connection to GreenDIGIT Architecture: Infrastructure level monitoring and reporting for cloud resources.

Dashboard-like Tools

- **CodeCarbon**
    - Outcomes for GreenDIGIT: Researcher awareness of code-level carbon footprint; encourages energy-efficient coding practices.
    - Recommendations for GreenDIGIT: Promote use amongst researchers; provide integration guidance.
    - Connection to GreenDIGIT Architecture: User/application-level tool for carbon monitoring.
- **Zeus**
    - Outcomes for GreenDIGIT: Energy efficiency optimisation for AI/ML workloads within GreenDIGIT.
    - Recommendations for GreenDIGIT: Recommend for AI/ML research; provide specialised support. – Into AI4EOSC
    - Connection to GreenDIGIT Architecture: Specialised profiling for AI/ML workloads within the computing layer.
- **Keit (from Aknostic)**
    - Outcomes for GreenDIGIT: Insights into energy consumption of containerised applications in Kubernetes.
    - Recommendations for GreenDIGIT: Deploy to monitor Kubernetes clusters (if applicable); configure for GreenDIGIT infrastructure.
    - Connection to GreenDIGIT Architecture: Infrastructure level monitoring for Kubernetes deployments.

Considering these two main categories within a VRE approach can effectively promote sustainability within the researcher workflow, abstracting technical complexity whilst delivering actionable and insightful data.

6.3 Experiment Reproducibility Metadata Formats

Experiment reproducibility is a cornerstone of scientific research, ensuring that results are verifiable, transparent, and reusable. Metadata plays a crucial role in this process, as it provides structured descriptions of experimental setups, execution parameters, and outcomes. Various metadata formats and frameworks have been proposed to facilitate the organization, sharing, and automation of experimental results, particularly in computer science and networking research. This document explores the key metadata formats that support experiment reproducibility, focusing on RO-Crate, FAIR principles, and structured experimental workflows.

6.3.1 RO-Crate: A Standardized Metadata Format

One of the leading approaches to structuring experiment metadata is **RO-Crate**, an initiative designed to encapsulate research artifacts and their associated metadata in a machine-readable and human-readable format. RO-Crate follows the principles of the **FAIR data movement**, which emphasizes **Findability, Accessibility, Interoperability, and Reusability**.

RO-Crate provides the following key features:

- **Standardized Packaging**: Research artifacts (such as experiment scripts, raw data, and analysis results) are stored in a structured format.
- **Rich Metadata Annotations**: Each experiment file or dataset is associated with metadata that describes its authorship, affiliations, software dependencies, hardware specifications, and execution environment.
- **Integration with Open Repositories**: RO-Crate packages can be deposited in platforms such as **Zenodo**, ensuring long-term accessibility and referenceability.
- **Machine-Readable Descriptions**: By using JSON-LD (a linked data format), RO-Crate enables automated processing, making it easier to validate, reproduce, and extend previous experiments [38].

RO-Crate has three key use cases that potentially can enhance reusability in research infrastructure experiments. These use cases - Autosubmit, COMPSs, and KEDO Data Lake - demonstrate how RO-Crate facilitates experiment management, workflow execution, and data sharing.

- **Autosubmit:** Autosubmit is a workflow management tool designed to orchestrate and automate complex climate and weather simulations across different computing platforms. By leveraging RO-Crate, Autosubmit ensures that experiment metadata, execution environments, and dependencies are captured in a structured format. This enhances reproducibility and enables researchers to share and reuse experiment configurations effortlessly.
- **COMPSs:** COMPSs (COMP Superscalar) is a programming framework that simplifies the development and execution of parallel applications in distributed computing environments. RO-Crate integration allows COMPSs workflows to be packaged with all necessary metadata, input data, and execution logs. This ensures seamless portability and reuse of computational experiments across diverse RIs.
- **KEDO Data Lake:** The KEDO Data Lake provides a scalable solution for storing and managing large datasets used in scientific research. By adopting RO-Crate, KEDO enables structured metadata

annotation, making datasets more discoverable, interoperable, and reusable. This facilitates seamless data sharing across research communities while maintaining data provenance and integrity.

By adopting RO-Crate in these use cases, RIs can achieve greater experiment reproducibility, streamline workflow management, and foster collaboration through standardized metadata-driven practices.

6.3.2 FAIR Principles and Metadata for Reproducibility

The **FAIR principles** provide a foundational framework for metadata structuring. These principles emphasize:

- **Findability**: Each experiment dataset should be uniquely identified and indexed in searchable repositories.
- **Accessibility**: Data should be retrievable using open protocols, with metadata remaining accessible even if the data itself is restricted.
- **Interoperability**: Metadata should use standard vocabularies to enable integration with other datasets.
- **Reusability**: Experiments should include detailed documentation, licensing information, and methodological descriptions to facilitate replication.

To align with these principles, reproducibility platforms such as **plain orchestrating service (pos)** and **SLICES** have adopted structured experiment workflows that enforce metadata generation at various stages of an experiment [39] [40].

6.3.3 Experiment Workflow and Metadata Structure

A well-defined experimental workflow consists of multiple stages, each requiring specific metadata:

1. **Setup Phase**:
   - Metadata includes hardware configurations, installed software, dependencies, and initial conditions.
   - RO-Crate and FAIR Digital Objects (FDOs) are used to document the environment.
2. **Measurement Phase**:
   - Metadata includes experiment parameters (e.g. network topology, data sampling rates), raw measurements, and logs.
   - Formats such as **Common Workflow Language (CWL)** enable structured execution and reproducibility.
3. **Evaluation Phase**:
   - Metadata includes processing scripts, statistical models, and visualization outputs.
   - Experimental outputs are structured in formats like **JSON or CSV**, with accompanying metadata ensuring clarity.
4. **Publication Phase**:
   - Metadata includes authorship, affiliations, funding sources, and dataset DOIs.
   - Platforms like **Zenodo and GitHub** support direct integration with structured metadata formats.

6.3.4 Metadata Tools and Integration

Several tools facilitate metadata generation and management for experiment reproducibility:

- **Jeppesen**: A tool used for capturing testbed hardware and network topology metadata.

- **RO-Crate-Py**: A Python library that automates the creation of RO-Crate packages for experiments.
- **LLDP (Link Layer Discovery Protocol)**: Used for capturing network configurations dynamically.
- **Ansible, Puppet, and Chef**: Configuration management tools that automate experiment setup and documentation.

Metadata is essential for ensuring reproducible experiments, and frameworks like RO-Crate and FAIR principles provide a structured way to achieve this. By embedding metadata into every stage of an experimental workflow, researchers can improve transparency, facilitate collaboration, and enhance the longevity of scientific findings. Future developments in metadata standards will continue to bridge the gap between experiment execution and data management, ensuring seamless interoperability across research domains.

7 Sustainable Software Design Practices for Scientific Software

7.1 Software Sustainability Recommendations by GSF

The GSF[43] offers several sustainable software design practices that can enhance energy efficiency in scientific software, particularly those focused on data processing, data analytics, and workflow optimization. These practices aim to reduce the carbon footprint of software applications by emphasizing energy efficiency, carbon awareness, and hardware optimization[44].

1. Energy-Efficient Algorithm Design: Optimizing algorithms to perform tasks more efficiently can significantly reduce computational resource usage. For scientific software, this involves selecting or designing algorithms that minimize computational complexity and energy consumption. Efficient algorithms not only speed up data processing and analytics but also lower energy usage, contributing to greener software solutions.

2. Resource Optimization and Virtualization: Efficient management of computational resources is crucial. Implementing virtualization techniques allows multiple applications to share the same hardware resources effectively, leading to better utilization and reduced energy consumption. For scientific workflows, this means consolidating tasks on fewer servers and leveraging cloud computing resources judiciously to optimize energy use.

3. Carbon Awareness in Computing: Designing software that is aware of its carbon emissions enables it to make informed decisions to minimize environmental impact. This includes scheduling intensive data processing tasks during periods when renewable energy sources are more available or when the carbon intensity of the power grid is lower. By aligning computational workloads with greener energy availability, scientific software can reduce its carbon footprint.

4. Efficient Data Management: Data storage and transfer consume significant energy. Implementing efficient data management practices, such as data compression and minimizing unnecessary data movement, can lead to substantial energy savings. For scientific applications dealing with large datasets, optimizing data handling reduces both processing time and energy consumption.

5. Sustainable Hardware Utilization: Selecting energy-efficient hardware and ensuring that software is optimized for the underlying hardware can lead to better performance per watt. This includes using processors and storage devices designed for low power consumption and high efficiency, which is particularly important in data-intensive scientific computations.

By integrating these practices into the development and deployment of scientific software, developers can enhance energy efficiency and contribute to environmental sustainability. The GSF provides

[43] https://greensoftware.foundation/

[44] https://greensoftware.foundation/articles/10-recommendations-for-green-software-development

resources[45] and guidelines[46] to assist in adopting these green software development practices, aiming to reduce the environmental impact of software systems across various domains.

7.2 The SCI Specification by GSF

The **SCI Specification**[47], developed by the GSF, provides a standardized methodology to calculate the carbon emissions associated with software systems. Its primary goal is to assist developers and organizations in making informed decisions to reduce the environmental impact of their software applications.

Overview of the SCI Specification:

The SCI metric quantifies the rate of carbon emissions per functional unit of a software system, expressed as:

$SCI = (E \times I) + M / R$

Where:

- **E**: Energy consumed by the software system.
- **I**: Carbon intensity of the energy source.
- **M**: Embodied emissions from hardware manufacturing.
- **R**: Functional unit representing the software's scale (e.g. per user, per transaction).

This formula encompasses both operational emissions (energy consumption and its carbon intensity) and embodied emissions (hardware production impact), normalized by the software's functional usage.

Applying SCI to Complex Scientific Software in Distributed data centres:

Scientific software often operates across multiple data centres, handling intensive data processing and analytics. Applying the SCI framework to such complex systems involves:

1. **Defining System Boundaries:**
   - **Scope Identification**: Determine which components of the distributed system contribute to carbon emissions, including computing nodes, storage units, and networking equipment.
2. **Measuring Energy Consumption (E):**
   - **Granular Monitoring**: Implement monitoring tools to capture energy usage data for each component across all data centres.
   - **Data Aggregation**: Consolidate energy consumption data to obtain a comprehensive view of the system's operational footprint.
3. **Assessing Carbon Intensity (I):**
   - **Regional Analysis**: Evaluate the carbon intensity of energy sources for each data centre location, considering local energy grids and their reliance on renewable versus non-renewable sources.

[45] GitHub's Green Software Directory [online] https://github.com/github/GreenSoftwareDirectory
[46] https://github.com/Green-Software-Foundation/awesome-green-software
[47] https://sci.greensoftware.foundation/

- **Temporal Factors**: Account for variations in carbon intensity due to time-of-use, as energy sources may shift throughout the day.

4. **Calculating Embodied Emissions (M):**
   - **Hardware Inventory**: Document all hardware components, noting their manufacturing emissions.
   - **Lifecycle Assessment**: Estimate emissions based on hardware production, transportation, and expected lifespan.

5. **Determining the Functional Unit (R):**
   - **Usage Metrics**: Define a functional unit that reflects the software's purpose, such as computations per second, data processed per hour, or user sessions.

6. **Computing the SCI Score:**
   - **Integration**: Apply the SCI formula using the collected data to calculate the carbon intensity per functional unit.
   - **Continuous Monitoring**: Regularly update the SCI score to reflect changes in energy consumption, hardware upgrades, or shifts in workload distribution.

Challenges and Considerations:

- **Data Availability**: Accurate SCI calculation requires detailed energy consumption and carbon intensity data, which may not always be readily accessible, especially in public cloud environments. Collaborating with service providers to obtain this information is crucial.
- **Complexity of Distributed Systems**: The dispersed nature of scientific applications necessitates a coordinated approach to data collection and analysis across all operational regions.
- **Dynamic Workloads**: Fluctuations in computational demand can affect energy usage patterns, requiring adaptive strategies to maintain an accurate SCI assessment.

By systematically applying the SCI Specification, organizations can identify emission hotspots within their scientific software systems and implement targeted optimizations. This proactive approach not only aids in achieving sustainability goals but also promotes operational efficiency and cost savings.

7.3 Sustainable Architecture Design Practices by Cloud Providers AWS and Microsoft Azure

The two leading cloud and Big Data services providers AWS and Microsoft Azure provide advanced resources and services to address the whole spectrum of requirements in designing, deploying and operating cloud based services spanning from front-end user services and large scale cloud based user focused services to advanced services that include IoT, sensor network, edge computing, data collection and management, and computation for data analytics and GenAI services. The technologies used and offered by large cloud providers are advanced and are at the frontline of the technologies developments. Both cloud providers are committed to address the sustainability aspects of their infrastructure and services offered to their customers, providing also necessary design optimisation for customer applications supported by extensive monitoring and fine grained metrics together with services and resources control and management possibilities. Knowing and understanding the technologies, solutions and tools provided by cloud providers is beneficial for effective applications and solutions development for digital RIs, which are generically based on whole spectrum of IoT-edge-cloud continuum of technologies. However, due to Azure and AWS' strong business orientation finding necessary and relevant technical information may be difficult. Presented in this section is an overview of the sustainability design principles and corresponding services formulated by Azure and AWS is intended to provide a guidance for developers of advanced cloud based scientific applications on using State of the Art technologies and architecture design principles, this is supported by references to necessary AWS and Azure resources for further study.

7.3.1 Microsoft Azure Sustainability Framework and Sustainability Design Principles

Microsoft's Azure Sustainability framework is a complex of architecture and design principles proposed by Microsoft Azure's Well-Architected Framework[48] aim to optimize software development, workflow execution, storage efficiency, and cloud resource usage to minimize carbon emissions and energy consumption. It provides a structured approach to reducing the environmental impact of cloud-based applications, focusing on energy-efficient software design, workflow optimization, storage management, and cloud resource utilization. By integrating carbon-aware computing principles, Azure enables developers and businesses to make more sustainable choices in how they design, deploy, and manage digital workloads.

A key principle in sustainable application design is minimizing unnecessary compute operations through optimized software architecture, microservices, and cloud-native execution models. Azure promotes event-driven serverless computing and containerized workloads, allowing applications to scale dynamically while reducing idle resource consumption. Developers are encouraged to use caching, optimize API calls, and structure applications to reduce processing overhead, ultimately lowering energy use and emissions.

Beyond application architecture, workflow optimization plays a critical role in sustainability. Azure enables carbon-aware scheduling, where AI-driven tools shift computational workloads to low-carbon energy periods or regions with cleaner energy grids. This approach is particularly effective for batch processing, ML training, and large-scale analytics, ensuring that intensive computations are performed with minimal environmental impact. Additionally, auto-scaling and adaptive resource allocation ensure that cloud resources are only used when needed, preventing over-provisioning and reducing overall energy waste.

Azure's approach to storage sustainability emphasizes efficient data management, leveraging tiered storage models to classify data based on usage frequency. By compressing files, reducing redundant storage, and implementing intelligent data retention policies, organizations can cut down on unnecessary storage-related energy consumption.

The efficient use of cloud computing resources is another cornerstone of Azure's sustainability strategy. By deploying workloads in regions powered by renewable energy, leveraging spot instances and burstable VMs, and using edge computing to minimize long-distance data transfers, cloud users can significantly reduce their carbon footprint. Serverless platforms and AI-driven workload

[48] https://learn.microsoft.com/en-us/azure/well-architected/

optimizations further enhance resource efficiency, allowing applications to dynamically adjust to demand without consuming unnecessary power.

In general, Azure's sustainability architecture provides a roadmap for building energy-efficient applications and optimizing cloud operations. By integrating carbon-aware computing, adaptive resource management, and sustainable storage strategies, organizations can lower emissions while maintaining high-performance digital services. Through business focused offering like Microsoft Cloud for Sustainability[49], businesses can track their impact in real-time and make data-driven decisions to move toward net-zero cloud operations. This is well presented in the Sustainability Guide[50] that provides also useful information for making decision technology selection.

The following structured text summarises the main sustainability aspects addressed in the Azure Well-Architected Framework with related technologies and services available from Azure (including also reference to related blog articles).

1. Software and Application Design Principles for Sustainability[51]

Microsoft advocates for a green software approach, which aligns with the GSF's principles to ensure applications are developed with energy efficiency, carbon awareness, and hardware efficiency in mind.

- Carbon Efficiency: Applications should minimize resource consumption by using optimized infrastructure and cloud-native services, reducing unnecessary compute and memory overhead.
- Energy Efficiency: Developers should optimize code execution by reducing redundant processing, improving API efficiency, and implementing microservice architectures to allow independent scaling of application components.
- Carbon Awareness: Applications can be programmed to execute computationally intensive workloads during periods of low-carbon energy availability, leveraging Azure's regional carbon intensity tracking to schedule workloads dynamically.
- Hardware Efficiency: Software should be compatible with older devices to extend hardware lifecycles and avoid unnecessary hardware replacements.
- Key Technologies:
- Microservices over Monolithic Design: Enables scalable, event-driven applications that dynamically adjust to demand.
- Server-side Rendering and Caching: Reduces energy-intensive client-side processing by leveraging pre-rendered content and data caching techniques.
- Cloud-native Design Patterns: Implements serverless computing and containerization to efficiently utilize shared cloud resources.

[49] https://www.microsoft.com/en-us/sustainability/cloud

[50] https://www.microsoft.com/en-us/sustainbility/sustainability-guide

[51] Design principles of a sustainable workload - https://learn.microsoft.com/en-us/azure/well-architected/sustainability/sustainability-design-principles

2. Workflow Management for Energy Efficiency and Reduced Environmental Impact[52][53]

Optimizing the workflow execution is critical to reducing unnecessary computation and energy use. Microsoft emphasizes:

- Demand Shifting: Scheduling workloads to execute during periods of renewable energy availability reduces carbon emissions.
- Auto-scaling and Right-sizing: Dynamically adjusting compute resources based on demand prevents over-provisioning.
- Batch Processing During Low-carbon Periods: Computationally intensive tasks such as data analytics and AI model training should be scheduled when energy grids rely more on renewable sources.
- Adaptive Resource Allocation: Workloads should be migrated to Azure data centres with lower carbon intensity, ensuring optimal energy use.
- Deployment and Testing[54]: Run integration, performance, load, or any other intense testing during low-carbon periods, Establish CPU and Memory thresholds in testing

Key Technologies:

- Azure Monitor and AI-driven Auto-scaling: Dynamically adjusts computing power to match workload demand.
- Azure Kubernetes Services (AKS) and Serverless Computing: Supports event-driven execution, reducing idle compute resource waste.

3. Storage Optimization for Sustainable Cloud Computing[55]

Storage solutions play a significant role in cloud sustainability, and Microsoft's Azure Well-Architected Framework provides a structured approach to optimizing data storage and retrieval.

- Storage Tiering: Frequently accessed data should reside in hot storage, while cold or archive storage should be used for infrequent data access, reducing energy-intensive storage operations.
- Compression and Deduplication: Efficient data storage can be achieved by compressing large files and eliminating redundant copies, significantly reducing storage footprint.
- Retention Policies and Smart Backup Strategies: Data backups should be strategically planned to avoid storing unnecessary logs or redundant copies, minimizing storage-related energy consumption.
- Log Optimization: Cloud applications should log only necessary data and avoid excessive data retention that leads to increased storage and processing overhead.

Key Technologies:

- Azure Blob Storage with Hot/Cold/Archive Tiers: Enables energy-efficient storage classification.
- Microsoft Purview for Data Governance: Implements data retention policies to eliminate unnecessary storage use.

[52] Application design of sustainable workloads on Azure - https://learn.microsoft.com/en-us/azure/well-architected/sustainability/sustainability-application-design

[53] Application platform considerations for sustainable workloads on Azure - https://learn.microsoft.com/en-us/azure/well-architected/sustainability/sustainability-application-platform

[54] Testing considerations for sustainable workloads on Azure - https://learn.microsoft.com/en-us/azure/well-architected/sustainability/sustainability-testing

[55] Data and storage design considerations for sustainable workloads on Azure – https://learn.microsoft.com/en-us/azure/well-architected/sustainability/sustainability-storage

- Azure Monitor Logs and Compression Strategies: Reduces redundant log storage and optimizes log retrieval efficiency.

4. Cloud Computing Resource Utilization and Carbon Awareness (refer to previously linked blog articles)

Microsoft emphasizes efficient cloud resource utilization to lower energy consumption. Several strategies are recommended:

- Optimized VMs: Use auto-scaling and spot instances to prevent underutilized compute resources.
- Regional Carbon Intensity Awareness: Applications should be deployed in Azure regions where data centres use renewable energy, reducing the carbon footprint. Connected to Microsoft Emissions Impact Dashboard (EID)[56] to Microsoft Sustainability Manager (MSM)[57] instance
- Serverless and Platform-as-a-Service (PaaS) Workloads: Instead of managing dedicated VMs, serverless computing allows resources to scale on-demand, reducing idle resource consumption.
- Edge Computing and CDNs: Reducing data transfers between cloud regions by caching content closer to users minimizes network-related emissions.

Key Technologies:

- Azure Spot VMs and B-Series Burstable Instances: Reduce carbon impact by using idle compute capacity.
- Azure Carbon-aware Scheduling and Low-carbon Data centres: Enable region-based workload execution for sustainability.
- AKS for Bin Packing: Optimizes VM utilization by grouping workloads efficiently.

Microsoft's Azure Sustainability Architecture provides a robust framework for organizations to reduce the carbon footprint of their cloud operations. By following sustainable application design principles, optimizing workflows, implementing storage best practices, and leveraging carbon-aware cloud computing, organizations can significantly improve their energy efficiency and environmental sustainability. The Microsoft Cloud for Sustainability platform offers real-time tracking and optimization tools that help companies achieve net-zero cloud computing operations, making energy-efficient cloud computing more accessible and scalable.

7.3.2 AWS Sustainability Design Principles of the AWS Well-Architectured Framework

AWS has developed a Sustainability Pillar[58] as part of its Well-Architected Framework[59], providing best practices to reduce the environmental impact of cloud workloads. This framework focuses on software design principles, workflow management, storage optimization, and efficient cloud resource utilization to help organizations build sustainable and energy-efficient cloud architectures. Below we summarise

[56] Measure your cloud carbon footprint with the Emissions Impact Dashboard for Microsoft 365 - https://techcommunity.microsoft.com/blog/microsoft_365blog/measure-your-cloud-carbon-footprint-with-the-emissions-impact-dashboard-for-micr/3144426

[57] Microsoft Sustainability Manager overview - https://learn.microsoft.com/en-us/industry/sustainability/sustainability-manager-overview

[58] Sustainability Pillar - AWS Well-Architected Framework - https://docs.aws.amazon.com/wellarchitected/latest/sustainability-pillar/sustainability-pillar.html

[59] AWS Well-Architected Framework - https://aws.amazon.com/architecture/well-architected/

the AWS architecture design principles for sustainability with reference to AWS technology solutions and cloud services.

Software and Application Design for Sustainability

AWS emphasizes energy-efficient software development by encouraging organizations to optimize resource utilization, minimize redundancy, and use carbon-aware cloud computing. Applications should be designed with dynamic scaling, workload consolidation, and event-driven processing to reduce unnecessary resource consumption.

A key principle is maximizing utilization, ensuring that compute, storage, and network resources are fully utilized before scaling up infrastructure. AWS recommends using managed services, such as AWS Lambda for serverless execution and AWS Fargate for containerized applications, to minimize idle resource usage. Additionally, newer energy-efficient hardware options, such as AWS Graviton processors[60] and AWS Inferentia AI chips[61], enable higher performance per watt, reducing overall energy demand.

AWS offers SageMaker[62] as a powerful cloud based IDE for ML and AI applications development. SageMaker provides functionality for ML models deployment as a part of the MLOps environment and process[63], connecting to multi-model endpoints for ML inference, which allow multiple models to share a single endpoint, reducing unnecessary compute resource duplication.

For ML workloads, AWS suggests adopting pre-trained models and transfer learning instead of training large models from scratch, significantly reducing carbon footprint and training costs. The Amazon SageMaker Training Compiler[64] further optimizes GPU and CPU utilization, cutting down processing time and energy consumption.

Workflow Management for Energy Efficiency

AWS promotes carbon-aware scheduling for workload execution, where tasks are shifted to times and regions with low-carbon energy availability. This ensures optimal workload execution with the lowest environmental impact. Services like AWS IoT Greengrass enable local inference, reducing the need for continuous cloud interactions. For AI/ML workloads, AWS recommends batch processing and edge computing together with using pre-trained models to reduce unnecessary data transfers and computation, especially in the model development stage[65].

[60] https://aws.amazon.com/ec2/graviton/

[61] https://aws.amazon.com/ai/machine-learning/inferentia/

[62] Amazon SageMaker - https://docs.aws.amazon.com/sagemaker/

[63] https://aws.amazon.com/what-is/mlops/

[64] Amazon SageMaker Training Compiler- https://docs.aws.amazon.com/sagemaker/latest/dg/training-compiler.html

[65] https://aws.amazon.com/blogs/architecture/optimize-ai-ml-workloads-for-sustainability-part-2-model-development/

Storage Optimization for Reduced Environmental Impact[66]

AWS advocates tiered storage management, ensuring that data is stored based on access frequency and long-term retention needs.

Best practices include:

- Using Amazon S3 Intelligent-Tiering to automatically move infrequently accessed data to lower-energy storage classes.
- Compressing and deduplicating data to reduce storage footprint and eliminate redundant copies.
- Minimizing log retention and automating data lifecycle management to delete obsolete data.

For database workloads, AWS recommends serverless databases such as Amazon Aurora Serverless, which dynamically scales based on workload demand, reducing idle database instances.

Efficient Use of Cloud Compute Resources[67]

AWS encourages organizations to right-size their compute instances, ensuring that resources are not over-provisioned.

Key strategies include:

- Using energy-efficient AWS Graviton-based instances, which offer better performance per watt than traditional x86 instances.
- Deploying workloads in AWS regions with renewable energy sources, reducing reliance on fossil-fuel-powered data centres.
- Employing AWS Trainium and Inferentia chips for AI/ML inference, which significantly cut down power consumption.
- Reducing data movement across networks by using AWS Direct Connect and edge computing solutions.

In summary, AWS provides a structured approach to sustainable cloud computing, enabling organizations to design, deploy, and manage workloads with minimal environmental impact. Through efficient software design, intelligent workload scheduling, storage optimization, and energy-conscious compute resource allocation, AWS helps businesses achieve their sustainability goals while maintaining high performance.

7.3.3 AWS Sustainability Design Principles and Recommended Architecture Styles

Understanding and applying right architecture style for applications and system design provides multiple benefits with sustainability as one benefits. The topic of the software architecture styles and system architecture styles is sufficiently supported by multiple technical blog articles[68] [69] and one of

[66] https://aws.amazon.com/blogs/architecture/optimizing-your-aws-infrastructure-for-sustainability-part-ii-storage/

[67] https://aws.amazon.com/blogs/architecture/optimizing-your-aws-infrastructure-for-sustainability-part-i-compute/

[68] https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/

[69] https://medium.com/@learnwithwhiteboard_digest/all-major-software-architecture-patterns-explained-8ed6b50a16e3

blog articles at Medium forum provides the most complete reference of software architecture styles[70]. The recent author papers [41] [42] provided summary and suggestions about architecture styles generally suitable for such projects as digital RIs and complex scientific applications.

AWS promotes sustainability through energy-efficient cloud architecture, integrating sustainability principles into software design, workflow execution, storage management, and compute resource allocation. By selecting the right architecture styles, organizations can maximize energy efficiency and reduce their cloud-based environmental impact. The following text provides a summary and suggestions based on the referenced above AWS technical blog posts.

1. Sustainability Design Principles in AWS Well-Architected Framework

AWS outlines key sustainability principles to help organizations reduce the carbon footprint of cloud workloads:

A. Optimize Utilization & Reduce Waste

Design applications to fully utilize allocated resources before scaling up. This reduces idle capacity and energy waste.

Best Architecture Styles: Serverless, Microservices, and Event-Driven Architectures

- Serverless computing (e.g. AWS Lambda) ensures compute resources are only used when needed.
- Microservices architectures improve efficiency by decoupling services, avoiding monolithic inefficiencies.
- Event-driven models (e.g. AWS Step Functions) eliminate always-on infrastructure, improving energy utilization.

B. Use Managed Services to Reduce Operational Overhead

AWS fully-managed services optimize hardware utilization across multiple tenants, reducing energy consumption per user.

Best Architecture Styles: Managed Cloud-Native & SaaS-Based Architectures

- Using AWS Fargate for containers or Aurora Serverless for databases minimizes idle resource waste.
- Cloud-native architectures leverage auto-scaling to efficiently match demand.

C. Optimize Workload Execution for Energy Efficiency

Applications should schedule workloads in energy-optimal locations and times based on grid carbon intensity.

Best Architecture Styles: Edge Computing, Hybrid Cloud, and AI-Driven Workload Scheduling

- Edge computing (AWS IoT Greengrass) moves processing closer to users, reducing cloud data transfers.
- Hybrid cloud architectures allow workloads to run on-premises when energy-efficient and move to AWS when needed.
- AI-based carbon-aware scheduling shifts workloads to AWS regions with renewable energy availability.

D. Optimize Storage and Data Transfer

[70] https://medium.com/bytebytego-system-design-alliance/the-architects-blueprint-understanding-software-styles-and-patterns-with-cheatsheet-5c1f5fd55bbd

Storing and transferring data more efficiently reduces storage costs, energy use, and network traffic. Best Architecture Styles: Data Mesh & Federated Data Processing

- Data mesh architecture reduces data duplication and improves energy efficiency.
- Amazon S3 Intelligent-Tiering moves data to lower-energy storage automatically.
- Federated data processing allows data to be processed at the source, reducing transfer-related energy costs.

2. Architecture Styles and Their Benefits for Energy Efficiency

A. Serverless Architectures (e.g. AWS Lambda, Fargate)

Serverless computing eliminates the need for provisioned infrastructure, using only the required compute capacity. Energy Benefits:

- Reduces idle energy consumption – no constantly running servers.
- Scales automatically, adjusting to demand.
- Lower hardware footprint – shared across multiple workloads.

B. Microservices Architectures (e.g. Amazon ECS, API Gateway)

Applications are broken into smaller, independent services, ensuring more efficient scaling. Energy Benefits:

- Prevents over-provisioning – services scale independently.
- Reduces resource waste by only running the necessary components.
- Compatible with auto-scaling, making it energy-efficient.

C. Event-Driven & Asynchronous Architectures (e.g. AWS Step Functions, SNS/SQS)

Workloads are triggered only when needed, reducing constant polling and CPU waste. Energy Benefits:

- Eliminates unnecessary background processing.
- Better suited for burst workloads, minimizing idle power consumption.
- Efficient use of compute resources, especially for ML and IoT workloads.

D. Edge Computing & Hybrid Cloud (e.g. AWS Wavelength, AWS IoT Greengrass)

Processes data closer to the user, reducing cloud round trips and improving network energy efficiency. Energy Benefits:

- Minimizes cloud data transfer, lowering carbon emissions.
- Uses local resources first, shifting to AWS only when needed.
- Improves application responsiveness while consuming less power.

E. Data Mesh & Federated Learning (e.g. Amazon S3, AWS SageMaker)

Allows distributed data storage and processing, reducing the need for centralized cloud operations. Energy Benefits:

- Lowers energy-intensive data transfers across multiple regions.
- Enables efficient, localized AI training, reducing cloud compute costs.
- Optimizes storage tiers, ensuring high-performance data retrieval with minimal energy impact.

By aligning AWS sustainability design principles with energy-efficient architectures (defined by Architecture Styles), organizations can optimize cloud workloads while minimizing environmental impact.

- Serverless and Microservices reduce idle capacity and scale dynamically, eliminating unnecessary energy use.

- Event-driven and asynchronous workflows ensure computing only happens when needed.
- Edge computing and hybrid cloud models shift workloads closer to the user, improving energy efficiency.
- Data mesh and federated learning optimize data storage and transfer, reducing cloud dependency.

AWS provides a range of sustainable cloud solutions that allow organizations to design energy-efficient architectures without compromising scalability or performance. As cloud adoption grows, selecting the right architecture will be critical to achieving sustainability goals. Learning from AWS can help RI and cloud based services developers to implement effective solutions on their platforms.

7.4 Software Energy Efficiency Tactics and Application to Scientific Software

A potentially useful approach to the general aspects of scientific software design is proposed in the papers by Patricia Lago (Vrije Universiteit Amsterdam) that explore multiple facets of sustainability in cloud computing, software architecture, and ML systems. One of her key studies focuses on optimizing energy efficiency in the public cloud. The papers identify a set of reusable architectural tactics aimed at cloud consumers rather than providers [43]. Through interviews with industry professionals, the authors outline methods such as dynamic scaling, workload migration, and energy-aware scheduling, emphasizing that cloud users must take active measures to optimize software for energy efficiency, despite the lack of transparency in cloud energy consumption reporting.

Another major contribution examines green architectural tactics for ML-enabled systems, addressing the growing energy demands of AI applications [44]. The authors synthesize thirty tactics from an extensive literature review and validate them with industry experts. Their findings emphasize reducing model complexity, optimizing data preprocessing, and leveraging energy-efficient hardware to mitigate the carbon footprint of ML workflows. This study highlights the tension between accuracy-driven AI development and sustainable computing practices, advocating for a balanced approach where performance does not come at an unsustainable energy cost.

In a broader survey of software energy efficiency tactics [45], the authors/researchers collaborate on a systematic review of 163 tactics extracted from over 140 studies. The research shows that interest in energy-efficient software peaked in 2015 but has since declined and that most existing tactics focus on code-level optimisations or dynamic monitoring. However, industry adoption remains low due to a disconnect between academic research and real-world implementation, with many developers unaware of software energy consumption or lacking practical tools to address it. The methodology used in the paper can be used for a new study to understand the software development companies attitude at the present time with the growing importance of sustainability and reduced environmental impact in the whole digital ecosystem.

Table 1 provides a mapping of energy efficiency software design tactics to scientific software design that can be further discussed and applied in project development.

Table 1. Mapping Energy Efficiency design tactics to Scientific Software Design

Category	Proposed Tactics	Applicability to Scientific Software Design
Resource Monitoring	- Automatic monitoring of workload efficiency	Highly relevant: Scientific software often involves large-scale data processing and

	- Experimentation with alternative architectures	benefits from automated performance monitoring to optimize energy consumption.
Resource Allocation	- Dynamic scaling (horizontal & vertical) - Task scheduling for energy efficiency - Workload migration	Highly relevant: Scientific workflows often involve HPC, cloud, and edge computing. Scheduling and workload migration are critical for energy savings.
Resource Adaptation	- Self-adaptive resource management - Reducing computation redundancy - Efficient data caching and storage	Highly relevant: Many scientific workflows involve iterative computations and large data transfers. Adaptation techniques can reduce duplicate computations and data movement overheads.
Green AI/ML Tactics	- Reducing ML model complexity - Efficient data preprocessing - Energy-efficient training (e.g. TPU, GPU)	Partially relevant: Applicable only if scientific software uses ML-based analysis (e.g. bioinformatics, climate modeling).
Cloud Data Storage KPIs	- Storage efficiency monitoring - Minimizing "dark data" - Energy-efficient replication	Highly relevant: Scientific datasets are large, and reducing unnecessary storage and replication is crucial.

The recent paper [46] is focused on the cloud distributed data storage energy efficiency aspects, that the study defines key performance indicators (KPIs) for monitoring the energy efficiency of cloud-based storage systems. The study proposes three primary KPIs—energy consumption, storage utilization, and capacity per watt—offering a structured approach to track and optimize data storage efficiency. The research underscores the critical role of cloud storage in the growing energy footprint of data centres and calls for greater transparency in cloud providers' sustainability metrics.

The summarised papers may provide an approach to bridge the gap between theoretical research and practical implementation, providing actionable frameworks for software architects, cloud users, and data scientists to make more energy-efficient design choices. The surveyed research underscores the urgent need for sustainability in software engineering, advocating for energy-aware architectures, smarter cloud resource management, and transparent energy reporting mechanisms to reduce the environmental impact of digital infrastructures. This is also linked to the research on the integrated/multi-factor energy monitoring in complex digital infrastructures discussed in section 6.1.

8 Data Management

8.1 Sustainability aspects in data management

Sustainability aspects in data management are not well defined and implemented in the research community. Current trends in research data management are primarily focused on ensuring efficient data management, including data, documenting, storing and sharing, what is best defined by FAIR data principles for research data that should ensure that data are FAIR. FAIR compliance provides a basis for the sustainability of data storage, sharing and contributes to the reproducibility of research and interdisciplinary research that require well-documented data and reliable/guaranteed data access with the possibility of their use/integration in future research, with potentially wider scope than initially created data.

However, sustainability aspects in building and operating DMI are primarily discussed and addressed by industry and, in particular, Big Data and cloud service providers. This section provides an overview of existing publications related to both sides/aspects in relation between data management and sustainability.

Sustainability aspects in data management are considered in two main aspects:

1. Data management practices and recommendations to support organisational measures and processes to support energy efficiency and environmental sustainability.
2. Infrastructure and architecture-related approaches to ensure energy efficiency and environmental sustainability (reduce environmental impact or CO_2 footprint) of the DMI and related services to support its corresponding operation. These aspects must be addressed by applying necessary infrastructure and system design principles.

There are known approaches and frameworks being developed and implemented by companies and operators that deal/require extensive/large scale data management related infrastructure services. This relates to necessary DMI for cloud services providers, data analytics applications, and emerging GenAI and Large Language Model (LLM) applications that besides requiring strong computational ability also must supported by distributed and multilayer data storage and access infrastructure (that includes/require access to raw and training dataset, well attributed models, knowledge bases, and semantic caching).

8.2 Data management recommendations and practices to support energy efficiency and environmental sustainability.

This also relates to research infrastructure and services that are focused on monitoring and protecting the environment and ecosystems. Well-established data management must ensure that data are well documented, reliably stored and maintained, and provide sufficient functionality for efficient data sharing. For modern digital RIs and wide research communities, these solutions and practices should comply with the FAIR data principles and be supported with well-defined metadata registries and WebAPI.

Valuable overview and analysis provided by the Dutch Coalition for Sustainable Digitalisation (Nederlands Coalitie Duurzame Digitalisering (NCDD) [47] [48]). Sustainable data management has therefore emerged as a critical strategy for organizations to address the environmental footprint of their data lifecycle. Sustainable data management is not just a technical or operational issue; it is a strategic imperative at the executive level as well as a cultural mind shift. This paper uses the DMBoK[71] phases to address the sustainability considerations in all the phases of your data lifecycle, from planning, creation until the data is destroyed.

NCDD also provides a reference to the position paper published by the group of Prof Patricia Lago (Vrije Universiteit Amsterdam) that has long time experience from years of active research 2010-2022 on the sustainability of software based applications and related infrastructure components. The referred paper of 2021 provided valuable analysis and an overview of disruptive technologies that are transforming industries, this defined as the Lower Energy Acceleration Program (LEAP) [49]. This guide helps stakeholders who want to contribute to building a future-proof energy-efficient digital infrastructure to identify and understand technology trends. Through in-depth analysis and up-to-date information, the LEAP Technology Landscape supports decision-makers and innovators in accelerating innovation.

The current approaches to sustainability data management are analysed in the one of industry leading companies dealing with environmental, social, and governance (ESG) related data management [50]. Current data management practices are largely ad-hoc within the enterprises, alarming for a shift from reactive to proactive methods. In addition, companies need to collaborate to manage their sustainability data efficiently (e.g. having common standards and definitions), since it becomes more difficult to survive as a sole fighter. Therefore, Competence Centre Corporate Data Quality (CC CDQ) launches co-innovation group to elaborate on these issues and to join forces in developing a scalable approach to sustainability data management. In this close collaboration between researchers and practitioners, the group will address various sustainability scenarios (e.g. carbon footprint along the supply chain, supplier qualification, and product labelling), the underlying data requirements, as well as the data management capabilities for sustainability.

Science Europe’s Practical Guide to Sustainable Research Data [51] offers complementary maturity matrices for funders, performers, and data infrastructures to create a common understanding of the approaches needed by the different stakeholders involved. It will support the alignment of policies and requirements for sustainable research data.

Sustainable Research Data Sharing and re-using data to reproduce research and build upon it, are a cornerstone of Open Science. The sustainability of research data refers to their long-term preservation, accessibility, and interoperability. It is something that needs to be addressed by everyone involved. Aligned policies and requirements can limit the collective effort needed and increase the usefulness of research data management.

This Practical Guide provides guidance to ensure the long-term preservation and accessibility of research data. Three complementary maturity matrices provide funders, performers, and data infrastructures with a way to create a common understanding of the approaches needed. These allow them to evaluate the current status of their policies and practices, and to identify next steps towards sustainable data sharing and seeking alignment with other organisations in doing so.

[71] DMBoK is a Data Management Body of Knowledge defined by the Data Management Association International (DAMAI). It is considered as industry standard accepted by majority of companies dealing with the data management in their business and operations.

8.3 Reducing energy consumption and environmental impact of the DMI

Data Dynamics company [52] provides a valuable introduction briefing on their view on environmentally sustainable data management that should balance the growth of data infrastructure with environmental responsibility. This includes the following key points:

1. Environmental Impact of data centres. Data centres contribute 2 % of global carbon emissions, with the US responsible for 0.5 %. The growing demand for digital services and exponential rise in unstructured data strains resources and increases environmental concerns.
2. Challenges with Unstructured Data: Unstructured data accounts for 80 % of stored data, much of which is unused or redundant ("dark data"), consuming energy unnecessarily. Organizations need to shift from indiscriminate data storage to strategic analysis, reducing clutter and environmental impact.
3. Strategies for Sustainable Data Management.

- Hybrid Cloud Adoption, combining on-premises and cloud resources for optimized storage and reduced carbon footprints.
- Data tiering ensures critical data is stored on-premises while less-used data is migrated to energy-efficient cloud environments.

4. Data Minimization: Reduces redundant, obsolete, and trivial data through governance, audits, and lifecycle management. Adopting technologies like data deduplication and compression enhances storage efficiency and lowers energy use.
5. Unified Data Management (UDM) supported by necessary tools for actionable data insight and use, employs AI, ML, and automation to streamline data operations securely and efficiently.

Dataversity company that is primarily focused on professional training on enterprise data management [53], highlights the growing environmental impact of digital data and the necessity of sustainable data management to mitigate it. The following aspects need to be addressed:

Environmental Impact of Data and IT Infrastructure

- Data growth: The exponential increase in digital data requires vast energy resources for storage, processing, and transmission, contributing to carbon emissions and resource depletion.
- E-Waste: Manufacturing and disposal of electronic devices create hazardous waste, harming ecosystems and human health.

Strategies for Sustainable Data Management

- Energy-Efficient data centres: Incorporate advanced cooling technologies, renewable energy sources (solar, wind, hydroelectric), and energy-efficient hardware to minimize energy consumption and carbon emissions.
- Cloud Computing: transitioning to sustainable cloud platforms leverages energy-efficient, large-scale data centres and reduces organizational carbon footprints.
- Data Optimization: Implement data compression, deduplication, and lifecycle management to reduce redundant and obsolete data.
- Waste Management: Embrace a circular economy through refurbishing, recycling, and proper disposal of outdated equipment.
- Governance and Compliance Data Governance: Establish policies ensuring efficient, secure, and transparent data handling while meeting regulatory and ethical standards.

Certifications:

- Achieving certifications like Leadership in Energy and Environmental Design (LEED), Energy Star, or EPAT demonstrates commitment to sustainable practices.

Technological and Operational Advancements

- Green Computing: environmentally friendly technologies, including energy-efficient hardware, virtualization, and renewable energy integration.
- UDM: tools like UDM enhance data quality, integration, and visibility, enabling better decision-making and sustainability.
- Metrics and Future Outlook Real-time monitoring of energy consumption, emissions, and waste ensures compliance and enables targeted sustainability measures. The importance of sustainable data management is expected to grow in 2024, driven by data value, regulatory changes, and environmental concerns.

The principles of sustainable data management align with reducing environmental impact while supporting business growth, paving the way for a greener digital future.

9 Ongoing Research on Sustainability in RIs

Sustainability in RIs revolves around their ability to provide cutting-edge services while adhering to principles of environmental, economic, and social responsibility. According to Demchenko et al., sustainable RIs require well-defined architectural principles that ensure long-term evolution and integration of modern technologies, addressing the lifecycle of resources from development to decommissioning [54]. Another paper emphasizes that adopting green infrastructure practices, such as renewable energy and resource-efficient technologies, can significantly reduce the ecological footprint of RIs while fostering socio-economic benefits [55].

Case studies like SLICES-RI highlight the importance of integrating digital and data-driven solutions to enhance operational efficiency while maintaining sustainability [54] [55]. A comprehensive framework for sustainable RIs includes policies promoting energy-efficient infrastructure, federated data management, and lifecycle-oriented design principles [55]. However, challenges persist in balancing high performance with sustainability, particularly as advanced computational needs increase energy consumption and resource dependency. Effective strategies, such as leveraging multi-functional green infrastructure and integrating AI-driven resource optimization, are crucial to maintaining this equilibrium while achieving broader sustainability goals.

9.1 Evaluation and Comparative Analysis

The evaluation of sustainability in RIs necessitates a comprehensive analysis of their environmental impact, especially regarding energy consumption, resource utilization, and waste production. Current practices, while effective in operational terms, often contribute significantly to greenhouse gas emissions and resource inefficiency. The adoption of green infrastructure, as highlighted in , represents a pivotal strategy for sustainable development, leveraging natural systems to reduce ecological footprints while providing economic and social benefits.

Technology Readiness Levels (TRLs) for sustainable solutions vary widely, with innovative approaches like renewable energy-powered data centres and advanced cooling systems reaching higher TRLs (7-9) in some applications, while others remain in early development stages (TRL 4-6). A comparative analysis of technologies reveals key trade-offs: for instance, while green roofs and permeable pavements significantly mitigate urban heat islands and water runoff, their implementation costs can be a barrier. Similarly, AI-driven resource optimization tools improve energy efficiency but demand substantial adaptation of existing infrastructures. As emphasized in [55], effective planning and investment in green technologies are crucial to balance development and ecosystem preservation. This underscores the need for a systematic framework that integrates advanced technologies with sustainable practices, ensuring long-term viability without compromising RI performance.

9.2 Recommendations for RIs

To ensure the sustainability of RIs, immediate actions should focus on implementing energy-efficient technologies, such as renewable energy sources and advanced cooling systems, and adopting green infrastructure practices to reduce environmental impact, as highlighted by Kumar et al. Existing RIs must also enhance data management frameworks, integrating federated solutions to optimize resource use and reduce redundancies. For long-term strategic directions, the paper [41] proposes adopting sustainable architectural design principles, which include lifecycle-oriented planning, modularity, and scalability to support evolving technological needs while minimizing waste.

Policies should prioritize funding mechanisms that incentivize green innovations, such as nature-based solutions and low-impact development strategies, alongside fostering international collaboration to share expertise and resources. Regulatory frameworks must emphasize compliance with sustainability standards and promote training programs for personnel to address skill gaps in managing complex, sustainable RI systems. These steps are essential to position RIs as leaders in sustainable development and as contributors to global ecological resilience.

9.3 GenAI impact on energy efficiency

GenAI has emerged as a transformative technology with significant implications for energy efficiency in data centres and broader energy systems [56]. The immense computational demands of GenAI models, such as training and inference for LLMs, require energy-intensive infrastructure. For instance, the energy consumption of a single training session for a large model can exceed the annual electricity use of multiple households. To address this challenge, we need to investigate systematically energy impact of GenAI models on infrastructures [57]. Companies like Hitachi are integrating renewable energy sources, energy storage systems, and smart grid technologies to power data centres sustainably, ensuring resilience and efficiency in energy consumption.

Moreover, GenAI itself offers tools for optimizing energy systems. As highlighted by recent research, GenAI can enhance energy harvesting technologies in wireless networks, improving resource allocation, network deployment, and real-time energy management [56]. Applications such as UAV-assisted energy transfer and renewable energy forecasting demonstrate the role of GenAI in creating dynamic, adaptive systems capable of balancing energy generation and consumption. These capabilities enable the reduction of carbon footprints and operational costs while maintaining high performance.

To fully realize sustainable GenAI applications, it is essential to align the development of AI models with green computing principles, leveraging energy-efficient hardware and algorithms. Additionally, policies must support investments in renewable energy and the integration of AI-driven solutions in energy management to minimize the environmental impact of this transformative technology. This multifaceted approach ensures that GenAI contributes not only to technological advancement but also to global sustainability goals.

10 Conclusion and Future Work

The GreenDIGIT project has provided a comprehensive SotA assessment of sustainability in RIs and digital infrastructure. This report has explored various aspects of energy-efficient computing, carbon-aware scheduling, green software design, and sustainable data management, highlighting the need for integrated solutions to reduce the environmental footprint of digital infrastructures.

The findings emphasize that sustainability in RIs is no longer optional - it is a necessary shift toward more energy-conscious digital operations. RIs must adopt a multi-layered approach that combines energy-efficient hardware, optimized workload scheduling, sustainable software practices, and responsible data management. The implementation of carbon-aware computing, as demonstrated by leading industry and research initiatives, showcases the potential of intelligent resource allocation to minimize carbon emissions while maintaining performance.

Despite the progress made in energy-efficient computing solutions, several challenges remain. The adoption of sustainability metrics and standardization across federated computing infrastructures requires further refinement. Additionally, the integration of ML and AI-driven optimizations for energy-aware scheduling is still an emerging field, with potential trade-offs between computational efficiency and sustainability that need to be further analysed.

The outcome of the presented State of the Art will provide the background for future research within the GreenDIGIT project that will focus on developing and validating prototype solutions that incorporate real-time energy monitoring, automated sustainability reporting, and federated workload scheduling strategies. The project will also investigate data lifecycle management techniques to reduce redundant data transfers and improve energy efficiency and sustainability of the DMI and distributed storage systems.

To support the practical implementation of these findings, GreenDIGIT will collaborate with research institutions, universities, industry stakeholders, cloud providers to ensure that sustainable practices become an integral part of digital infrastructure design. Additionally, efforts will be made to develop training programs for research communities, raising awareness about the environmental impact of computing and equipping researchers with tools for greener digital practices.

Ultimately, the goal is to position European RIs at the forefront of sustainable computing. By leveraging carbon-aware computing, energy-efficient software development, and optimized data workflows, GreenDIGIT aims to create a scalable framework for the next generation of environmentally responsible RIs.

11 References

[1] Z. Zhang, S. Liang, F. Yao and X. and Gao, “Red Alert for Power Leakage: Exploiting Intel RAPL-Induced Side Channels,” in *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, 2021.

[2] L. Moritz, A. Kogler, D. Oswald, M. Schwarz, C. Easdon, C. Canella and a. D. Gruss, “PLATYPUS: Software-based power side-channel attacks on x86,” in *Proceedings - IEEE Symposium on Security and Privacy*, 2021.

[3] P. M. K., S. W. Poole, C. H. Hsu, D. Maxwell, W. Tschudi, H. Coles, D. J. Martinez and a. N. Bates, “TUE, a new energy-efficiency metric applied at ORNL's Jaguar,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013.

[4] M. Ferroni, J. A. Colmenares, S. Hofmeyr, J. D. Kubiatowicz and a. M. D. Santambrogio, “Enabling power-awareness for the Xen hypervisor,” *ACM SIGBED Review,* vol. 15, no. 1, pp. 36-42, 2018.

[5] G. Achim, R. Bender, C. Calero, G. S. Fernando, M. Funke, J. Gröger, L. M. Hilty and e. al, “Development and evaluation of a reference measurement model for assessing the resource and energy efficiency of software products and components—Green Software Measurement Model (GSMM),” *Future Generation Computer Systems,* vol. 155, pp. 402-418, 2024.

[6] K. Bartłomiej, P. Czarnul and a. J. Proficz, “Energy-Aware Scheduling for High-Performance Computing Systems: A Survey,” *Energies,* vol. 16, no. 2, p. 890, 2023.

[7] I. Oleinikova, “How Flexibility Can Support Power Grid Resilience,” IEEE Smart Grid, 2022. [Online]. Available: https://smartgrid.ieee.org/bulletins/february-2022/how-flexibility-can-support-power-grid-resilience.

[8] W. Buchanan, J. Foxon, D. Cooke, S. Iyer, E. Graham, B. DeRusha, C. Binder, K. Chiu and e. al, “Carbon-aware computing: Measuring and reducing the carbon footprint associated with software in execution,” *Microsoft White Paper,* p. 15, 2023.

[9] R. Ana, R. Koningstein, I. Schneider, B. Chen, A. Duarte, B. Roy, D. Xiao and e. al., “Carbon-Aware Computing for Datacenters,” *IEEE Transactions on Power Systems,* vol. 38, no. 2, pp. 1270-1280, 2023.

[10] S. Ilias, K. Dimitris, M. Nikos and K. Thanasis, “EELAS: Energy Efficient and Latency Aware Scheduling of Cloud-Native ML Workloads,” in *15th International Conference on COMmunication Systems and NETworkS, COMSNETS*, 2023.

[11] A. Kamatar, M. Gonthier, V. Hayot-Sasson, A. Bauer, M. Copik, T. Hoefler, R. C. Fernandez, K. Chard and a. I. Foster, *Core Hours and Carbon Credits: Incentivizing Sustainability in HPC,* Arxiv, Cornel University, 2025.

[12] A. Tsaregorodtsev and a. t. D. Project, “DIRAC Distributed Computing Services,” *Journal of Physics: Conference Series,* vol. 513, no. 3, 2014.

[13] A. Casajus, R. Graciani, S. Paterson, A. Tsaregorodtsev and a. L. D. TEam, “DIRAC pilot framework and the DIRAC Workload Management System,” *Journal of Physics: Conference Series,* vol. 219, 2010.

[14] M. S. Danladi and a. M. Baykara, “Low Power Wide Area Network Technologies: Open Problems, Challenges, and Potential Applications,” *Review of Computer Engineering Studies,* vol. 9, no. 2, pp. 71-78, 2022.

[15] I. Syrigos, D. Kefalas, N. Makris and a. T. Korakis, “EELAS: Energy Efficient and Latency Aware Scheduling of Cloud-Native ML Workloads,” in *15th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, 2023.

[16] X. Ding and a. J. Wu, “Study on Energy Consumption Optimization Scheduling for Internet of Things,” *IEEE Access,* p. 99, 2019.

[17] B. K. Jha, A. Tiwari, R. B. Kuhada and a. N. M. Pindoriya, “IoT-enabled Smart Energy Management Device for Optimal Scheduling of Distributed Energy Resources,” *Electric Power Systems Research,* vol. 229, p. 110121, 2024.

[18] G. Edouard, “Assessing the environmental impact of mobile applications: a measure framework toward DevGreenOps,” in *Proceedings - IEEE/ACM 11th International Conference on Mobile Software Engineering and Systems, MOBILESoft 2024*, 2024.

[19] R. Horn, A. Lahnaoui, E. Reinoso, S. Peng, V. Isakov, T. Islam and I. and Malavolta, “Native vs Web Apps: Comparing the Energy Consumption and Performance of Android Apps and their Web Counterparts,” in *Proceedings - IEEE/ACM 10th International Conference on Mobile Software Engineering and Systems, MOBILESoft 2023*, 2023.

[20] V. Frattaroli, O. Le Goaer and O. and Philippot, “Ecological Impact of Native Versus Cross-Platform Mobile Apps: A Preliminary Study,” in *Proceedings - 38th IEEE/ACM International Conference on Automated Software Engineering Workshops, ASEW 2023*, 2023.

[21] X. Zhan, T. Liu, L. Fan, L. Li, S. Chen, X. Luo and Y. and Liu, “Research on Third-Party Libraries in Android Apps: A Taxonomy and Systematic Literature Review,” *IEEE Transactions on Software Engineering,* vol. 48, no. 10, pp. 4181-4213, 2022.

[22] L. Mosesso, N. Maudet, E. Nano, T. Thibault and A. and Tabard, “Obsolescence Paths: Living with Aging Devices,” in *Proceedings - International Conference on ICT for Sustainability, ICT4S 2023*, 2023.

[23] A. Bogdan and I. and Malavolta, “An Empirical Study on the Impact of CSS Prefixes on the Energy Consumption and Performance of Mobile Web Apps,” in *Proceedings - IEEE/ACM 11th International Conference on Mobile Software Engineering and Systems, MOBILESoft 2024*, 2024.

[24] W. Oliveira, R. Oliveira and F. and Castor, “A study on the energy consumption of android app development approaches,” in *IEEE International Working Conference on Mining Software Repositories*, 2017.

[25] J. Ferreira, B. Santos, W. Oliveira, N. Antunes, B. Cabral and J. P. and Fernandes, “On Security and Energy Efficiency in Android Smartphones,” in *Proceedings - IEEE/ACM 10th International Conference on Mobile Software Engineering and Systems, MOBILESoft 2023*, 2023.

[26] H. David, E. Gorbatov, U. R. Hanebutte, R. Khanna and C. and Le, “RAPL: Memory power estimation and capping,” in *Proceedings of the International Symposium on Low Power Electronics and Design*, 2010.

[27] B. Dornauer and M. and Felderer, “Energy-Saving Strategies for Mobile Web Apps and their Measurement: Results from a Decade of Research,” in *Proceedings - IEEE/ACM 10th International Conference on Mobile Software Engineering and Systems, MOBILESoft 2023*, 2023.

[28] E. Guegain, T. Simon, A. Rahier and R. and Rouvoy, “Managing Uncertainties in ICT Services Life Cycle Assessment Using Fuzzy Logic,” in *Proceedings - 10th International Conference on ICT for Sustainability, ICT4S 2024*, 2024.

[29] O. Le Goaër and J. and Hertout, “EcoCode: A SonarQube Plugin to Remove Energy Smells from Android Projects,” in *ACM International Conference Proceeding Series*, 2022.

[30] H. Anwar, D. Pfahl and S. N. and Srirama, “Evaluating the Impact of Code Smell Refactoring on the Energy Consumption of Android Applications,” in *Proceedings - 45th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2019*, 2019.

[31] É. Guégain, R. Raes, N. Chachignot, C. Quinton and R. and Rouvoy, “AndroWatts: Unpacking the Power Consumption of Mobile Device's Components,” in *MOBILESoft'25 - 12th International Conference on Mobile Software Engineering and Systems*, 2025.

[32] R. Buyya, S. Ilager and P. Arroba, “Energy-efficiency and sustainability in new generation cloud computing: A vision and directions for integrated management of data centre resources and workloads,” *SPE,* 2023.

[33] S. Ilager, R. Ramamohanar and R. Buyya, “Thermal prediction for efficient energy management of clouds using machine learning,” *TPDS,* 2021.

[34] S. Ilager, R. Ramamohanar and R. Buyya, “ETAS: Energy and thermal-aware dynamic virtual machine consolidation in cloud data center with proactive hotspot mitigation,” *CCPE,* 2019.

[35] K. Jeffery, L. Candela and &. H. Glaves, “Virtual Research Environments for Environmental and Earth Sciences: Approaches and Experiences,” in *Towards Interoperable Research Infrastructures for Environmental and Earth Sciences* , Springer Nature Lecture Notes in Computer Science, 2020, pp. 272-289.

[36] G. Sipos, G. L. Rocca, D. Scardaci and a. P. Solagna, “The EGI applications on Demand service,” *Future Generation Computer Systems,* vol. 98, pp. 171-179, 2019.

[37] A. Souza, N. Bashir, J. Murillo, W. Hanafy, Q. Liang, D. Irwin and a. P. Shenoy, “Ecovisor: A Virtual Energy System for Carbon-Efficient Applications,” in *ASPLOS 2023: Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, 2023.

[38] E. Hauser, S. Gallenmuller and G. and Carle, “RO-Crate for Testbeds: Automated Packaging of Experimental Results,” in *IFIP Networking Conference, IFIP Networking*, 2024.

[39] Y. Demchenko, S. Gallenmuller, S. Fdida, P. Andreou, C. Crettaz and M. and Kirkeng, “Experimental Research Reproducibility and Experiment Workflow Management,” in *15th International Conference on COMmunication Systems and NETworkS, COMSNETS*, 2023.

[40] S. Gallenmüller, D. Scholz, H. Stubbe, E. Hauser and G. and Carle, “Reproducible by Design: Network Experiments with pos,” in *Würzburg Workshop on Next-Generation Communication Networks (WueWoWas'22)*, 2022.

[41] Y. Demchenko, “Sustainable Architecture Design Principles for Large Scale Research Infrastructure Projects,” in *Proceedings the 25th International Conferences on High Performance Computing and Communications (HPCC2023), 13-15 December 2023, Melbourne, Australia*.

[42] Y. Demchenko, “The Importance of System Engineering Competences and Knowledge in Large Scale Digital Research Infrastructure Projects,” in *Proceedings EDUCON2024 Conference, 8-11 May 2024, Kos, Greece*.

[43] S. Vos, P. Lago, R. Verdecchia and I. Heitlager, “Architectural Tactics to Optimize Software for Energy Efficiency in the Public Cloud,” in *Proc. 2024 IEEE/ACM 46th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS), 14-20 April 2024*.

[44] H. Järvenpää, P. Lago, J. Bogner, G. Lewis, H. Muccini and I. Ozkaya, “A Synthesis of Green Architectural Tactics for ML-Enabled Systems,” in *Heli Järvenpää; Patricia Lago; Justus Bogner; Grace Lewis; Henry Muccini; Ipek Ozkaya*.

[45] J. Balanza-Martinez, P. Lago and R. Verdecchia, “Tactics for Software Energy Efficiency: A Review.,” *Environmental Informatics 2023. ENVIROINFO 2023. Progress in IS. Springer, Cham. https://doi.org/10.1007/978-3-031-46902-2_7.*

[46] P. Banijamali, I. Fatima, P. Lago and I. Heitlager, “Key Performance Indicators for the Energy Efficiency of Cloud Data Storage,” *IEEE, https://research.vu.nl/ws/portalfiles/portal/311058933/GREENS-8-CameraReady.pdf.*

[47] “Nederlands Coalitie Duurzame Digitalisering (NCDD), Publications,” [Online]. Available: https://coalitieduurzamedigitalisering.nl/publicaties/. [Accessed 2025].

[48] “Sustainable data management: Implementing a Sustainable Data Strategy Across the Data Lifecycle, NCCD White Paper, 29 Sept 2024,” [Online]. Available: https://coalitieduurzamedigitalisering.nl/wp-content/uploads/NCDD-DLCM-v1.01.pdf.

[49] R. Verdecchia, Lago, P. and de Vries, C., “The LEAP Technology Landscape: Lower Energy Acceleration Program (LEAP) Solutions, Adoption Factors, Impediments, Open Problems, and

Scenarios, VU Research Portal, 2021," [Online]. Available: https://amsterdameconomicboard.com/app/uploads/2021/06/LEAP-Technology-Landscape-Trends-Scenarios-longread-1.pdf.

[50] CC CDQ , "Data Management Sustainability for consistent ESG compliance," [Online]. Available: https://www.cdq.com/events-insights/research/data-management-sustainability.

[51] "Practical Guide to Sustainable Research Data, Science Europe, 2021," [Online]. Available: https://scienceeurope.org/our-resources/practical-guide-to-sustainable-research-data/.

[52] "Transitioning from Dark Data to Green Growth: Two Essential Pillars for Data Sustainability in the Digital Age, Data Dynamics Inc., 2024," [Online]. Available: https://www.datadynamicsinc.com/blog-transitioning-from-dark-data-to-green-growth-two-essential-pillars-for-data-sustainability-in-the-digital-age/.

[53] P. Ghosh, "Sustainable Data Management: Trends and Benefits, Dataversity, March 21, 2024," [Online]. Available: https://www.dataversity.net/sustainable-data-management-an-overview.

[54] Y. Demchenko, d. L. Cees and W. Los, "Sustainable Architecture Design Principles for Large Scale Research Infrastructure Projects," in *Proc. The International Conference on High Performance Computing and Simulation (HPCS 2020), 10-14 Dec 2020*.

[55] K. Pawan, Mukul, K. Dilpreet and K. Amrit, "Green Infrastructure- A Roadmap Towards Sustainable Development," in *2023 IOP Conf. Ser.: Earth Environ. Sci.*.

[56] Y. Bo, "Taking on Generative AI's Green Energy Dilemma," [Online]. Available: https://social-innovation.hitachi/en-us/think-ahead/digital/green-energy-for-generative-ai/.

[57] P. Maliakel, S. Ilager and a. I. Brandic, "Investigating Energy Efficiency and Performance Trade-offs in LLM Inference Across Tasks and DVFS Settings," *https://arxiv.org/abs/2501.08219.*