TEADAL

D3.2 Reducing energy footprint in federated stretched data lakes

Revision: v.1.0

Work package	WP3
Task	Task 3.3
Due date	31/08/2024
Submission date	31/08/2024
Deliverable lead	Ubiwhere
Version	1.0
Authors	Bruno Feitais *(UBIWHERE),* Hélio Simeão *(UBIWHERE),* Eduardo Brito *(CYB)*, Mattia Salnitri *(POLIMI)*, Monica Vitali *(POLIMI),* Josep Escrig *(I2CAT)*
Reviewers	Josep Escrig *(I2CAT),* Sebastian Werner *(TUB),* Ofer Biran *(IBM),* Kathrine Barabash *(IBM)*
Abstract	This deliverable focuses on addressing the energy efficient challenge by developing strategies to better understand the energy footprint of federated stretched data lakes.
Keywords	Efficient, Energy, Cluster, Deployment, Modelling, Node, Pipeline, TEADAL, TEE

Grant Agreement No.: 101070186
Call: HORIZON-CL4-2021-DATA-01
Topic: HORIZON-CL4-2021-DATA-01-01
Type of action: HORIZON-RIA

Document Revision History

Version	Date	Description of change	List of contributor(s)
V0.1	20/05/2024	ToC	Hélio Simeão *(UW)*, Bruno Feitais *(UW)*
V0.2	28/06/2024	First draft	Bruno Feitais *(UW)*, Hélio Simeão *(UW)*, Eduardo Brito *(CYB)*, Mattia Salnitri *(POLIMI)*, Monica Vitali *(POLIMI)*, Josep Escrig *(I2CAT)*
V0.3	19/07/2024	Complete contents, ready for review	Bruno Feitais *(UW)*, Hélio Simeão *(UW)*, Eduardo Brito *(CYB)*, Mattia Salnitri *(POLIMI)*, Monica Vitali *(POLIMI)*, Josep Escrig *(I2CAT)*, Josep Escrig *(I2CAT)*, Sebastian Werner *(TUB)*, Ofer Biran *(IBM)*
V1.0	31/08/2024	Final version	Bruno Feitais *(UW)*, Hélio Simeão *(UW)*, Eduardo Brito *(CYB)*, Mattia Salnitri *(POLIMI)*, Monica Vitali *(POLIMI)*, Josep Escrig *(I2CAT)*, Josep Escrig *(I2CAT)*, Sebastian Werner *(TUB)*, Ofer Biran *(IBM)*, Kathrine Barabash *(IBM)*

Disclaimer

Funded by the European Union

Funded by the European Union (TEADAL, 101070186). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Copyright Notice

Project funded by the European Commission in the Horizon Europe Programme		
Nature of the deliverable:	**R**	
Dissemination Level		
PU	*Public, fully open, e.g. web (Deliverables flagged as public will be automatically published in CORDIS project's page)*	✔
SEN	*Sensitive, limited under the conditions of the Grant Agreement*	
Classified R-UE/ EU-R	*EU RESTRICTED under the Commission Decision No2015/ 444*	
Classified C-UE/ EU-C	*EU CONFIDENTIAL under the Commission Decision No2015/ 444*	
Classified S-UE/ EU-S	*EU SECRET under the Commission Decision No2015/ 444*	

EXECUTIVE SUMMARY

This deliverable is focused on understanding the energy footprint of federated stretched data lakes, a growing concern as data storage and processing demands increase. Deliverable D3.2 outlines the project's strategy to enhance energy efficiency through the development of innovative tools, policies, and governance frameworks.

Key to this strategy are tools like Kepler and Scaphandre, which monitor energy consumption in real-time. These tools provide critical insights that inform more precise strategies for reducing energy usage across data lake operations. The project has also designed energy-aware policies that carefully balance energy efficiency with robust data security, especially within Trusted Execution Environments (TEEs).

In addition to these policies, this deliverable has established governance strategies that focus on reducing energy consumption while managing data effectively. A decision-making system has been introduced to guide the energy-efficient execution of data pipelines. These efforts ensure that energy considerations are integrated into every aspect of data management.

The findings of Deliverable D3.2 demonstrate that significant energy savings can be achieved without sacrificing security or efficiency. By embedding energy efficiency into data governance, the TEADAL node not only addresses immediate technical challenges but also contributes to broader environmental sustainability goals. The project's outcomes provide a solid foundation for future innovations in sustainable data management.

Funded by
the European Union

TABLE OF CONTENTS

Funded by
the European Union

LIST OF FIGURES

Funded by the European Union

LIST OF TABLES

Funded by
the European Union

ABBREVIATIONS

ACPI	Advanced Configuration and Power Interface
AES	Advanced Encryption Standard
BPMN	Business Process Model and Notation
CPU	Central Processing Unit
CSV	Comma-Separated Values
DRAM	Dynamic Random Access Memory
eBPF	extended Berkeley Packet Filter
FDP	Federated Data Product
GDPR	General Data Protection Regulation
JSON	JavaScript Object Notation
KPI	Key Performance Indicators
RAM	Random Access Memory
RAPL	Running Average Power Limit
SLA	Service Level Agreements
sFDP	shared Federated Data Product
TEE	Trusted Execution Environments
VM	Virtual Machine
ZKP	Zero-Knowledge Proofs

Funded by
the European Union

INTRODUCTION

The increasing energy consumption of computing systems has become a significant concern in today's technology-driven world. As the demand for data processing and storage grows, so does the energy footprint of data lakes, which serve as repositories for vast amounts of data.

The TEADAL ("Trustworthy, Energy-Aware federated Data Lakes along the computing continuum") project aims to address this challenge by developing strategies to better understand the energy footprint of federated stretched data lakes.

This deliverable presents an overview of the exploitation plans of the TEADAL project, focusing on key exploitable results and pilot perspectives. It explores various tools and methodologies for measuring and estimating energy consumption in data lakes and proposes models for energy-aware data storage, processing, and transmission.

By leveraging technologies such as Kubernetes, Kepler, and Scaphandre, TEADAL seeks to create an energy-efficient and environmentally sustainable data management framework.

This deliverable is structured into several key sections, each addressing a specific aspect of the project's objectives:

1. **Energy Consumption in TEADAL**
   This section explores the methodologies and tools used to measure and monitor energy consumption within the TEADAL node. It introduces tools like Kepler and Scaphandre, which are integral to capturing accurate energy metrics and understanding the relationship between data operations and energy use.
2. **Energy Consumption Policies**
   The focus here is on the development of policies that balance the need for energy efficiency with the requirements of data security. This section discusses strategies for optimising the placement and scheduling of tasks within Trusted Execution Environments (TEEs) to ensure that security considerations do not hinder energy efficiency.
3. **Energy-Aware Data Governance**
   Effective data governance is crucial for managing data in a way that minimises energy consumption. This section reviews strategies for managing data friction and gravity while maintaining energy efficiency. It also presents a decision-making system designed to guide the execution of data pipelines with energy considerations at the forefront.
4. **Employing ZK SLA Monitoring for Environmental Agreements**
   This section introduces an approach for monitoring environmental agreements within TEADAL using Zero-Knowledge Service Level Agreement (ZK SLA) monitoring. It outlines how this method enhances compliance, trust, and resource allocation while adhering to environmental standards.

Funded by the European Union

1. Energy Consumption in TEADAL

Energy consumption of computing systems has become a major concern. Constrained by cost, environmental concerns, and policy, minimising the energy footprint of computing systems is one of the primary goals of TEADAL. It's important to take the first step in this matter, and for that, it's important to start being aware of what consumes energy inside the TEADAL node.
TEADAL uses many different tools/technologies and processes that consume energy. To be mindful of the TEADAL node's consumption, it is necessary to find the right tools and metrics to measure the energy consumption. Some of those metrics can be Watt used by each pod, service or process, the CPU utilisation, or the memory utilisation.
Some tools are available as open-source that can be deployed on the TEADAL node.
Those tools will help every TEADAL user understand and study the energy consumption of the TEADAL node, and subsequently, their impact on the environment.

1.1 Measuring Energy Consumption of Data Lakes

Data Lakes have a lot of different processes that consume energy on a daily basis, some of them are data ingestion, data processing, data transmission, data security, data storage, etc, as seen in Figure 1.

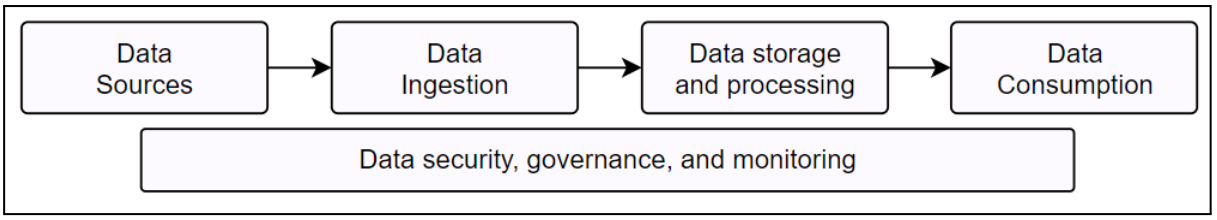

Figure 1: Data Lake Architecture

Assessing a Data Lake's energy consumption requires identifying and calculating the energy used by these processes. Accurately measuring a Data Lake's energy consumption requires analysing the energy usage of the specified hardware components across all data lake activities. Currently, there is no solution for comprehensively measuring the energy consumption of Data Lakes.
To be mindful of TEADAL's energy usage, it will be necessary to implement a tool that analyses each process and hardware component, and develop models to estimate the environmental impact of processes that can't be directly measured, like storage, computation, and network (data transmission).
The tools considered for this effect are Grafana, Prometheus, Kepler, and Scaphandre. All of these tools were developed with the intention of making a monitoring system. Kepler and Scaphandre work with Prometheus to probe energy-related system stats on a dashboard available on Grafana.
For example, for CPU and memory utilisation Kubernetes Metrics Server, Prometheus and Grafana are ideal, they give the user free access to a customisable dashboard with the chosen parameters by the user. However, to read power consumption values this is not enough, it's necessary to deploy Grafana, Prometheus, and Kepler or Scaphandre.

Funded by the European Union

1.1.1 Kepler

Kepler (Kubernetes Efficient Power Level Exporter) uses eBPF (extended Berkeley Packet Filter) to probe energy-related system stats and exports them as Prometheus metrics [1]. Kepler Exporter exposes various metrics about the energy consumption of Kubernetes components such as Pods and Nodes.
To visualise the power consumption metrics made available by the Kepler Exporter, it is possible to use the Kepler Dashboard on Grafana, Figure 2.

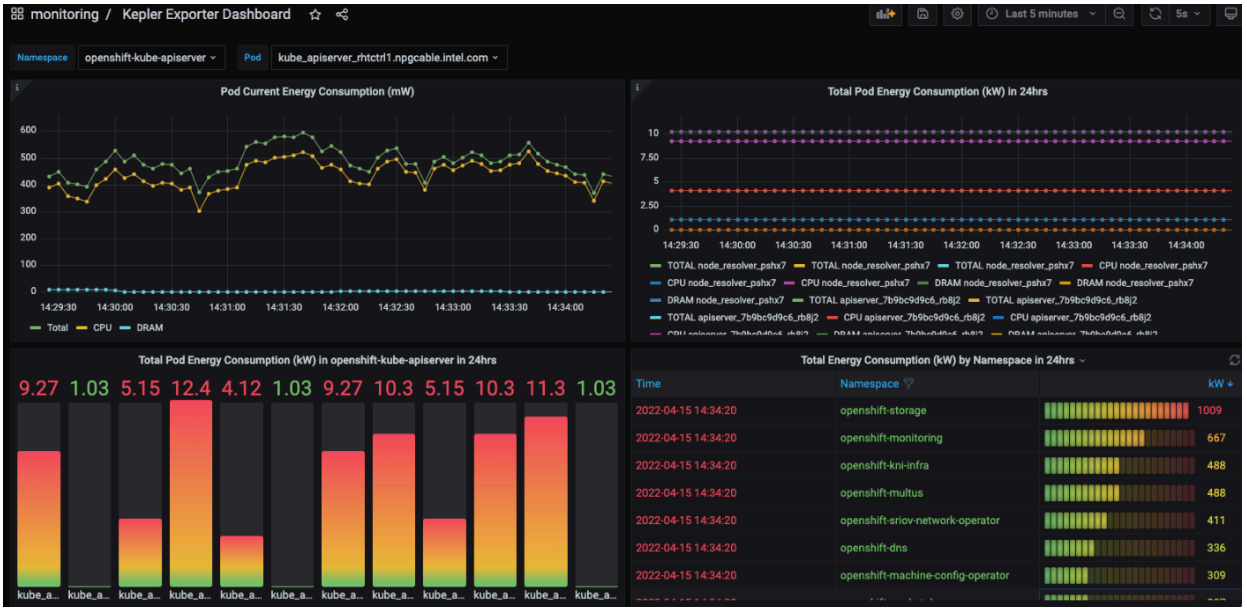

Figure 2: Kepler Dashboard (Grafana) [2]

Kepler architecture is designed to be extensible, enabling industrial and research projects to contribute novel power models for diverse system architectures. Kepler architecture is demonstrated in Figure 3.

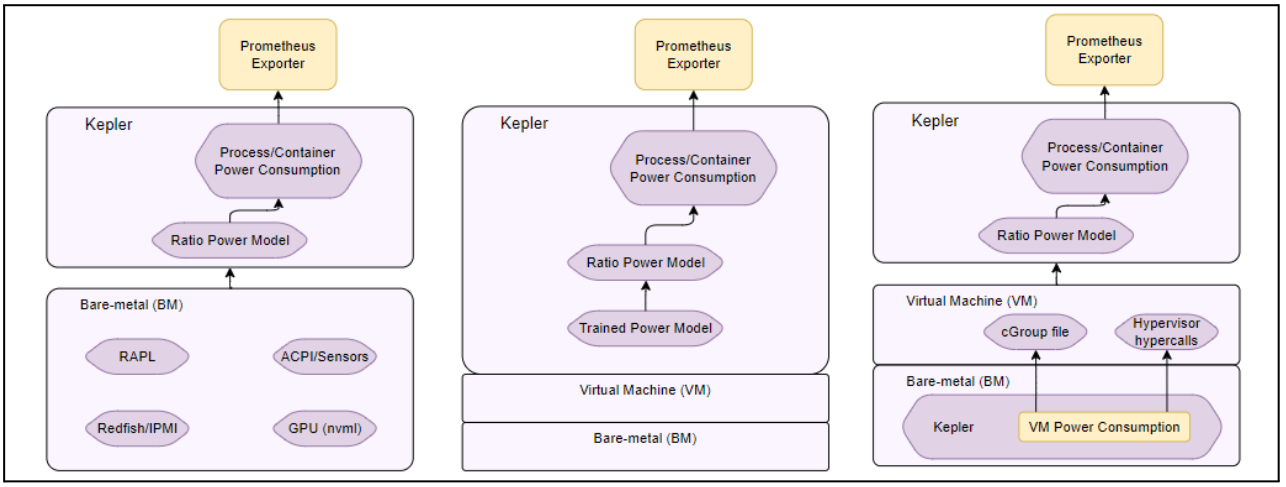

Figure 3: Kepler Architecture on Bare-metal and Virtual Machines

Funded by the European Union

Kepler can measure the energy consumption of Virtual Machines (VMs) on public clouds, however, there is a notable distinction between Bare-metal and VMs. Bare metal nodes provide power metrics directly from their hardware components, such as the CPU and DRAM. For instance, in x86 machines using RAPL, or ACPI. In contrast, VMs do not expose power metrics. The primary reason behind this is the absence of mechanisms to make these metrics available. This divergence leads Kepler to employ different approaches for these two distinct scenarios [3].

1.1.2 Scaphandre

Scaphandre is a metrology agent dedicated to electric power and energy consumption metrics, it exports the metrics to Prometheus, like Kepler. The project aims to enable any company or individual to measure the power consumption of its tech services and get this data conveniently, sending it through any monitoring or data analysis toolchain [4].
Scaphandre has a dashboard available on Grafana to visualise the metrics, Figure 4.

Figure 4: Scaphandre Dashboard (Grafana) [5]

Like Kepler, Scaphandre also works on Kubernetes, which makes it a valuable tool to implement on TEADAL. It can store all the power consumption metrics in a JSON or CSV file and supports most existing operating systems (Gnu/Linux, Windows 10, 11, and Server 2016/2019/2022).
It can also measure power consumption metrics from VMs, exposing the virtual machine's power/energy metrics, to allow manipulating those metrics in the VM as if it were a bare metal machine (relies on hypervisor features).
It's possible to better understand Scaphandre architecture with Figure 5.

Funded by the European Union

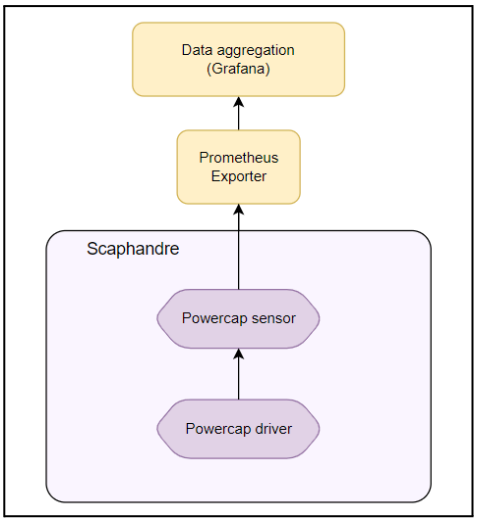

Figure 5: Scaphandre Architecture

1.2 Energy Aware Data Lake

Tools like Kepler or Scaphandre can be deployed inside the TEADAL node to make an energy-aware Data Lake. They are going to probe the node and find all the processes/pods running inside that consume energy.
Figure 6 makes it possible to understand the desired energy-aware system architecture. A similar system can be developed to achieve an energy-aware Data Lake.

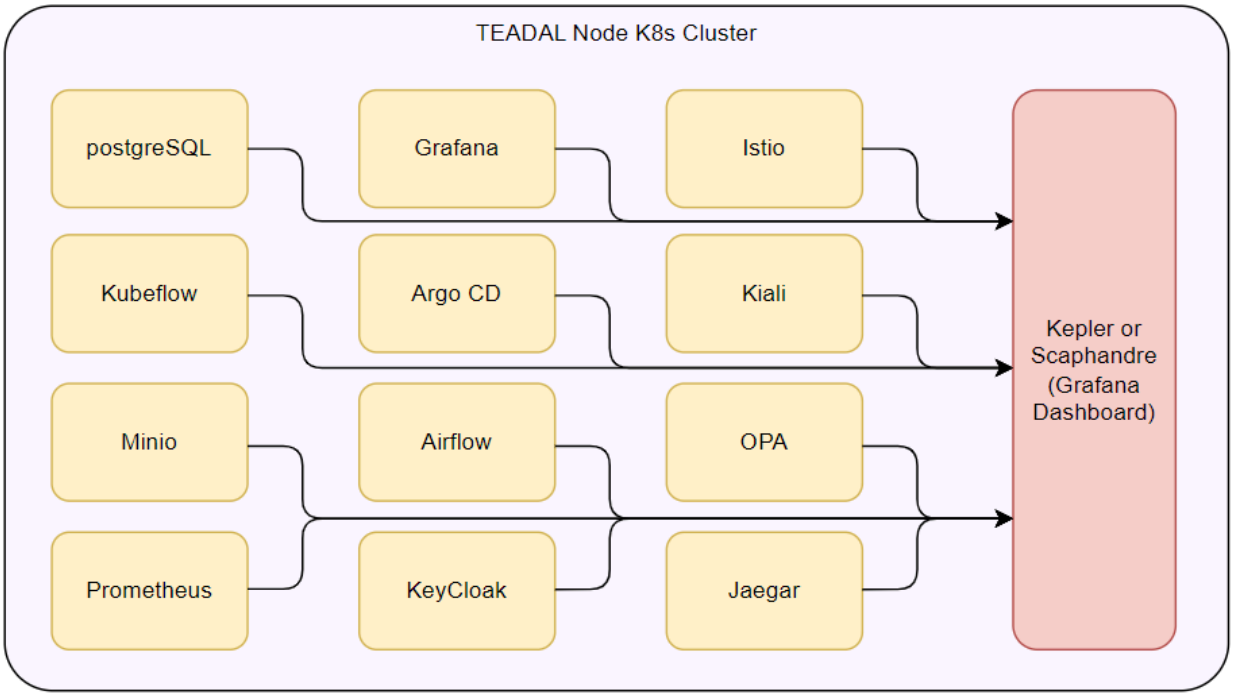

Figure 6: Energy-aware TEADAL

Funded by the European Union

The energy monitoring tools will always monitor the processes within TEADAL. These tools let the user choose which namespace, pod, or process of TEADAL will be monitored, and later, it's possible to save all the metrics recorded.
With the recorded metrics, it becomes easier to study and assess the energy consumption of a Data Lake, in our case, of TEADAL.

After installing the right tools to measure power consumption, a study should start on this matter. If Kepler or Scaphandre are used, the metrics can be recorded in a JSON or CSV file, making comparing values easier, as shown in Figure 7, if other tools are chosen different files can be produced.

Time	argocd	default	istio-sys	kernel	kubeflow	minio-ope	monitorin	system
20/05/2024 12:58	0.0213	0.0105	0.0109	0.00424	0.0128	0.00492	0.0180	0.150
20/05/2024 12:59	0.0213	0.0105	0.0109	0.00425	0.0128	0.00492	0.0179	0.151
20/05/2024 12:59	0.0214	0.0106	0.0109	0.00425	0.0129	0.00494	0.0180	0.151
20/05/2024 12:59	0.0214	0.0106	0.0110	0.00426	0.0129	0.00495	0.0180	0.151
20/05/2024 12:59	0.0215	0.0106	0.0110	0.00426	0.0129	0.00496	0.0181	0.152
20/05/2024 13:00	0.0215	0.0107	0.0110	0.00427	0.0130	0.00497	0.0181	0.152
20/05/2024 13:00	0.0215	0.0107	0.0111	0.00427	0.0130	0.00498	0.0181	0.153
20/05/2024 13:00	0.0216	0.0107	0.0111	0.00428	0.0130	0.00500	0.0182	0.153
20/05/2024 13:00	0.0216	0.0108	0.0111	0.00428	0.0131	0.00501	0.0183	0.153
20/05/2024 13:01	0.0216	0.0108	0.0111	0.00429	0.0131	0.00502	0.0183	0.154
20/05/2024 13:01	0.0217	0.0108	0.0112	0.00430	0.0131	0.00503	0.0183	0.154
20/05/2024 13:01	0.0217	0.0108	0.0112	0.00430	0.0131	0.00504	0.0184	0.154
20/05/2024 13:01	0.0217	0.0109	0.0112	0.00431	0.0132	0.00506	0.0184	0.154
20/05/2024 13:02	0.0218	0.0109	0.0112	0.00432	0.0132	0.00506	0.0184	0.155
20/05/2024 13:02	0.0218	0.0109	0.0113	0.00432	0.0132	0.00508	0.0185	0.155
20/05/2024 13:02	0.0218	0.0109	0.0113	0.00433	0.0133	0.00509	0.0185	0.155
20/05/2024 13:02	0.0219	0.0109	0.0113	0.00433	0.0133	0.00511	0.0186	0.156
20/05/2024 13:03	0.0219	0.0110	0.0113	0.00434	0.0133	0.00511	0.0186	0.156
20/05/2024 13:03	0.0219	0.0110	0.0114	0.00435	0.0134	0.00513	0.0186	0.157
20/05/2024 13:03	0.0220	0.0110	0.0114	0.00435	0.0134	0.00514	0.0187	0.157

Figure 7: Kepler/Scaphandre
Power Consumption Metrics (kWh per day)

With all the recorded metrics it's possible to start testing different solutions that can make the Data Lake more efficient.

1.3 Estimating Energy Consumption

In the previous section, we have analysed existing tools that allow the direct measurement of the energy consumption of TEADAL components supporting the Federated Data Lake in the whole data lifecycle. As can be observed, these tools have some limitations: (i) they provide an estimation of the real energy consumption based on the resource usage of the containers in which they are installed, since actual probes are not usually available in the servers; (ii) they are limited to a subset of the aspects we aim at monitoring (i.e., computational components).

As shown in Fig. 1 (Data Lake Architecture), to ensure an energy-aware data lake management in TEADAL, several aspects need to be monitored:

Funded by the European Union

- Data Storage: the cost of storing the FDPs and, if needed, the sFDPs made available in TEADAL;
- Data Processing: the cost of the computation required to transform the FDP in the sFDPs required by the different consumers, building and executing a proper data pipeline generated according to the specified policies;
- Data Transmission: the cost of moving the data from a location to another in the stretched data lake, also in between different steps of the data processing pipeline.

In order to obtain a full monitorability of the TEADAL solution, all these aspects need to be considered. Since tools providing a full solution to this issue are not available, we propose an hybrid approach in which we combine direct monitoring with estimations. Estimation is also needed when a direct monitoring solution is available in order to compare different possible configurations and select the most efficient one before the actual deployment of the nodes in TEADAL. As an example, the availability of estimation tools for data storage energy impact can help in evaluating in which storage node to store an FDP and in which format (raw, compressed) to store the data. Similarly, estimation tools for data processing enable us to select the best combination and configuration of the pipeline components if alternative solutions are available.

In this section we propose a preliminary version of an energy estimation model focusing on the most relevant aspects impacting the TEADAL solution: data processing, data storage, and data transmission.

1.3.1 Modelling data storage energy consumption

The energy impact of data storage depends on the size of the data to be stored and the energy efficiency of the node in which the data is saved [6]. In most cases, the information about the efficiency of the node is not available, since the exact location of the data is not disclosed. Given that our aim is not to provide an exact measurement, but to have all the tools to perform a comparison between different solutions and configurations, we disregard this information.

The model to estimate the energy consumption of data storage can be expressed as:

$$ES = h * sizest(FDP)$$

where $sizest(FDP)$ is the size in Gigabyte of the data stored, and h is a parameter expressing the energy cost in kWh/GB.

According to the Sustainable Web Design Model [7], parameter h can be estimated as $h = 0.059\ kWh/GB$.

1.3.2 Modelling data processing energy consumption

Data processing in TEADAL is carried out in Kubernetes pods. Estimating the energy consumption of this process involves determining the energy consumption of the pod execution and adding it to the consumption baseline of the platform. The energy impact of the pod execution depends on the specific task that the pod performs and the energy efficiency and status of the server in which it is executed. Modelling all the potential tasks and server configurations is not feasible following an analytic or semi-empirical method since it will require studying in detail many different topologies. For this reason, the strategy that has been considered in TEADAL is to use a data-driven approach based on machine learning.

Funded by the European Union

The dataset to create the machine learning model is generated capturing data from Prometheus while the Kubernetes pods are being executed in different servers. The trained machine learning model will return the estimated energy consumption of each pod as a function of different features such as the size of the pod, the CPU and RAM requested by Kubernetes, the occupation of the server in terms of CPU and RAM, or the type of task executed by the pod.

The model to estimate the energy consumption of data storage can be expressed as:

$$ES = f(x; \theta)$$

where x are the pod features extracted from Kubernetes and θ are the parameters of the trained machine learning model.

1.3.3 Modelling data transmission energy consumption

The energy impact of data transmission depends on the size of the data to be transmitted and the energy efficiency of the link on which the data is transmitted [1]. In most cases, the information about the efficiency of the link is not available. For this reason, in our model we approximate the estimation only considering the size of the data.

The model to estimate the energy consumption of data transmission can be expressed as:

$$ET = k * sizetr(FDP)$$

where $sizetr(FDP)$ is the size in Gigabyte of the data to be transmitted, and k is a parameter expressing the energy cost in kWh/GB.

According to the Sustainable Web Design Model, parameter k can be estimated as $k = 0.055\, kWh/GB$.

1.4 Conclusive remarks on monitoring and estimating energy in data lakes

In this section we have discussed how to assess the environmental impact of data management in data lakes, including data storage, data transmission, and data processing. TEADAL adopts a hybrid approach, in which part of the information is retrieved through monitoring services attached to specific architectural components, while other information is estimated using appropriate models which, starting from the monitoring data collected, impute the environmental impact to components that can't be directly measured.

The purpose of assessing the environmental impact of data lakes is twofold. First of all, an assessment is needed to enable energy-awareness. The collected data can be used inside TEADAL and by the different stakeholders to understand the impact of the data sharing services in terms of energy and emissions. This is crucial to allow a direct involvement of the stakeholders in the reduction of this impact. Secondarily, the collected data is necessary to enable improvements and what if analysis: through an adaptive approach, the different services of the data lake can adapt their execution according to the collected information about the environmental impact, aiming at reducing the emissions and/or the energy consumption. In the rest of the document, we are going to discuss how this information is exploited in TEADAL to enable adaptation.

Funded by the European Union

2. Energy Consumption Policies

2.1 Energy-Aware Placement and Scheduling in Trusted Execution Environments

In the era of big data, the integrity, confidentiality, and privacy of data are paramount. Traditional data pipelines, which involve the collection, processing, and analysis of large volumes of data, often face challenges related to security and privacy. This is especially true when it comes to the placement or scheduling of workloads across different computational environments. Privacy-Preserving Data Pipelines aim to address these challenges by integrating advanced security and privacy mechanisms, such as Trusted Execution Environments (TEEs), into pipeline workflows. The work developed and detailed in D5.2 introduces the architectural framework and implementation strategies for these privacy-preserving data pipelines.

TEEs provide a secure area within a processor, ensuring that data and code are protected while being processed. This isolation prevents unauthorised access and tampering, thus preserving the confidentiality and integrity of sensitive data. Incorporating TEEs into data pipelines makes it possible to execute confidential computations securely, even on less trusted hosts [8]. This approach not only enhances data security but also uncovers new possibilities for optimising resource utilisation and energy consumption without compromising data privacy.

2.1.1 TEE Technologies and Performance Overhead

For example, Intel Trusted Domain Extensions (Intel TDX) and similar technologies from other vendors, such as AMD SEV, Arm TrustZone, and Nvidia, provide robust security for virtual machines and sensitive workloads in various environments, ensuring data protection and integrity across cloud and multi-tenant platforms. These approaches are further listed and detailed in D5.2.

The performance overhead of TEE technologies, such as Intel TDX and AMD SEV, is to be taken into consideration for their adoption. Intel TDX, for instance, has been reported to introduce a performance overhead of up to 4.5% in terms of CPU usage [9, 10]. This overhead can be attributed to the additional security operations required to ensure the integrity and confidentiality of data within the TEE. AMD SEV has also been found to have a minor performance overhead, ranging from 2% to 8% [11]. However, both TEE technologies are designed to balance security and performance, often using specialised hardware to minimise the impact of additional security operations. This means that while there may be some performance overhead, it is generally optimised to ensure efficient execution of sensitive code within the TEE [10, 11].

2.1.2 Kubernetes-Based Pipeline Engines and Integration with TEE frameworks

Kubernetes already provides powerful tools and abstractions for deploying, scaling, and operating application containers. In the context of data pipelines, Kubernetes-based engines like Argo Workflows [12] and Kubeflow [13] are particularly relevant. Argo Workflows and Kubeflow can enhance data security by utilising TEEs for sensitive task execution, ensuring data confidentiality and integrity. Key Kubernetes concepts like node affinity labels and tolerations are essential for effectively leveraging TEEs within a cluster. These mechanisms allow precise workload placement based on hardware capabilities, as specified in

Funded by the European Union

deliverables D5.2 and D4.2. Projects like KubeVirt [14], Kata Containers [15], and Confidential Containers [16] extend Kubernetes to facilitate secure computations within TEEs. They leverage Kubernetes concepts like node affinity, tolerations, and taints to ensure sensitive workloads are executed in secure environments, enabling, for instance, secure data processing and model training in pipelines, protecting sensitive data and computations.

2.1.3 Combining Security with Energy Efficiency

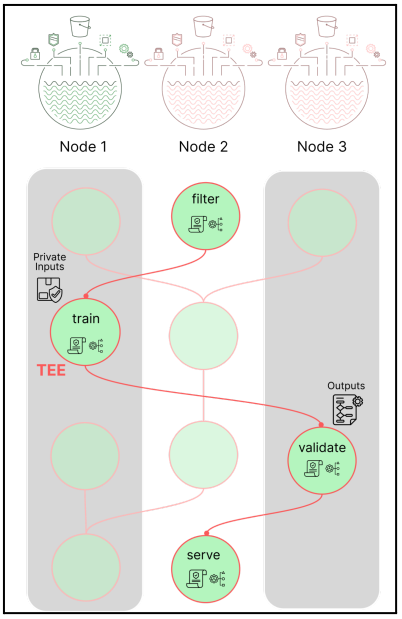

Figure 8: Illustration of Privacy-Preserving Data Pipelines leveraging TEEs. Energy-intensive sensitive computations are strategically placed in nodes optimised for energy efficiency, while safeguarding the privacy of the tasks and data.

Ensuring data and code integrity and confidentiality via hardware isolation prevents unauthorised access and tampering, guaranteeing that only authorised code can process the data. This makes TEEs ideal for maintaining data security and reliability during its processing. The lifecycle of sensitive data in TEE-based environments involves several phases. Initially, data is created in a trusted environment. During transit to the TEE, it is protected through encryption and secure communication channels. Once inside the TEE, hardware-based guarantees ensure data remains secure during processing. If data needs to be stored, it is encrypted outside the TEE with robust key management processes. When retrieving outputs, secure authentication mechanisms ensure only authorised access, preventing data leakage or inference.

Integrating TEEs within data pipelines not only enhances security and privacy but also allows to orient the efforts of aligning with energy efficiency strategies in pipeline and task deployment and execution. To optimise task placement and scheduling while maintaining security, TEEs may pair with dynamic placement and scheduling strategies, allowing tasks to run in energy-efficient hosts, during low energy costs or when renewable energy is available. TEEs and the code running inside them may support CPU idling, task suspension and resumption, cooperative or non-preemptive tasks, event loops, and other forms of suspendable computation that could dynamically match energy consumption schedules and strategies, to ensure continuous data protection, while data is at rest or its processing is suspended. This enables, for instance, the execution of energy-intensive sensitive

Funded by the European Union

computations, which can be strategically scheduled on nodes optimised for energy efficiency, like renewable energy-powered data centres or cloud platforms, or during off-peak hours, ensuring high security standards without compromising sustainability goals, and vice-versa. Furthermore, by leveraging TEEs and intelligent placement strategies, organisations can effectively offload computations from edge nodes to such data centres, maintaining both security and energy efficiency. This approach allows organisations to process large volumes of data securely and efficiently, without having to compromise privacy for environmental sustainability.

While the integration of TEEs into data pipelines presents numerous advantages, it also comes with its own set of challenges. Ensuring compatibility between different TEEs and existing pipeline infrastructure can be complex. Additionally, the overhead associated with secure execution in TEEs may impact performance, necessitating careful optimization. Future research in this area shall focus on developing standardised frameworks for the integration of TEEs into energy-aware data pipelines. This includes developing tools for seamless scheduling and workload placement, as well as optimising the performance and computational execution of TEEs to minimise overhead, while coordinating with overall cluster and multi-cluster energy-aware optimization efforts. Furthermore, exploring the potential of data pipeline integration with other privacy-preserving technologies, such as homomorphic encryption, differential privacy, or zero-knowledge, could lead to even more robust solutions for security and privacy of energy-aware data pipelines.

2.2 Enriching Policies with Energy-Aware Metadata

Security policies are the statement of top management intent on how to protect data lakes and ensure the security and privacy of sensitive data. Security policies are materialised in documents that describe the directions, responsibilities, and technical security solutions to achieve their mission. Trusted Execution Environments (TEEs) mentioned in the above section are examples of security mechanisms that can be employed to satisfy security policies. The creation and revision of security policies involves the top management, which gives the vision and supports managerial choices in what it is called top-down security, and security experts that provide knowledge of technical security solutions, in what it is called bottom-up security [17]. Notably, top-down security evaluates aspects as security objectives of data lakes and security responsibilities of the actors involved, while bottom-up security evaluates security procedures and how they can be integrated in pipelines executed in data lakes, and technical measures that need to be deployed.

In both the top-down and bottom-up approaches, the creation of security policies include a strenuous negotiation that considers the tradeoff between the (cyber)security that must be granted and a multitude of aspects that characterise complex systems as the one designed and deployed in the TEADAL project. The energy consumption aspect faced in this deliverable is one of the prominent aspects that need to be considered since cybersecurity, considered as the set of security mechanisms employed to meet security policies, is considered liable for a relevant part of IT consumption [18]. In light of these considerations, an energy aware design of security policies will have a great impact on energy consumption and it will help reduce the energy footprint of the system developed in this project.

To allow an energy aware design of security policies, this section proposes a method for the creation of green security policies, i.e, a method to support the design of security policies that includes energy consumption as a first citizen selection criteria for security measures and security goals.

Funded by the European Union

The proposed method approaches security policies by dividing them in three conceptual levels, each of them addressing different aspects of security of data lakes.

The first level addresses security goals, i.e., the security-related objectives of each actor involved in the data lake. Security goals are refinements of a more general security strategy and, thanks to this method, they report an aggregated value of their energy consumption. For example, the energy consumption of a security goal, such as confidentiality of sensitive data, will be estimated and reported.

The second level defines the security measures, such as encryption algorithms or firewalls, that are deployed and used in data lakes. These security measures are intended at large, and include security details for technical systems, e.g., the communication protocol used between two hosts. Security measures are characterised by the properties of their energy consumption.

The third level consists of pipelines that are executed in data lakes, where security measures are deployed to guarantee security properties and, therefore, to execute secure processes. Examples of pipelines that use security measures include all processes that store and manipulate sensitive data of patients of hospitals, or processes executed to manipulate the results of experiments and analyses on clinical data.

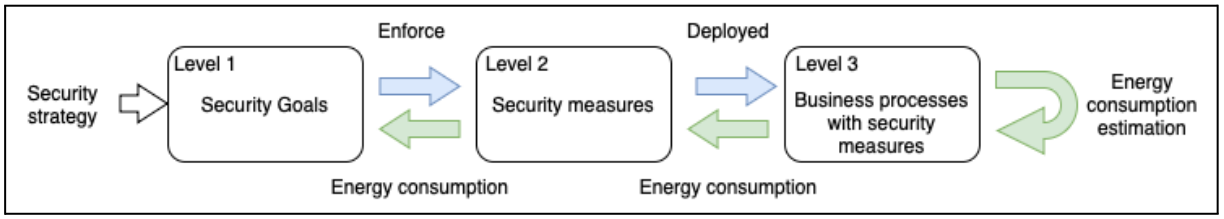

Figure 9: Conceptual modelling levels and their connections

Figure 9 shows the levels defined above and how they are logically connected. The security strategy of the socio-technical system is used to define the security goals that are enforced with security measures that are, in turn, deployed in pipelines. Alongside these connections, information on energy consumption is propagated among conceptual levels. At each level, information about energy consumption is shown. If enough data are provided, an estimation of energy consumption will be calculated at the third level, considering properties such as the quantity of data processed and the frequency of usage of security measures. This information will be propagated upward to the first level to identify the energy consumption of security goals, allowing security experts to use this information to modify security strategies and goals for less consuming ones.

Funded by the European Union

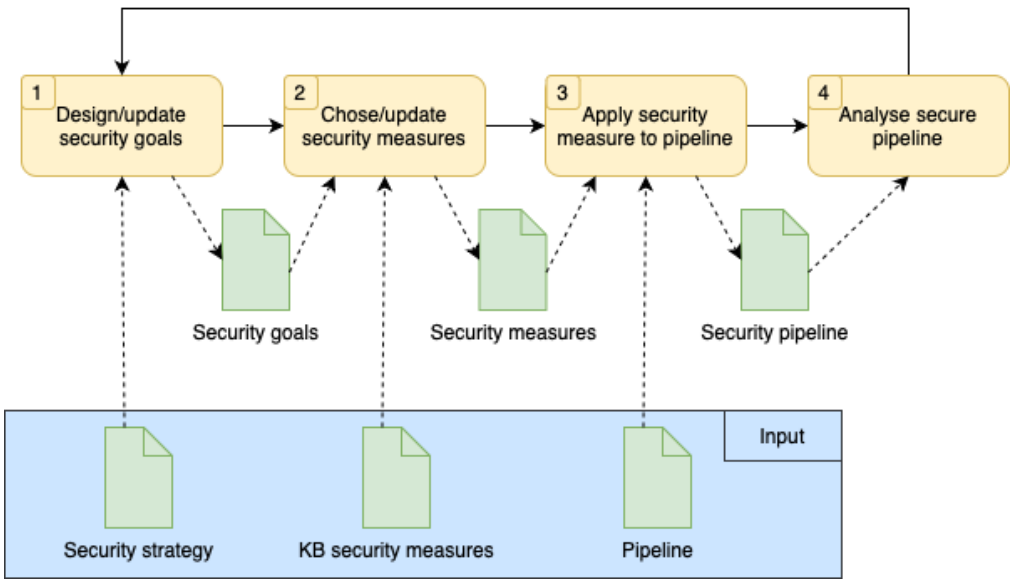

Figure 10: The process included in the method proposed in this section

We defined a process to guide target users on the design and analyses of green security policies conceptualised in the levels defined in Fig. 9. The process proposed by the method is depicted in Fig. 10. In light blue are highlighted the input of the method, i.e., a security document that highlights the security strategies of the organisation in charge of the data lake, the pipelines executed in the system and the knowledge-based as a list of security measures that can be deployed in the processes ("KB security measure" in the figure). The first step consists of the analyses of the security strategies and the definition of security goals. This creates a model of security goals (represented with the "Security goal" data object) that identifies the security objectives of each actor in the system and that is used by the second step where security measures are identified. The second step uses the knowledge base that can be deployed to satisfy security goals. This step creates a model of the security measures ("Security measure" in Fig. 10), which consists of a selection of measures to be deployed and the security goals enforced by each measure. This model is used for the third step where security measures, previously identified, are applied to pipeline elements executed in the data lake. Step 4 analyses the pipeline enriched with security measures and estimates their energy consumption. Using the conceptual connections between security goals, security measures and their application to pipelines, defined in the following sections, step 4 can also estimate the energy consumption of security goals. This allows to include energy consumption in the decision process of selecting which security goals to achieve, how to possibly modify the security strategy, or which security measure better fits the security goals while minimising energy consumption.

The process is iterative, allowing the modification of security goals, security measures, and their application to the pipeline. Moreover, this is compatible with incremental software development life cycles as the agile method. While on the first phases of the design of data lakes, only partial information on security strategies may be available, on later phases the security strategies and most of the pipelines will be available, allowing more precise outputs.

2.2.1 Modelling Conceptual Levels

In order to specify the concepts required at each conceptual level defined in Fig. 9, we propose two graphical modelling languages. A goal-based modelling language for the first two levels and one for pipelines for the third level.

Funded by the European Union

Goal based modelling language

Goal-based modelling languages are graphical modelling languages centred on the notion of goal [19, 20], that represents an objective that can be achieved. The top part of Fig. 11 shows a diagram obtained using a goal-based modelling language, where examples of Goals are “GDPR compliance” or “Prevent impacts”. The foundational concepts of goal-based modelling languages have been extended with the concepts and relations needed to model energy consumption properties and security measures. The goal based modelling language proposed in this deliverable takes inspiration from existing languages such as SecTRO [21] and BIM [22]. Unfortunately, none of these languages have all the relations and concepts needed, while including other ones that are not relevant for this deliverable.

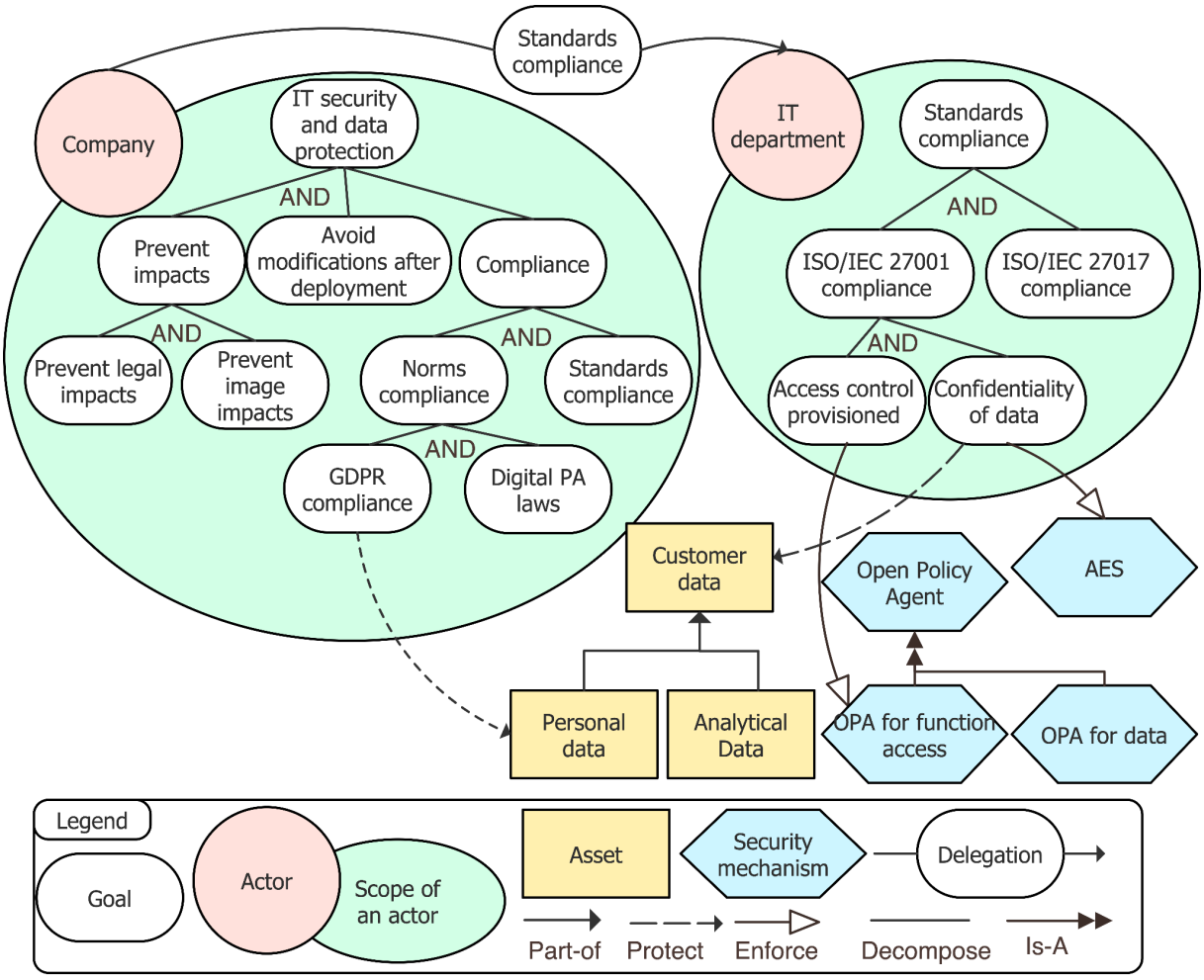

Figure 11: Example of a model representing a security policy

We included the elements of goal-based modelling languages as goal, goal decomposition, actor and delegation, and included the following security concepts.

- **Security measure**: it represents a method that enforces security goals. This concept is based on the concept of security mechanisms of SecTRO.
- **Asset**: it represents any valuable resource in data lakes, e.g., a piece of data, a data source, a physical computing node, etc. Fig. 11 shows three assets: “Customer data”, “Personal data” and “Analytical data”.

Funded by the European Union

The following relations are included.

- **Enforce(Goal → Security Measure)**: it specifies that a security measure will satisfy a security goal. This is based on the relation between Goal and Plan in TROPOS [5] or Goal and Task in BIM [6]. For example, in Fig. 11 “OPA for data” security measure enforces the security goal “Access control provisioned”. Only one security measure can be linked to a security goal.
- **Protect(Goal → Asset)**: it specifies that a security goal aims to protect an asset. For example, in Fig. 2 “Access control provisioned” security goal protects the asset “Customer data”. Only one goal can be linked to an asset.
- **Part-of(Asset → Asset)**: it allows to specify that an asset is part of another asset, following the Mereology approach [23]. For example, in Fig. 11 “Personal Data” and “Analytical data” are part of “Customer data” asset.
- **Is-A(Security measure → Security measure)**: it allows to model specific instances of the same security measure. For example, in Fig. 11 “Open Policy Agent” is linked to “OPA for data” and “OPA for function access”.

The modelling language proposed uses the well-known concepts of goal-based research field and security to specify cybersecurity policies. It allows modelling security goals of actors identifying, therefore, the security responsibilities within the system. Each security goals can then be linked to one or more assets it needs to protect and to the security measure(s) that will be deployed to enforce the goals and, therefore, protect the assets.

The estimation of energy consumption of security measures will be calculated using the third conceptual layer. Following the Protect relations, the energy consumed to protect an asset can be determined, while following the Enforce relations and the goal decomposition, the energy consumption of security goals can be estimated. To support this analysis, the modelling language is further extended with a property that specifies the energy consumption, on all concepts.

These rules allow us to estimate energy consumption of security goals and assets, allowing us to evaluate the best security strategy to reduce energy consumption.

Pipelines modelling language

We chose to include pipelines in the proposed method since they represent the know-how of data lakes, i.e., they model how such systems manipulate data. To achieve the aim of this deliverable, pipelines modelled for the method need to include information on the security measures, specified with the modelling language defined in the previous subsection. Furthermore, to estimate energy consumption, information related to the execution frequency and amount of data that are fed to security measures need to be specified in the model. We choose to use BPMN 2.0 [24] modelling language to model pipelines as it is the most used language for processes. Moreover, BPMN allows to model both automated and non-automated tasks allowing the definition of fully automated pipelines, as a sequence of tasks executed to manipulate data, or semi automated pipelines, that can be used to model also organisational aspects as synchronisation of different actors involved in the execution. Yet, this modelling language does not include all properties required. We, therefore, extend it with the following properties: (i) security measures deployed in the process; (ii) frequency of execution on pipelines; (iii) size of data object representing data sets.

The following BPMN elements are extended with the properties listed before..

- **Business process**: the frequency of execution of the pipeline is included and may be specified, if known. The top part of Fig. 12 shows an example of an extended BPMN

Funded by the European Union

diagram with the frequency of execution reported. For each exclusive, inclusive or event based gateway a distribution of probability of execution of outgoing control flows will need to be provided.

- **Task**: the security measure(s) that are activated and an estimation of their energy consumption can be specified.
- **Data object**: the security measure(s) that are activated when the data object is used. If the data object is an electronic document, information on its size will be included. An estimation of energy consumed can be specified.
- **Message flow**: the security measure(s) that are activated when the message flow is activated. If the message exchanges an electronic document, its size will be included. An estimation of energy consumed may be specified.

These properties allow to determine how many security measures will be executed, on which task/data. With this information and knowing the energy consumption of a security measure, it is possible to estimate the overall energy consumption of a security measure within a process.

Figure 12 shows an example of a pipeline modelled using the extension of BPMN diagram proposed in the deliverable. The modelled pipeline is derived from the pipeline defined in Figure 2 of Deliverable 2.2 of the TEADAL project.

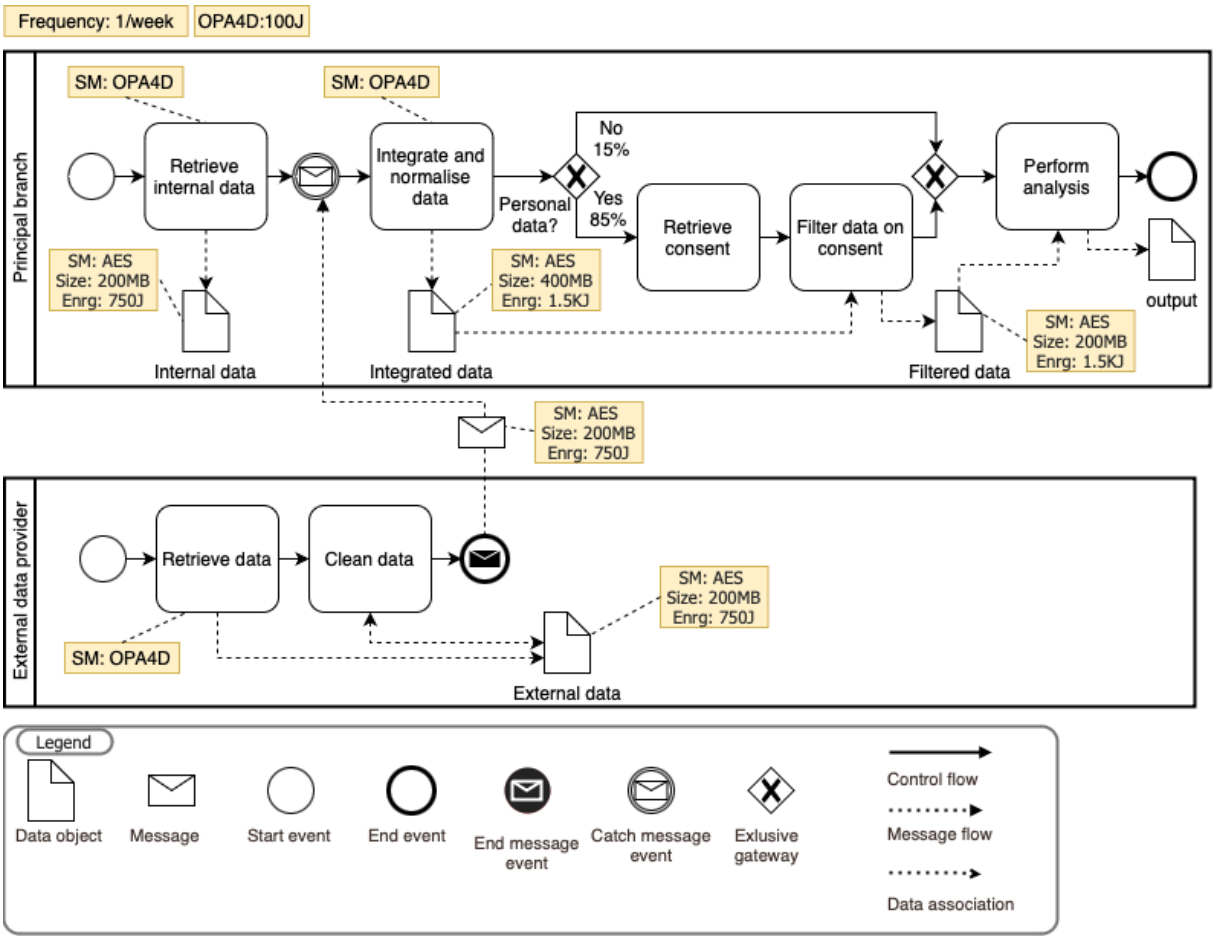

Figure 12: Example of a pipeline modelled using the BPMN 2.0 extension proposed in this section.

Funded by the European Union

If it is possible to estimate the energy consumption based on element execution, the information will be added on the property of the element. If it is possible to estimate the energy consumption per pipeline execution, the information will be added on the property of the model. If it is only possible to estimate the energy consumption of the security measure as a whole, then this information will be specified on the security measure in the goal model.

Mapping concepts

To support the propagation of energy consumption calculated in the pipelines to the security goals of the modelling language, we define a conceptual map between concepts of the two languages. We linked the concept of Security measure of the goal-based modelling language with the same concept of BPMN. This map allows to propagate energy consumption, if present, from the pipeline concept to the goal model one. The propagated values are the overall consumption estimated per process. We also mapped Asset with Data object and Message. Table 1 shows the mapping of concepts with the cardinality, while Table 2 shows the mapping between concepts of diagrams shown in Figures 2 and 3. “OPA for Data” is mapped to “OPA4D”, that is the OPA implementation. The “4D” part is added on the pipeline for mapping readability.

Goal-based concept	**Cardinality**		**Pipeline concept**
	->	**<-**	
Security measure	0:N	1:1	Security measure
Asset	0:N	0:N	Data object
Asset	0:N	0:N	Message

Table 1: Cardinality of mapping relations

Goal based concept	**Pipeline concept**
Personal data	Internal data
Personal data	External data
Personal data	Integrated data

Funded by the European Union

Goal based concept	Pipeline concept
Personal data	Filtered Data
Analytical data	Output
OPA for data	OPA4D
AES	AES

Table 2: Example of mapping relations

2.2.2 Conclusions

This section proposes an innovative method for the design of green security policies of socio-technical systems. The method guides security experts and sustainability experts in the design of security policies, considering energy consumption as a first citizen selection criteria of security goals to be achieved, and security measures to be adopted.

Funded by the European Union

3. Energy-Aware Data Governance

Following the methodology introduced in D3.1, the TEADAL approach to data governance is regulated by two main forces: data friction and data gravity. From an environmental perspective, these two concepts can be re-elaborated as follows.

Energy-aware data friction can be defined as the set of strategies supported and enacted for reducing the energy cost and/or environmental impact for sharing the data between the provider and the consumer.

Energy-aware data gravity can be defined as the set of strategies supported and enacted for reducing the energy cost and/or environmental impact for deploying and executing the pipeline components in the cloud continuum and for storing the Federated Data Products and the Shared Federated Data Products in such environment.

In the next sections we are going to list a set of possible strategies that can be supported by TEADAL using the concepts described in the previous sections of this deliverable, as well as the models and techniques introduced in D3.1 and D4.1.

3.1 Strategies for Energy-Aware Data Friction

Friction in data governance has been defined in D3.1 as the effort required to share the data between the data owner and the data consumer. Friction can be classified in static and dynamic.

Static friction relates to the definition of the data trading process plan. It relates to the operations that need to be applied in order to enable the consumer to access the data she needs while respecting security and privacy constraints. The result will be the selection of a pipeline of components to be executed in order to transform the FDP in the SFDP agreed with the consumer.

A pipeline p is defined as a set of capabilities c , i.e.:

$p = \{c_1, ..., c_n\}$

executed in a specific order. Linking this concept with the goal model introduced in Sect. 2.2, we can see the capabilities as the set of security measures that must be enacted in order to satisfy the goals. The goal model supports the data provider in the identification of the set of capabilities that need to be enacted as part of the pipeline. However, each capability can be implemented in several ways. For instance, if an encryption is required, this can be obtained using several encryption methods, each one providing a different result in terms of quality of service, performance, and energy consumption. Thus, we can see each capability c associated with multiple implementations:

$c = \{i_1, ..., i_n\}$

where each implementation is associated with a set of metadata describing its behaviour. Metadata for a capability implementation can be classified as follows:

$mi = <f, q, e>$

Funded by the European Union

where *f* represents a set of functional related metadata, describing the behaviour of the capability supported by the component; *q* are quality of service related metadata (e.g., response time, security level achieved, etc.); and *e* are energy related metadata, describing the energy consumption of the component.

Starting from an FDP, multiple sFDPs might be generated in order to share the same data with different consumers. Each sFDP will require the implementation of a dedicated pipeline. However, some of the preparation tasks might be common to multiple pipelines. Given two data pipelines p_1 and p_2, $p_1 \cap p_2$ corresponds to the shared capabilities. Reusing capabilities for multiple pipelines can be convenient from an environmental perspective by enabling to execute a subset of the tasks only once for multiple requests, thus reducing the computational power required for the computation.

Several strategies can be enacted to reduce the environmental impact of a pipeline in this context:

- **S1 - Energy-driven capability implementation selection:** the metadata describing the energy impact of each capability implementation can be used to select the best implementation in a given context of execution. The selection phase needs to consider several aspects, including the quality of service provided by different implementations and the data size that might influence this performance as well as the energy consumption. Also, we expect that the way in which a capability is implemented might have an impact on other capabilities of the pipeline. A trade-off analysis needs to be executed to find the best combination.
  STRATEGY IMPLEMENTATION: a catalogue of capabilities will be managed in order to provide different tools to enact data transformations required by the different pipelines. Each capability implementation will be enriched in the catalogue with a set of metadata providing all the relevant information about their quality and their environmental impact. Each capability implementation will be associated with two scores: (i) a *fit for use* score, describing the quality provided by a specific implementation for the context of the pipeline; (ii) an *environmental impact score*, describing the expected energy consumption of enacting the capability with a specific implementation. These scores will be used to enrich the capability metadata. This information can be edited by the data provider implementing the capability but will be refined and improved starting from the data collected by the monitoring system during the capability enactment.
- **S2 - Reuse-first capability selection:** sharing capabilities implementations between multiple pipelines can positively impact on the energy consumption of the overall data sharing. However, one specific capability implementation can be good for a pipeline while sub-optimal for another one from a quality perspective. A trade-off between advantages and disadvantages needs to be considered in this decision.
  STRATEGY IMPLEMENTATION: a functionality will be implemented to optimise the deployment of multiple pipelines. The functionality will compare the set of capabilities that the pipelines have in common and will update the scores of the capabilities implementation in order to incentivize the reuse of the same implementation by multiple pipelines deployments.

Dynamic friction refers to the effort required for the execution of the pipeline that has been selected in the trading process plan. In the TEADAL approach, each pipeline capability will be executed in a dedicated container deployed in a Kubernetes environment. As described in Sect. 1, it is possible to monitor the actual energy consumption of the execution of a container. The energy consumption will depend on several factors, including the capability implementation, the size of the data to be processed, and the node in the infrastructure where the capability will be executed. If we fix this parameter, we can consider the energy

Funded by the European Union

consumption of a capability and, composing all the capabilities, of the whole pipeline as invariant. This is not true if, instead of focusing only on the energy consumption, we consider the environmental impact of the computation (e.g., carbon emissions). Given a location, carbon emissions change over time, making it more or less convenient to execute a task in a specific node at different times. Assuming that the tasks composing the pipelines are not time sensitive (with some limitations), we can reduce the environmental impact of the pipeline execution by defining the following strategy:

- **S3 - Renewable-aware pipeline execution:** the execution of one or more capabilities of the pipeline can be postponed according to the monitored and predicted energy-mix of the node in which the capability has been deployed.
  STRATEGY IMPLEMENTATION: exploiting existing libraries to know the current energy mix of a specific geographical location, the implementation of this strategy will provide suggestions to the pipeline optimization (WP 4) about when to execute a specific pipeline capability, considering executions deadlines and predictions about renewable availability in the considered location.

3.2 Strategies for Energy-Aware Data Gravity

TEADAL supports data sharing in the cloud continuum. The nodes in which the data are stored and/or are processed through the pipeline can be located in different geographical locations and might be controlled by different actors (e.g., the data owner and the data consumer). As already discussed in D3.1, this implies that the pipeline deployment can be spread between different nodes as follows:

Given a data pipeline $p = \{c_1, ..., c_n\}$, the set $\leftarrow p$ represents the portion of the pipeline that will be deployed at the provider side, while the set $p \rightarrow$ represents all the capabilities that will be deployed at the consumer side. The selection of the set of nodes in which the pipeline component can be deployed can be based on:
(i) resource availability: each FDP/sFDP has a data volume as well as each pipeline capability implementation requires a specific amount of computational resources to be executed. In order to be able to store an FDP/sFDP in a specific node or to execute a capability, the selected node must have enough resources;
(ii) security constraints: privacy and security constraints might require that specific data in a specific format cannot be moved and analysed outside a set of authorised resources. To move a capability or a sFDP outside the provider resources, some conditions might be met or additional transformation might need to be executed;
(iii) environmental impact: each node is characterised by different computational/storage performance (energy required to execute a task/store data) as well as a different energy mix due to the location in which the node is placed.

Several strategies can be enacted to reduce the environmental impact of a pipeline in this context:

- **S4 - Energy-driven capabilities deployment:** the information collected through the monitoring system about the energy-mix/carbon footprint of each node where the pipeline components can be deployed can be exploited to decide where to place each pipeline component, while considering security and privacy constraints.
  STRATEGY IMPLEMENTATION: For each of the pipeline capabilities, a ranking of the nodes in which to deploy the container implementing it will be provided. The list of nodes can include provider side as well as consumer side nodes. In this second case, the privacy and security constraints might be met to allow the deployment. This list will be added to the Resource Inventory to be used by the Stretched Data Lake

Funded by the European Union

Compiler (WP4). The set of nodes can be extended exploiting Trusted Execution Environments (TEE) introduced in Sect. 2. This can enlarge the list of possible deployment solutions (enabling to deploy a capability in a more efficient node) while adding an overhead.

- **S5 - Energy-driven pipeline deployment:** the pipeline can be seen as a collection of capabilities. While each capability has its own requirements and constraints, their deployment cannot be seen as independent. In fact, the elements of the pipeline will influence each other by exchanging data: the result generated by a capability is the input of the following one. If the components are deployed in different nodes, this generates a network traffic that has a cost in terms of energy. The mutual dependency of multiple capabilities needs to be considered when taking deployment decisions.
  STRATEGY IMPLEMENTATION: To support the decisions on the deployment of a pipeline, a model for estimating the overall environmental impact of a specific deployment will be generated and implemented. This model will consider the dependencies between the different capabilities, their order, the amount of data they need to analyse, and the location in which the computation is executed. Also, suggestions about the dependency between capabilities will be generated. To this aim, the models introduced in Sect. 1.3 will be exploited.
- **S6 - Energy-driven FDP/sFDP storage:** Data Storage is another relevant aspect to consider from an energy perspective. Some considerations need to be made about if, where, and how to store the data. When the data sharing process starts with the generation and storage of an FDP. The FDP can be stored in several nodes, each one with a different energy cost and energy mix. Also, data can be stored as raw data or as compressed data. Compression can reduce the data volume, and as a consequence the environmental impact of storage. However, compression and decompression activities have a cost. The trade-off between the two options has to be considered. From an FDP, multiple sFDPs can be generated, each one for a specific consumer. The sFDP can be generated on the fly (when a data access request is generated) or can be stored in a permanent way for being queried multiple times. Additionally, intermediate sFDPs might be generated in order to share some common pipeline tasks between multiple users.
  STRATEGY IMPLEMENTATION: a model for estimating the cost of storing an FDP/sFDP on a specific node will be provided enabling the comparison between different deployment solutions. A set of guidelines about when to compress or not tocompress data, and when to create or not to create a persistent copy of an sFDP will be generated to be used by components developed in WP4.

Funded by the European Union

4. Employing ZK SLA Monitoring for Environmental Agreements

As explained in D5.2, Zero-Knowledge Proofs (ZKPs) offer robust mechanisms for privacy-preserving and selective disclosure of information. This chapter explores the application of ZKPs in monitoring environmental Key Performance Indicators (KPIs) within the framework of Service Level Agreements (SLAs), focusing on energy efficiency and sustainability targets. Monitoring environmental KPIs often involves handling sensitive data, such as proprietary energy consumption metrics, emission levels, or other operational details that organisations may not wish to disclose fully. Zero-Knowledge SLA Monitoring provides a solution by allowing only the necessary information to be disclosed, in order to evaluate, compute, or prove compliance with energy efficiency KPIs and thresholds. This approach falls within the scope of Privacy Preserving Data Pipelines, introduced in D5.2, ensuring data confidentiality and integrity throughout the monitoring process. The concept involves two primary roles:

- **Prover**: The entity that holds private data and needs to prove compliance with environmental KPIs.
- **Verifier**: The entity responsible for validating the compliance proofs without accessing the Prover's underlying private data.

For instance, in a scenario where a company needs to prove its adherence to a carbon emission reduction target, the Prover can generate a ZKP that demonstrates compliance without revealing the actual emission data. In the context of environmental agreements, such as regulatory compliance for energy efficiency or sustainability commitments, ZK SLA monitoring can be particularly valuable. For example:

- **Energy Efficiency**: A company can prove that its energy consumption for a specific period is below a regulatory threshold without disclosing exact usage figures.
- **Emissions Reporting**: An organisation can demonstrate compliance with emissions reduction targets without revealing specific operational details.

4.1 Monitoring Process Definition and Integrations

Implementing ZK SLA Monitoring for environmental agreements involves several key steps to ensure seamless integration with existing tools and frameworks. As outlined in D5.2, the existing prototype implementation leverages advanced monitoring tools like OpenTelemetry and integrates with established dashboard technologies such as Grafana for proof verification and presentation. This approach enables a robust and scalable solution for environmental KPI monitoring, following industry standard monitoring tools, and can also be easily integrated and extended to support the tooling introduced above in this deliverable.

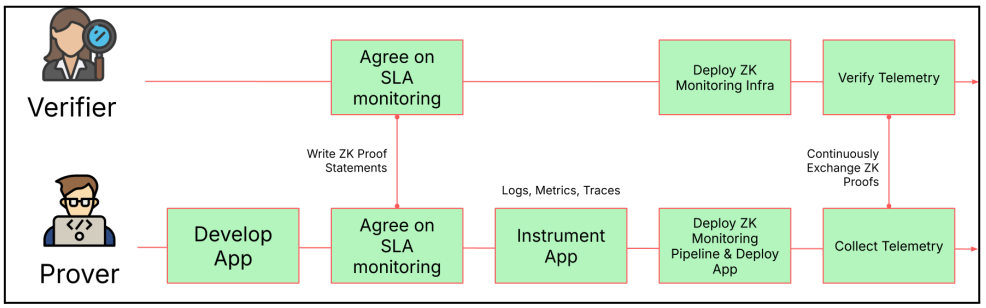

Figure 13: Workflow for implementing and integrating ZK SLA Monitoring technology.

Funded by the European Union

Figure 13 illustrates the workflow for onboarding this technology, which is further detailed in D5.2. These are the key steps following the implementation of ZK SLA Monitoring processes:

1. **Defining the KPI and Thresholds:** Establish the specific environmental KPIs and thresholds that need to be monitored, such as maximum allowable energy usage, emission levels, etc…
2. **Telemetry Data Collection:** Integrate telemetry tools (e.g., OpenTelemetry, as explained in D5.2, or the tools described above for energy monitoring) to collect relevant data continuously. This data remains within the Prover's domain, ensuring privacy.
3. **ZK Proof Generation:** Develop ZK coprocessors to process the telemetry data and generate ZK proofs that attest to compliance with the defined KPIs and thresholds. The detailed process for setting up these ZK coprocessors is discussed in D5.2.
4. **Verification Process:** The Verifier receives the ZK proofs and verifies them using tailored ZK verification tools. This process ensures that the Prover complies with the environmental agreements without disclosing sensitive data. Verification tools can be integrated with visualisation platforms like Grafana, allowing stakeholders to see compliance status in real-time.
5. **Auditability and Reporting:** Utilise state anchoring on a public blockchain for auditability, ensuring that compliance proofs are recorded and can be independently verified over time.

4.2 Enhancing Trust, Regulatory Compliance, and Resource Allocation

ZK SLA Monitoring for environmental agreements offers significant advantages, enhancing both operational efficiency and regulatory compliance. The primary benefit lies in the assurance of data privacy. Organisations can prove compliance with environmental standards without disclosing sensitive information such as specific energy consumption levels or proprietary operational details. This privacy-preserving approach builds trust among stakeholders, as it mitigates the risks associated with data exposure and potential misuse. Moreover, the integration of ZKP monitoring within existing frameworks like OpenTelemetry and Grafana streamlines the process of data collection, proof generation, and verification. Organisations can leverage familiar tools to implement these advanced monitoring techniques, reducing the learning curve and ensuring a smooth transition to the new system. This compatibility with established platforms not only simplifies implementation but also enhances the reliability and scalability of the monitoring solution.

In terms of regulatory compliance, ZKP-based monitoring provides a robust mechanism for demonstrating adherence to environmental regulations and standards. Organisations can generate verifiable proofs of compliance that can be independently audited and verified, thereby enhancing transparency and accountability. This capability is particularly valuable in sectors where regulatory scrutiny is intense, and the ability to demonstrate compliance quickly and accurately can lead to significant competitive advantages. Additionally, the use of blockchain technology for state anchoring ensures the immutability and auditability of compliance proofs. This feature adds an extra layer of security and trust, as it allows stakeholders to verify compliance history independently and prevents tampering with historical data. This transparency is crucial for building long-term trust and demonstrating ongoing commitment to environmental sustainability. Furthermore, ZK SLA Monitoring can lead to cost savings by optimising resource usage and reducing the need for extensive manual audits. Automated proof generation and verification processes streamline compliance monitoring, allowing organisations to allocate resources more efficiently. This efficiency gain is not only beneficial for regulatory compliance but also for internal sustainability initiatives,

Funded by the European Union

as it enables organisations to track and optimise their environmental performance continuously.

Work on ZK SLA Monitoring will continue in the next deliverables, with a focus on requirements gathering, practical implementation, and integration with other TEADAL tools. The goal is to showcase these ZK SLA Monitoring mechanisms in a real-world scenario, demonstrating their efficacy and value. This will involve close integration with the TEADAL node infrastructure to ensure that the solutions meet the needs of the existing scenarios and pilot cases, providing a comprehensive approach to privacy-preserving environmental KPI and SLA monitoring.

Funded by
the European Union

CONCLUSION

The deliverable D3.2 represents an effort in understanding the energy footprint of federated stretched data lakes. This initiative has resulted in the development of various innovative methodologies and tools aimed at better monitoring, estimating, and reducing the energy consumption associated with data storage, processing, and transmission.

1. **Energy Consumption in TEADAL**:
   The project examined energy consumption within TEADAL through the use of tools like Kepler and Scaphandre. These tools provide a comprehensive analysis of energy metrics within the data lakes, forming the foundation for understanding the complex interplay between data management and energy usage.
2. **Energy Consumption Policies**:
   A significant focus was placed on developing energy-aware policies that balance the requirements of energy efficiency with those of security. The deliverable explored strategies for optimal placement and scheduling in Trusted Execution Environments (TEEs), ensuring that security considerations do not compromise energy efficiency.
3. **Energy-Aware Data Governance**:
   This section reviewed strategies for managing data friction and gravity while maintaining energy efficiency. A key outcome was the creation of a decision system that drives the execution of data pipelines with energy consumption in mind, promoting sustainable data operations.
4. **Employing ZK SLA Monitoring for Environmental Agreements**:
   The deliverable also developed an approach for monitoring environmental agreements using ZK SLA monitoring. This ensures compliance with environmental standards and enhances trust in the system, all while optimising resource allocation.

These insights and methodologies are not just technical advancements but are aligned with broader environmental objectives. By integrating energy considerations into security policy design and data management practices, TEADAL enhances the efficiency, cost-effectiveness, and sustainability of data operations.

As TEADAL moves forward, the collaborative efforts and technological advancements made in this project pave the way for a more sustainable future in data management. The project's alignment with global goals of reducing energy consumption and mitigating climate change marks it as a critical step forward in sustainable data management practices.

In conclusion, TEADAL not only addresses the technical challenges of managing federated data lakes but also contributes significantly to environmental sustainability, making it a comprehensive solution for the future of data management.

Funded by the European Union

REFERENCES

[1] Sustainable Computing I/O, "Kepler: Kubernetes Efficient Power Level Exporter," GitHub. Available: https://github.com/sustainable-computing-io/kepler.

[2] Sustainable Computing I/O, "Kepler: Readme Overview," GitHub. Available: https://github.com/sustainable-computing-io/kepler/tree/main?tab=readme-ov-file.

[3] CNCF, "Exploring Kepler's Potentials: Unveiling Cloud Application Power Consumption," CNCF Blog, Oct.11, 2023. Available: https://www.cncf.io/blog/2023/10/11/exploring-keplers-potentials-unveiling-cloud-application-power-consumption/.

[4] Hubblo Org, "Scaphandre: Power and Energy Consumption Metrics," GitHub. Available: https://github.com/hubblo-org/scaphandre/blob/main/README.md.

[5] Hubblo Org, "Scaphandre Metrics Dashboard," Metrics Hub. Available: https://metrics.hubblo.org/d/GOHnbBO7z/scaphandre?orgId=1&refresh=15m.

[6] IEEE, "Exploring Cloud Application Power Consumption," IEEE Xplore, 2018. Available: https://ieeexplore.ieee.org/document/8567702/.

[7] Sustainable Web Design, "Estimating Digital Emissions," Sustainable Web Design. Available: https://sustainablewebdesign.org/estimating-digital-emissions/.

[8] "Modeling Strategic Relationships for Process Reengineering," Frontiers in Computer Science, 2022. Available: https://www.frontiersin.org/articles/10.3389/fcomp.2022.930741/full.

[9] Intel Corporation, "Protecting Kubernetes Clusters in the Cloud with Confidential Computing and Intel Xeon Processors," Intel, 2023. Available: https://www.intel.com/content/www/us/en/content-details/783205/protecting-kubernetes-clusters-in-the-cloud-with-confidential-computing-and-intel-xeon-processors.html.

[10] Intel Corporation, "Trust Domain Extensions on 4th Gen Xeon Processors," Intel, 2024. [Online]. Available: https://www.intel.com/content/www/us/en/developer/articles/technical/trust-domain-extensions-on-4th-gen-xeon-processors.html.

[11] Edgeless Systems, "Constellation: Performance Overview," Edgeless Systems Documentation. Available: https://docs.edgeless.systems/constellation/overview/performance/.

[12] Argo Workflows, "Argo Workflows Documentation," 2024. [Online]. Available: https://argo-workflows.readthedocs.io/en/latest/.

[13] Kubeflow, "Kubeflow Pipelines V2 Introduction," Kubeflow Documentation. Available: https://www.kubeflow.org/docs/components/pipelines/v2/introduction/.

[14] KubeVirt, "User Guide," 2024. Available: https://kubevirt.io/user-guide/.

[15] Kata Containers, "Kata Containers Documentation," 2024. Available: https://katacontainers.io/docs/.

Funded by the European Union

[16] Confidential Containers, "Confidential Containers Overview," Confidential Containers Project. Available: https://confidentialcontainers.org/docs/overview/.

[17] J. Lubell, "Integrating Top-Down and Bottom-Up Cybersecurity Guidance Using XML," Balisage Series on Markup Technologies, vol. 17, 2016.

[18] A. Merlo, M. Migliardi, and L. Caviglione, "A Survey on Energy-Aware Security Mechanisms," Pervasive and Mobile Computing, vol. 24, pp. 77–90, 2015.

[19] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, "Tropos: An Agent-Oriented Software Development Methodology," Autonomous Agents and Multi-Agent Systems, vol. 8, no. 3, pp. 203–236, 2004.

[20] E. Yu, "Modeling Strategic Relationships for Process Reengineering," Social Modeling for Requirements Engineering, vol. 11, pp. 66–87, 2011.

[21] H. Mouratidis and P. Giorgini, "Secure Tropos: A Security-Oriented Extension of the Tropos Methodology," International Journal of Software Engineering and Knowledge Engineering, vol. 17, no. 2, pp. 285–309, 2007.

[22] J. Horkoff, A. Borgida, J. Mylopoulos, D. Barone, L. Jiang, E. Yu, and D. Amyot, "Making Data Meaningful: The Business Intelligence Model and Its Formal Semantics in Description Logics," in OTM Conferences, Springer, 2012, pp. 700–717.

[23] A. Cotnoir and A. C. Varzi, *Mereology*, Oxford University Press, 2021.

[24] OMG, "BPMN 2.0," Technical Report, Object Management Group, Jan. 2011. Available: http://www.omg.org/spec/BPMN/2.0.

Funded by the European Union