PROCEEDINGS OF SCIENCE

Confidential Computing for Sensitive Data Analytics: The EOSC-SIESTA Approach

Viet Tran[a],*, Alvaro Lopez[b], Andres Heredia[b], Judith Sainz-Pardo[b], Saul Fernandez[b], Marta Obregon[b], Peter Kissa[c], Martin Seleng[a], Jan Astalos[a], Ladislav Hluchy[a], Vincent Legoll[d], Jérôme Pansanel[d], Aida Palacio Hoz[b], Hans-Jürgen Wolf[e], Mikulas Strelecky[c], Pedro Castro[f], Juan José Velasco[f]

[a] *Institute of Informatics, Slovak Academy of Sciences, Dubravska cesta 9, 84507 Bratislava, Slovakia*

[b] *IFCA - CSIC - UC, Instituto de Física de Cantabria, Avda. de los Castros s/n, Santander, 39005, Cantabria, Spain*

[c] *InterWay a.s. Stara Vajnorska 21, 83104 Bratislava, Slovakia*

[d] *Université de Strasbourg, CNRS, IPHC UMR 7178, F-67000 Strasbourg, France*

[e] *Inserm 101 rue de Tolbiac, 75654 Paris Cedex 13, France*

[f] *Predictia Intelligent Data Solutions SL Fernando de los Ríos st., 48 bajo, Santander, 39006, Cantabria, Spain*

E-mail: viet.tran@savba.sk

The FAIR principles have become a cornerstone of the European Open Science Cloud (EOSC), promoting data sharing and reuse across scientific domains. However, enabling access to sensitive datasets—such as medical records, neuroimaging data, or demographic registries—remains a major challenge due to strict regulatory requirements and security concerns. In particular, data in use represents a critical vulnerability, as decrypted data in memory is exposed to privileged attacks that cannot be mitigated by conventional software-based protections. The EOSC-SIESTA project addresses this challenge by developing a distributed cloud platform based on hardware-enforced Trusted Execution Environments (TEEs), specifically AMD SEV-SNP. The platform integrates confidential computing with secure storage, identity management, auditability, network-isolated workloads, secure software supply chain controls, and DevSecOps practices to support compliant and scalable sensitive data analytics. This paper presents the motivation, reviews the state of the art, and describes the architecture and implementation of the EOSC-SIESTA platform. We demonstrate how the combination of hardware-based security and integrated services enables FAIR-compliant processing of sensitive data across distributed cloud infrastructures.

© Copyright owned by the author(s) under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (CC BY-NC-ND 4.0)
All rights for text and data mining, AI training, and similar technologies for commercial purposes, are reserved.
ISSN 1824-8039 . Published by SISSA Medialab.

PoS(ISGC2026)031

International Symposium on Grids and Clouds (ISGC2026)
15-20, March, 2026
Academia Sinica, Taipei, Taiwan

PoS(ISGC2026)031

1. Introduction

Open science mandates broad access to research data, yet a large and growing fraction of the open science paradigm promotes transparency, reproducibility, and collaboration through the sharing of research data. Within the European Open Science Cloud (EOSC), the FAIR principles (Findable, Accessible, Interoperable, and Reusable) provide a framework for enabling such data sharing. However, a substantial portion of scientifically valuable datasets contains sensitive information that cannot be openly shared due to legal, ethical, or intellectual property and proprietary constraints.

Regulatory frameworks such as the General Data Protection Regulation (GDPR) [1], the Data Governance Act [2], and the AI Act [3] impose strict requirements on the handling of personal data. These regulations require strong guarantees regarding data confidentiality, integrity, and access control, significantly limiting the usability of sensitive datasets in collaborative research environments.

Traditional privacy-preserving techniques, including anonymisation and pseudonymisation, aim to reduce disclosure risks. However, these techniques are often difficult to validate and may fail under advanced re-identification and linkage attacks. Consequently, many datasets remain inaccessible despite their potential value.

Cloud computing offers scalable infrastructure for data analytics but introduces additional risks. Sensitive data processed on third-party infrastructure may be exposed to privileged actors, including cloud administrators or compromised system components. This creates a fundamental tension between broad access and security.

The EOSC-SIESTA project addresses these challenges by leveraging confidential computing technologies to protect data during processing. By combining hardware-based Trusted Execution Environments (TEEs) with a comprehensive set of platform services, SIESTA enables secure, compliant, and scalable analysis of sensitive data.

The platform is designed around five representative use cases: epidemiology, neuroimaging, energy analytics, text anonymisation, and demography. In this context, three roles are available: Data Rights Holder, Data User, and Platform Operator.

This paper presents the design, architecture, and implementation of the EOSC-SIESTA platform. Section 2 reviews the threat landscape and existing approaches. Section 3 introduces the system architecture. Section 4 describes the confidential computing backbone and Section 5 details supporting components. Section 6 outlines the DevSecOps methodology, and Section 7 outlines the conclusions.

2.Motivation and State of the Art

2.1 The Three States of Data and the In-Use Gap

Effective data security must address data across all three states of its operational lifecycle [5]. Data at rest (i.e. stored in databases, archives, or backup media) is protected through storage-level encryption (AES-256), a mature and widely mandated practice. Data in transit (traversing public or private networks) is protected through transport-layer protocols such as TLS 1.3, which are similarly well-established. However, data in use (the state in which data are

PoS(ISGC2026)031

actively decrypted in volatile memory for computation), represents a fundamentally more difficult problem. The very act of processing requires that plaintext be exposed within the processor's memory address space, where it is accessible to any sufficiently privileged software entity on the same host.

In cloud environments, this exposure is particularly acute. A privileged actor with access to the host infrastructure (whether a malicious insider, a rogue or negligent infrastructure administrator, a compromised hypervisor, or a co-resident VM exploiting a hypervisor vulnerability) can compromise in-use data through several well-documented attack classes [5, 6]:

- Memory-dump attacks: direct extraction of the raw contents of a running VM's RAM by a privileged host process [7].
- Memory-replay attacks: re-injection of captured memory pages to manipulate the execution state of a victim workload [8].
- DMA attacks: exploitation of Direct Memory Access peripherals to read from or write to arbitrary physical memory regions, bypassing CPU memory protection [9].
- Side-channel attacks: inference of secret data through observable computational side effects, including cache-timing (Spectre [10], Meltdown [11]) and power-consumption signatures.
- Cold-boot attacks: recovery of encryption keys from DRAM chips following a power cycle, exploiting DRAM data remanence [12].

Addressing these classes of threats requires hardware-level memory isolation enforced below the OS and hypervisor beyond the reach of any software-based attacker, including the cloud operator. This is the capability delivered by Trusted Execution Environments (TEEs), and it constitutes the foundational security primitive of the SIESTA platform.

2.2 Existing Approaches to Sensitive-Data Computing

Prior to the widespread adoption of TEE-based approaches, several complementary paradigms were developed for sensitive-data computation.

Homomorphic Encryption (HE) [13] allows arbitrary computations to be performed directly on ciphertext, producing results that, when decrypted, match those obtained on plaintext. While theoretically powerful, fully homomorphic schemes (e.g., the Gentry construction [13]) incur computational overheads of several orders of magnitude. Practically efficient levelled schemes such as CKKS [14] support approximate arithmetic on real-valued data, but remain restricted to specific operation classes and are unsuitable for general-purpose analytical workloads.

Secure Multi-Party Computation (SMPC) [15] enables mutually distrusting parties to jointly compute a function over their private inputs without revealing them. SMPC protocols have been applied to privacy-preserving genomics [16], federated statistics, and auction mechanisms, but their communication complexity and the requirement for multiple non-colluding parties limit scalability in practical research environments.

Differential Privacy (DP) [17] provides a rigorous mathematical definition for bounding information leakage about individuals through statistical query responses. The goal is to provide each individual with approximately the same level of privacy as they would have if their data were deleted from the database analyzed. To this end, randomization techniques are used to limit

PoS(ISGC2026)031

the influence of each individual data point on the results. However, DP introduces a fundamental accuracy-privacy trade-off that constrains its applicability for high-precision scientific analyses requiring exact record-level computations.

Federated Learning (FL) [18] enables model training across decentralised data sources without centralising raw data, and has been extensively studied in the medical imaging domain [19] among others. While FL data transmission is used for training purposes, it does not eliminate the risk of data reconstruction: gradient inversion attacks [20] can be used to reconstruct training examples from model updates, and FL does not account for analytical workloads outside of training.

Trusted Research Environments (TREs) and Safe Havens [21], formalised through the five-safes governance framework [22], represent an organisational approach: sensitive data are made accessible only within physically or logically controlled environments subject to strict governance and output-checking. While TREs have achieved wide adoption in health research, they impose significant operational overheads, typically restrict the available toolset, limit reproducibility, and remain largely siloed, inhibiting cross-institutional collaboration.

Anonymisation techniques, including k-anonymity [23], l-diversity [24], t-closeness [25] among others, which are implemented in the ANJANA Python library developed within SIESTA [26], transform datasets to reduce re-identification, linkage, inference and related risks while preserving statistical utility. These techniques are necessary but not sufficient: they operate on static datasets, may not generalise to all data types, and may not protect against all inference attacks on the published output.

2.3 TEE-Based Confidential Computing

TEE-based confidential computing represents a qualitatively different approach that is orthogonal and complementary to all of the above. Rather than transforming data or constraining the computation, TEEs protect the computation environment itself at the hardware level, ensuring that both the data and the analysis code remain confidential even from the infrastructure operator. This approach was identified by the Confidential Computing Consortium [5] — a Linux Foundation project convening AMD, Intel, ARM, IBM, Microsoft, Google, Amazon, and others — as the key enabling technology for cloud-native sensitive-data computing.

TEEs do not incur the performance overheads of HE, do not require non-colluding parties as in SMPC, do not introduce accuracy loss as in DP, and are not limited to training workloads as in FL. They can, in principle, run any general-purpose computation inside a hardware-protected enclave. Their primary limitations are a hardware dependency (as security guarantees are tied to specific processor models) and a history of micro-architectural side-channel vulnerabilities (e.g., SGX-related Foreshadow [27]) that have been discovered and patched iteratively. The SNP extension of AMD SEV specifically addresses the most critical of these integrity vulnerabilities through Reverse Map Table protection.

The SIESTA platform builds on the TEE paradigm by selecting AMD SEV-SNP as the primary hardware technology, and by integrating it with the broader set of organisational, operational, and software-engineering controls required to build a production-ready research platform, addressing the governance, usability, and supply-chain integrity dimensions that hardware alone cannot resolve.

PoS(ISGC2026)031

3. Platform Architecture

This section presents the updated SIESTA platform architecture as formalised in D8.1, Section 5.1 [33]. The architecture builds on the initial design described in D6.1 [4] and the subsequent implementation refinements documented in D7.1 [32]. While the core design decisions remain valid, a number of services and modules were identified as desirable during the first platform release (WP7) and through structured use-case feedback collected during WP8. These additions are reflected in the three updated architectural views presented below, which constitute the reference architecture for ongoing implementation in WP9. The architecture is organised around three tiers:Secure Compute, Common Services, and Data, which are instantiated across three geographically distributed provider sites: CSIC/IFCA (Santander, Spain), CNRS/IPHC (Strasbourg, France), and IISAS (Bratislava, Slovakia).

3.1 System Landscape

Figure 1 presents the updated system landscape, depicting the two operational personas: Data Rights Holder, Data User (EOSC User), their interactions with the platform, and the principal external integrations. There is an additional persona not depicted in the diagram which is the Platform Operator, that has overall access and control of the whole platform. The EOSC-SIESTA Platform cloud boundary encloses three nested tiers. The Secure Compute Tier hosts Confidential VMs (AMD SEV-SNP) and Confidential Containers (CoCo/Kata) on OpenStack/bare-metal infrastructure. The audit system is composed of Google Trillian/Tessera, Wazuh and Grafana for monitoring. The Data Tier (SIESTA Secure Storage) encompasses LUKS/Vault-encrypted secure storage, the new FAIR-compliant Dataset Catalogue, and Nextcloud for file synchronisation and sharing. External integrations encompass the EOSC EU-Node and MyAccessID (future authentication federation), EGI Check-in (infrastructure-level AAI), and GreenDIGIT/wattnet (environmental sustainability monitoring). The Common Services Tier will be detailed in the next subsection

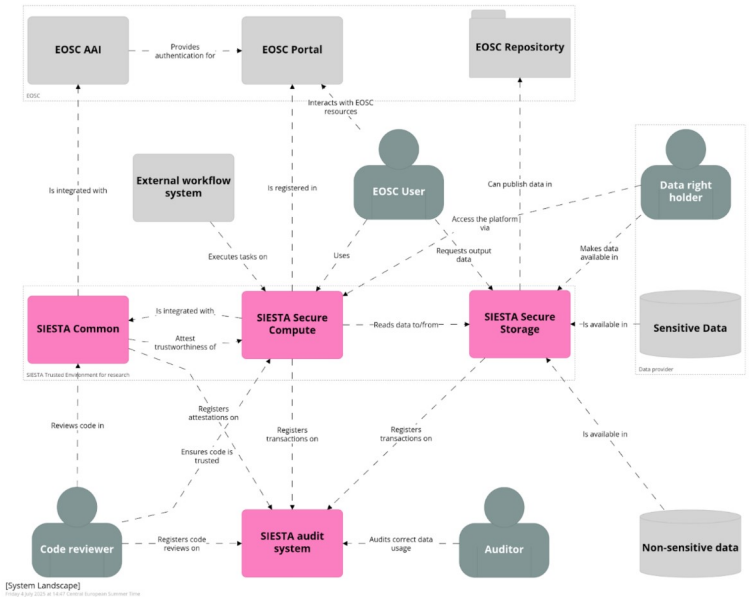

Figure 1. System landscape.

PoS(ISGC2026)031

3.2 Common Services

Figure 2 details the Common Services tier, grouping platform components by functional domain. Identity & Access encompasses Keycloak AAI (OIDC/SAML) and EGI Check-in for infrastructure AAI. Audit & Security comprises Google Trillian/Tessera for cryptographically verifiable audit logging, Wazuh/OSSEC for host intrusion detection. The software mirror and code repository is a self-hosted GitLab instance for source-code management and CI/CD automation using pipelines. Compute Management covers Kubernetes orchestration, the CoCo Operator for confidential container runtime management, and the Rook.io Operator for distributed storage. Developer Services includes Harbor with Trivy and Cosign, and the Onyxia portal. During the second half of the project, some additional services are added to the platform: Nextcloud, the Dataset Catalogue, GitLab-based issue tracker (helpdesk), the service catalogue (Jekyll/Backstage), and HashiCorp Vault for secrets management.

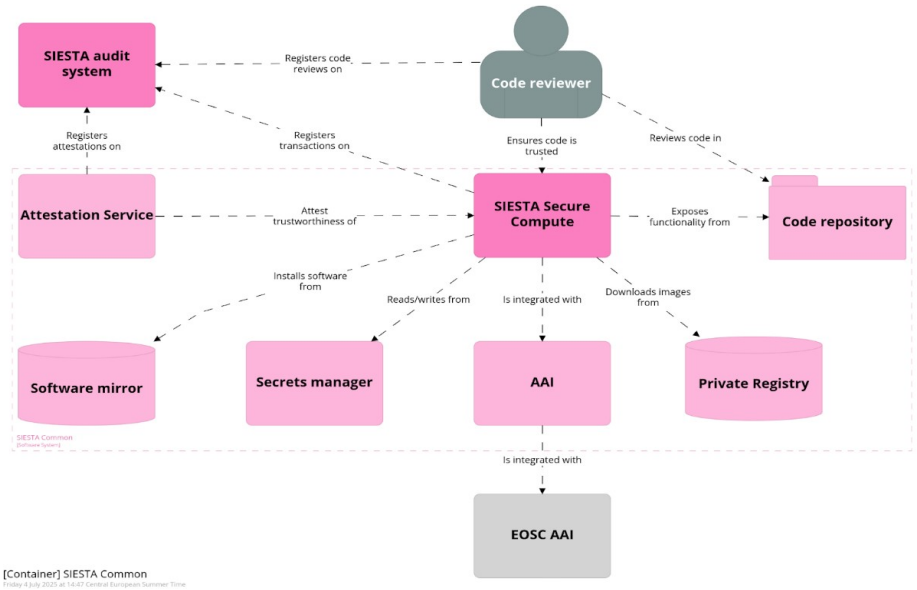

Figure 2. SIESTA common services.

3.3 Secure Compute Components

Figure 3 presents the updated secure compute architecture, showing the interaction of the two main operational personas in the platform. To perform confidential data analysis, there are two possible compute paths operating within the AMD SEV-SNP hardware memory-encryption boundary. The Confidential VM path employs OpenStack Nova for provisioning, QEMU/KVM as the hypervisor, and Cendio ThinLinc for interactive remote desktop access. The Confidential Containers path employs Kubernetes for orchestration, Kata Containers as the micro-VM runtime, the CoCo Operator for runtime-class management, and the CoCo Trustee/Key Broker Service for attestation-based key release. The main difference for the user is the interface they will find after their service is launched (VM vs Container) but both paths share the common supporting services layer described before. Given the API definition there is also the option to develop an external workflow system such as a CLI to interact with the system.

PoS(ISGC2026)031

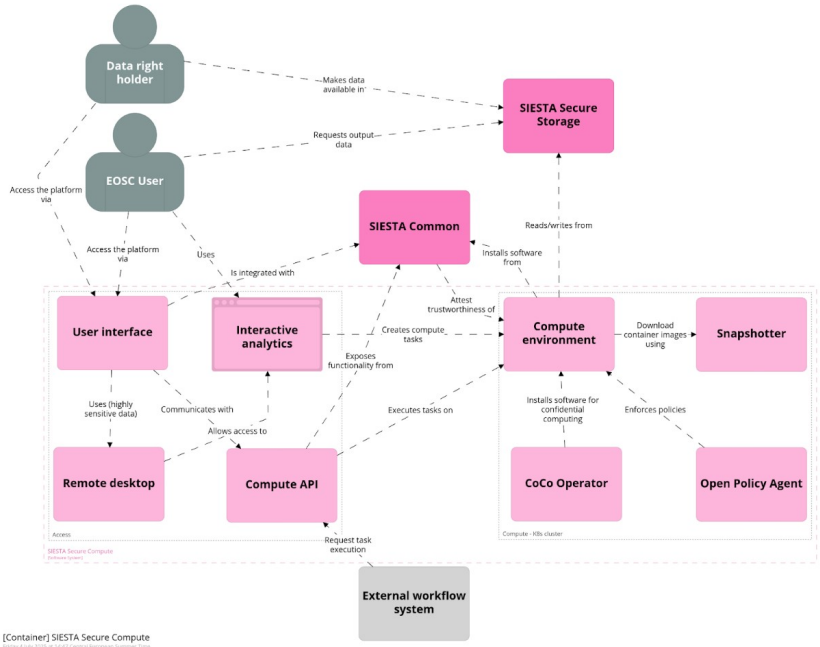

Figure 3. Secure Compute Components.

4. The Confidential-Computing Backbone: TEEs and AMD SEV-SNP

4.1 Trusted Execution Environments

A Trusted Execution Environment (TEE) is a hardware-isolated region of a processor that provides memory encryption, access control, and remote attestation, isolating the code and data of a tenant workload from the rest of the system, including the OS, the hypervisor, and the physical platform operator [5]. All major processor vendors have introduced TEE implementations: AMD SEV-SNP, ARM TrustZone, IBM Secure Service Container, Intel SGX and TDX, and RISC-V MultiZone. These technologies differ in granularity, attestation protocol, and threat model, but share the property that their security guarantees are rooted in dedicated hardware rather than software.

4.2 AMD SEV-SNP: Implementation and Security Properties

The three SIESTA provider sites are equipped with AMD EPYC processors and deploy AMD SEV-SNP [6]. Memory encryption is performed transparently by a dedicated AES-128 engine within the SoC memory controllers, interposed on all DRAM transactions so that DRAM always stores ciphertext while CPU registers see plaintext. Key management is handled by a dedicated AMD Secure Processor (ARM Cortex-A5 co-processor on the SoC die): each VM instance is assigned a unique encryption key generated at instantiation time following NIST SP 800-90 [28], and the key material never leaves the hardware boundary. The SNP extension adds memory-integrity protection through Reverse Map Tables (RMPTables), which validate the physical-to-GPA mapping on each memory access, preventing a compromised hypervisor from remapping pages between VMs or between the VM and hypervisor address space. Together, SEV and SNP address both the confidentiality and integrity dimensions of data-in-use protection.

PoS(ISGC2026)031

4.3 Confidential Containers and Kata Containers

Alongside confidential VMs, SIESTA implements Confidential Containers (CoCo) [29], an open-source CNCF project enabling cloud-native confidential computing via TEEs. CoCo employs Kata Containers as its runtime: each containerised workload executes inside a hardware-encrypted micro-VM, providing TEE-grade isolation to Kubernetes-orchestrated workloads without exposing the host kernel. The CoCo Trustee (Key Broker Service) provides remote-attestation-based key release, cryptographically verifying TEE identity and integrity before releasing decryption keys. Because nested virtualisation is not currently supported by AMD SEV-SNP, CoCo deployments require bare-metal nodes; CSIC/IFCA has deployed both a hybrid cluster (bare-metal SEV nodes plus OpenStack VM nodes) and a fully bare-metal cluster for this purpose.

5. Auxiliary Platform Components

Hardware-enforced memory isolation is necessary but insufficient for a production-ready secure research platform. Researchers and data rights holders require additional assurances: demonstrable tamper-evidence of the audit trail, verifiable provenance of the software supply chain, end-to-end encryption of stored data without operator key custody, fine-grained and auditable access control, scalable orchestration, and real-time intrusion detection. SIESTA addresses each of these requirements through purpose-selected, integrated components, as illustrated in Figures 2 and 3.

5.1 Tamper-Evident Audit Logging: Trillian Tessera

SIESTA integrates Wazuh XDR & SIEM [30] with Trillian Tessera[31] to protect the integrity of security-relevant events. Wazuh collects and analyzes security-relevant activity, while Tessera records the resulting events in a cryptographically verifiable, append-only log based on Merkle trees. Thus, the Merkle-tree construction guarantees that any post-hoc modification of the log is computationally detectable, providing data rights holders and external auditors with mathematically sound evidence of platform integrity. This directly addresses the auditability and traceability constraints identified in D6.1 [4].

5.2 Secure Container Registry and Software Supply Chain: Harbor

SIESTA deploys Harbor [32], a CNCF-graduated, self-hosted container image registry, as the authoritative source for platform container images. Harbor strengthens the platform's security posture by securing the software supply chain through: (i) automated Trivy [33] vulnerability scanning on every image push, enforcing a configurable CVE severity threshold; (ii) Cosign image signing for cryptographically verified provenance; and (iii) SBOM generation satisfying requirements of the EU Cyber Resilience Act and DORA.

5.3 Secure Storage: LUKS Encryption and Secrets Management

The SIESTA zero-trust storage model is implemented through a layered architecture [4, 34]. OpenStack Cinder provides virtual block volumes to confidential VMs. LUKS (Linux Unified Key Setup) with Argon2id-based PBKDF applies encryption exclusively from inside of

PoS(ISGC2026)031

the TEE, ensuring the cloud operator never sees plaintext or keys. HashiCorp Vault manages secrets; keys and tokens reside exclusively in an unswappable tmpfs backed by the TEE's encrypted memory, provisioned by the Data Rights Holder post-boot after TEE attestation completes.

5.4 Identity and Access Management: Keycloak AAI

Authentication and authorisation are centralised in a self-hosted Keycloak instance at IFCA, implementing OIDC and SAML for SSO and enforcing RBAC across all platform services. As documented in D8.1 [35], the decision to deploy self-hosted Keycloak rather than relying solely on EGI Check-in at the platform level, was motivated by limitations in EGI Check-in's social-identity-provider certification, concurrent deployment disruptions, and strategic alignment with MyAccessID, the European Commission's chosen EOSC EU Node authentication framework.

5.5 Real-Time Network-Isolated Workloads

SIESTA implements real-time workload sandboxing to ensure that use case specific analytics workloads access only the sensitive data explicitly permitted by evaluated platform policies. The implemented solution combines S3-compatible object storage as the controlled data backend with Cilium Network Policies enforcing network isolation around containerized analytics workloads. As a result, only authorized deployments can reach the relevant S3 endpoints and datasets for secure analysis, limiting lateral access, reducing unintended data exposure, and providing a cloud-native enforcement layer for policy-driven data access control.

5.6 Orchestration, Monitoring, and Policy Enforcement

Scalability and replicability are achieved through Kubernetes independently instantiated at each provider site. The stack incorporates: the CoCo Operator for automated confidential container runtime management; the Rook Operator for distributed storage; Wazuh for real-time host intrusion detection, log analysis, and file-integrity monitoring; and Open Policy Agent for declarative policy-as-code enforcement across all Kubernetes API operations. The Onyxia portal exposes a service catalogue through which data users may launch JupyterLab instances, Ubuntu workspaces, and domain-specific tools (e.g. anonymization ones such as ANJANA) without direct interaction with the Kubernetes API.

6. DevSecOps Methodology

Confidential hardware and a comprehensive service portfolio are only as trustworthy as the processes through which they are built, deployed, and operated. SIESTA adopts a DevSecOps methodology that embeds security at every phase of the software development lifecycle.

The Kubernetes cluster is provisioned using Kubespray (infrastructure-as-code) and configured using Ansible (configuration management), expressing the complete platform state as version-controlled, reproducible artefacts. All cluster components are deployed and reconciled through a GitOps workflow driven by ArgoCD: the desired cluster state is declared in Git, and ArgoCD continuously reconciles the live cluster to match it, providing an auditable, automatically enforced configuration.

PoS(ISGC2026)031

A self-hosted GitLab instance at IFCA provides the code repository and CI/CD backbone. The CI/CD backbone is implemented as a dedicated component project, with reusable pipeline definitions and release-management workflows managed and distributed through GitLab's native CI/CD Catalog. Every commit to a production branch triggers an automated pipeline: static analysis, unit testing, container image build, Cosign signing, and push to Harbor, where Trivy immediately scans for vulnerabilities. Images failing the vulnerability policy are blocked from deployment. Secure storage provisioning is similarly automated through Ansible playbooks, ensuring LUKS encryption is applied consistently and secrets remain in encrypted Ansible Vault files under the exclusive custody of the Data Rights Holder.

Security hardening (WP12) employs Cilium-based eBPF network policies for fine-grained workload isolation, Open Policy Agent for policy-as-code enforcement, and periodic external security audits. The combined effect is a fully automated, policy-enforced, end-to-end auditable path from developer commit to running confidential workload: researchers interact exclusively through the Onyxia portal and focus entirely on scientific analysis.

7. Use cases

The SIESTA platform is designed to support heterogeneous workflows involving sensitive data, while providing a common confidential-computing environment for analyzing sensitive data. The five initial use cases of the project cover distinct application areas and therefore exhibit different data characteristics, confidentiality requirements, workflows, and levels of sensitivity. Their integration provides a representative validation environment for assessing the applicability of the platform architecture and its confidential-computing capabilities to real-world sensitive-data analytics.

7.1 Presentation of the Five Use Cases

The project supports five distinct use cases, spanning epidemiology, medical imaging, energy, text anonymisation on sensitive data, and demography. Although their analytical objectives differ, they share the requirement to process data in an environment where confidentiality, controlled access, and protection of sensitive information are fundamental. The use cases consequently provide complementary scenarios for evaluating the platform's ability to support secure data ingestion, processing, analysis, and delivery of results.

- **Epidemiology**: Supports the collection and integration of population, mobility, and surveillance data for epidemiological modelling. The resulting datasets are used to simulate infectious-disease propagation under different scenarios. These simulations can be performed using the dedicated service integrated in the SIESTA platform for that purpose.
- **Medical imaging**: Enables the secure staging and analysis of sensitive medical and neuroimaging data together with associated participant metadata. Existing imaging-analysis pipelines are executed while supporting anonymisation and assessment of re-identification risks. Dedicated tools have been developed with the objective of: (1) generating scrambled datasets that are organized according to the Brain Imaging Data Structure (BIDScramble [37]) (2) detecting privacy leakage in neuroimaging and tabular datasets (DatLeak [36]).

PoS(ISGC2026)031

- **Energy**: Provides forecasting and optimisation services for energy generation and consumption in solar-power applications. The use case combines time-series data with external meteorological information while preserving the confidentiality of sensitive energy data. In addition to the platform capabilities for secure computing and data processing, this use case has exploited the anonymization and differential privacy-related tools developed within SIESTA (e.g. concerning metric-privacy and geo-indistinguishability guarantees). In addition, a dedicated service for this use case has been integrated in the platform
- **Text anonymisation on sensitive data**: Addresses the analysis of sensitive textual information for machine-learning applications where real datasets cannot be openly shared. Anonymization, pseudonymisation, and synthetic-data generation are used to preserve privacy while retaining analytical utility. The use case provides a dedicated service in the platform with different methods for text masking.
- **Demography**: Supports secure and reproducible analysis of sensitive socio-demographic and health microdata. Anonymization and synthetic-population techniques facilitate broader data use while reducing risks associated with direct access to confidential datasets.

The diversity of these use cases is a key feature of the platform's evaluation. Workloads range from numerical modelling and time-series forecasting to medical-image processing, natural-language processing, and statistical analysis. Consequently, the platform must accommodate different computational patterns and data-management requirements while maintaining a common security and confidentiality model.

7.2 Pilot Use Case: Energy

The **Energy** use case constitutes a pilot implementation for demonstrating the end-to-end execution of a sensitive-data analytical workflow on the platform. Its objective is to provide forecasting of energy generation and consumption based on historical energy time series and meteorological variables (Figure 4). The workflow illustrates how sensitive input data, external data, analytical models, and computational services can be combined within a controlled cloud environment.

The workflow begins with the data rights holders uploading their energy data, including both their household solar panel generation and overall energy consumption, covering an initial period of 8 days. They then select the household location and configure the system parameters, including the number of solar panels, their inclination, and the system's peak power. The system takes over, first preprocessing the data to ensure consistency, then fetching meteorological variables from a storage service. These variables are merged with the user's time series data, and a pre-trained model is loaded to generate forecasts for energy generation and consumption. The system returns the forecasted energy generation/consumption timeseries. Finally, the system provides the user with the forecasted time series for energy consumption and generation, enabling them to make informed decisions about energy management. All uploaded data is fully anonymized and stored in a database for future reference by researchers. Overall, this workflow ensures seamless integration of data, models, and external variables to deliver actionable insights while maintaining data privacy and security.

PoS(ISGC2026)031

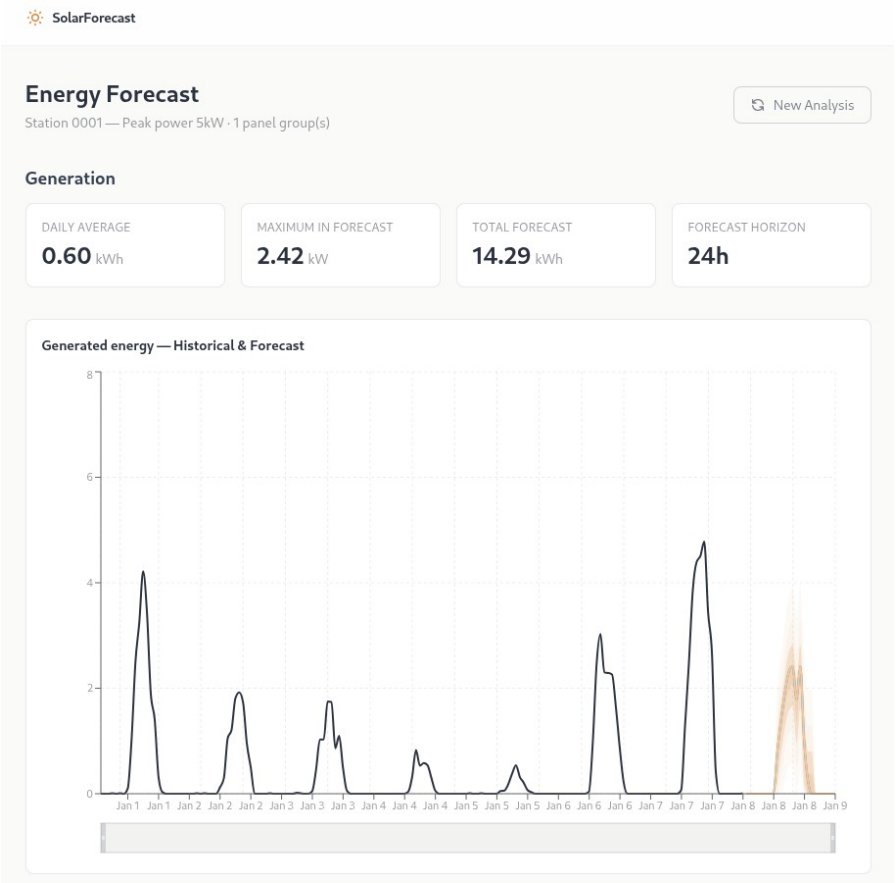

Figure 4: Example of a forecast for energy generation. The black line represents the data uploaded by the data rights holder, and the orange line shows the forecast for the next 24 hours, along with its confidence intervals (shaded area)

The use case constitutes a pilot demonstrating the platform's role-based security framework across distinct user profiles. The use case has two different services: a public client web ui which allows all dashboard users to upload their energy generation/consumption timeseries and get a forecast, once uploaded through a form, and a service only accessible for researchers which is shown conditionally if the user has the required role linked to their SIESTA's account.

This pilot illustrates the integration of several platform capabilities within a single workflow (Figure 5). In particular, it requires controlled handling of user-provided data, access to platform storage services, execution of computational workloads, integration of external variables, and controlled delivery of analytical results. The workflow therefore provides a concrete scenario for assessing how confidential-computing mechanisms can protect sensitive data throughout cloud-based processing while still allowing users to obtain useful analytical results.

Importantly, the objective is not merely to execute a forecasting model in the cloud, but to establish an integrated processing chain in which data, models, and computational services are orchestrated within a privacy- and security-aware environment. The Energy pilot consequently serves as a representative demonstration of the platform's ability to transform sensitive input

PoS(ISGC2026)031

data into actionable analytical outputs without requiring the underlying data to be exposed beyond the authorised processing environment.

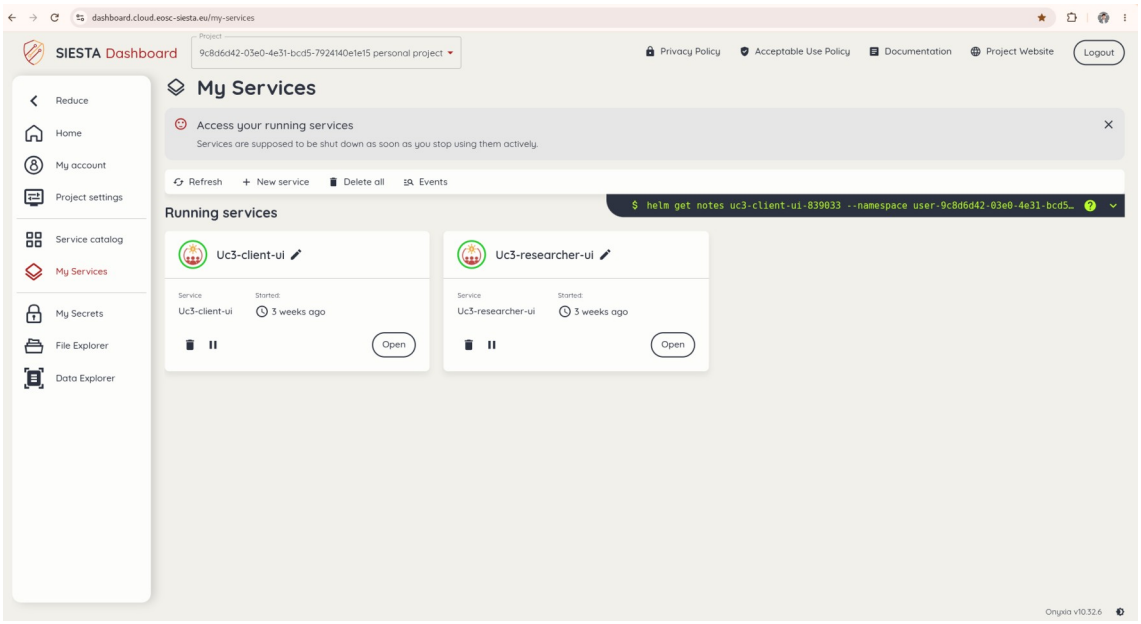

Figure 5: Dashboard with My Services for the Energy use case.

8. Summary

This paper has presented the EOSC-SIESTA platform for confidential computing in sensitive data analytics, contextualising it within the state of the art and describing its updated architecture, implementation, and operational methodology. The principal contributions are as follows.

- A hardware-rooted confidential-compute foundation based on AMD SEV-SNP and Confidential Containers (Kata) provides end-to-end confidentiality for both VM and container workloads, protecting data in use against the full class of privileged-access threats reviewed in Section 2 and not addressed by HE, SMPC, DP, FL, or TREs alone.
- The updated D8.1 platform architecture — comprising Secure Compute, Common Services, and Data tiers across three provider sites — integrates a FAIR-compliant dataset catalogue, service catalogue, issue tracking, legal management module, and formal non-functional requirements for post-quantum cryptographic resilience and environmental sustainability.
- A layered set of platform services — Wazuh and Trillian Tessera audit logging, Harbor/Trivy/Cosign container registry, LUKS/Vault secure storage, Keycloak AAI, and Onyxia portal — addresses the security requirements that TEEs alone cannot satisfy.
- A complete DevSecOps methodology — Terraform/Ansible IaC, FluxCD GitOps, GitLab CI/CD with automated image signing and vulnerability scanning, and OPA policy-as-code — enforces security continuously throughout the software lifecycle.

The SIESTA platform demonstrates that FAIR principles and rigorous data-protection obligations are mutually compatible: researchers focus on scientific analytics while the layered

PoS(ISGC2026)031

security architecture ensures sensitive data remain protected across all three operational states. Platform code, documentation, and deliverables are publicly accessible at https://eosc-siesta.eu / and https://gitlab.ifca.es/eosc-siesta.

Acknowledgements

This work is funded by European Union through the SIESTA project (Horizon Europe) under Grant number 101131957.

References

[1] European Parliament and Council of the EU, Regulation (EU) 2016/679: General Data Protection Regulation, Official Journal of the EU, 2016.

[2] European Parliament and Council of the EU, Regulation (EU) 2022/868: Data Governance Act, Official Journal of the EU, 2022.

[3] European Parliament and Council of the EU, Regulation (EU) 2024/1689: Artificial Intelligence Act, Official Journal of the EU, 2024.

[4] Á. López García et al., D6.1 — Requirements and Architecture Definition, SIESTA Deliverable, DOI: 10.5281/zenodo.11485214, 2024.

[5] Confidential Computing Consortium, A Technical Analysis of Confidential Computing, Linux Foundation, v1.3, 2022.

[6] AMD, AMD SEV-SNP: Strengthening VM Isolation with Integrity Protection and More, AMD White Paper, 2020.

[7] P. Karger, R. Schell, Thirty Years Later: Lessons from the Multics Security Evaluation, Proc. ACSAC, 2002.

[8] T. Garfinkel, M. Rosenblum, A Virtual Machine Introspection Based Architecture for Intrusion Detection, Proc. NDSS, 2003.

[9] T. Stewin, I. Bystrov, Understanding DMA Malware, Proc. DIMVA, 2013.

[10] P. Kocher et al., Spectre Attacks: Exploiting Speculative Execution, Proc. IEEE S&P, 2019.

[11] M. Lipp et al., Meltdown: Reading Kernel Memory from User Space, Proc. USENIX Security, 2018.

[12] J. Halderman et al., Lest We Remember: Cold Boot Attacks on Encryption Keys, Proc. USENIX Security, 2008.

[13] C. Gentry, Fully Homomorphic Encryption Using Ideal Lattices, Proc. STOC, 2009.

[14] J. Cheon, A. Kim, M. Kim, Y. Song, Homomorphic Encryption for Arithmetic of Approximate Numbers (CKKS), Proc. ASIACRYPT, 2017.

[15] A. Yao, Protocols for Secure Computations, Proc. IEEE FOCS, 1982.

[16] Y. Huang et al., Faster Secure Two-Party Computation Using Garbled Circuits, Proc. USENIX Security, 2011.

PoS(ISGC2026)031

[17] C. Dwork, A. Roth, The Algorithmic Foundations of Differential Privacy, Foundations and Trends in Theoretical Computer Science, 2014.

[18] H. McMahan et al., Communication-Efficient Learning of Deep Networks from Decentralized Data, Proc. AISTATS, 2017.

[19] D. Rieke et al., The Future of Digital Health with Federated Learning, npj Digital Medicine, vol. 4, no. 119, 2021.

[20] L. Zhu, Z. Liu, S. Han, Deep Leakage from Gradients, Proc. NeurIPS, 2019.

[21] UK Health Data Research Alliance, Building Trusted Research Environments — Principles and Best Practices, UKRI, 2021.

[22] F. Ritchie, The Five Safes: Data Privacy at UK Data Service, UK Data Service Working Paper, 2017.

[23] L. Sweeney, k-Anonymity: A Model for Protecting Privacy, Int. J. Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 10, no. 5, 2002.

[24] A. Machanavajjhala et al., l-Diversity: Privacy Beyond k-Anonymity, ACM TKDD, vol. 1, no. 1, 2007.

[25] N. Li, T. Li, S. Venkatasubramanian, t-Closeness: Privacy Beyond k-Anonymity and l-Diversity, Proc. ICDE, 2007.

[26] J. Sáinz-Pardo Díaz, Á. López García, An open source python library for anonymizing sensitive data. Scientific data, vol 11, 1289, 2024.

[27] J. Van Bulck et al., Foreshadow: Extracting the Keys to the Intel SGX Kingdom, Proc. USENIX Security, 2018.

[28] NIST, SP 800-90A Rev. 1: Recommendation for Random Number Generation Using Deterministic Random Bit Generators, NIST, 2015.

[29] Confidential Containers Community, Confidential Containers Project, https://confidentialcontainers.org/, accessed April 2026.

[30] Wazuh Inc.: Wazuh Open source XDR and SIEM, https://wazuh.com/platform/overview/, accessed April 2026.

[31] Transparency.dev, Trillian Tessera, https://blog.transparency.dev/introducing-trillian-tessera, accessed April 2026.

[32] CNCF, Harbor: Cloud Native Registry, https://goharbor.io/, accessed April 2026.

[33] Aqua Security, Trivy: Vulnerability Scanner for Containers and Other Artefacts, https://trivy.dev/, accessed April 2026.

[34] V. Legoll et al., D7.1 — 1st SIESTA Platform Release, SIESTA Deliverable, DOI: 10.5281/zenodo.15533052, 2025.

[35] Á. López García et al., D8.1 — Requirements and Architecture Definition (Update), SIESTA Deliverable, DOI: 10.5281/zenodo.15533063, 2025.

[36] E. Kibsgaard, A. S. Jwa, C. J. Markiewicz, D. Rodriguez Gonzalez, J. Sainz Pardo, R. A. Poldrack, C. R. Pernet; Assessing metadata privacy in neuroimaging. Imaging Neuroscience 2026; doi: https://doi.org/10.1162/IMAG.a.11442

[37] BIDScramble. https://github.com/Donders-Institute/bidscramble, accessed September 2026.

PoS(ISGC2026)031