IOWN GLOBAL FORUM ™

Functional Architecture for Protection of Data in Use: IOWN Privacy Enhancing Technologies

Classification: APPROVED REFERENCE DOCUMENT

Confidentiality: PUBLIC

Version 1

October 2024

[PETs Functional Architecture]

Legal

Contents

List of Figures

List of Tables

1. Introduction

1.1. Purpose

Open data distribution schemes are gaining momentum as a global trend, as seen in International Data Space Association (IDSA) and Catena-X [IDSA] [Catena-X]. However, although today's data distribution schemes define conditions for data handling, such as policies for data access control, they do not provide technical guarantees for the protection of the data itself.

Data can generally be classified into three states (data in motion, data at rest and data in use). Most regulations on data distribution between entities only specify the protection of data in motion and data at rest, and do not provide technical guarantees on how data is protected once passed from one entity to another.

As a result, the following data security threats exist on current data distribution platforms.

- Data use that is not compliant with agreed-upon usage policies between the data provider and the user: Users with data access rights or privileged users of the shared infrastructure use data beyond the agreed scope in violation of policies.
- Malicious data attacks within the shared infrastructure: Information is compromised when external attackers break into virtual machines or underlying hypervisors (so-called Dom0 VMs) or when insider attackers exist in the shared infrastructure and have access to protected data.

As described above, from the data owner's perspective, data distribution needs to be based on mutual trust between the data owner and the data user that contracts are honored. This means that data distribution is currently not at a stage where the data owner can provide their data with any degree of confidence. As a promising solution for the protection of data in use, privacy-enhancing technologies (PETs), a generic term for technologies that enhance privacy protection, including confidential computing, have been attracting significant attention in recent years as a potential solution to these issues. By terminating secure network connections in a secure space where transmitted data and its usage are protected and governed through PETs, it is possible to protect the entire data life cycle even if cloud hosting infrastructure is used. Therefore, the purpose of this document is to define an IOWN Privacy-Enhancing Technologies (IOWN PETs) architecture that can maintain data sovereignty throughout data life cycle in order to promote active data distribution on the IOWN infrastructure.

In the IOWN era of disaggregated computing, it is assumed that data owners will combine distributed resources on demand to handle live data. In such an era, the conventional access model, in which data is collected and then authorization is set in a centralized manner, cannot fully protect data. Therefore, it is important to define policies that govern how data is handled when it is generated and to have a mechanism to enforce these policies when the data is distributed.
In addition, to prevent data siloing and realize active, secure data distribution across organizations and countries, it is necessary to construct the above data protection mechanisms using open interfaces that do not depend on specific vendors or operators. By defining the minimum necessary open interfaces, this document aims to achieve system-wide policy enforcement with minimum additional burden on the providers and the users of the open data distribution infrastructure that is already being discussed globally.

1.2. Objective

The objective of this document is to describe the IOWN PETs architecture which is capable of consistently guaranteeing the technical protection of data according to the data protection policy of the data owner throughout the state of distribution of the data (data in motion, data at rest, and data in use), from its creation to its extinction.

In the IOWN PETs architecture, a "PETs Space" is defined as a virtual space where data confidentiality and governance are technically guaranteed throughout the data life cycle by encryption technologies, etc., regardless of its state. Regarding this PETs Space, the following objectives are achieved in this document.

- Clarification of the PETs Space requirements to encourage open data distribution.
- Definition of an open architecture for IOWN PETs to enable the PETs Space.

1.3. Scope

This document provides definition of requirements toward IOWN PETs architecture with use case analysis and semantics-level definitions of the IOWN PETs architecture that enables the PETs Space regarding the following, and the Syntax-level definitions are the scope of subsequent documents.

- Components that consist of the IOWN PETs architecture and their object models
- Intercomponent interfaces of the IOWN PETs architecture

This document also describes examples of the use and benefit of PETs Space.

2. Use Cases and Issues

2.1. Use cases

This section describes specific use cases that require secure data distribution, and the data security issues in those use cases.

2.1.1. Use case 1: Providing high-value industrial analysis algorithms to other companies

This is a use case in which a company provides another company with a proprietary, high-value industrial analysis algorithm that implements technical knowledge or intellectual property that is a source of competitive differentiation. Potential examples include the prediction and analysis of machining times of machine tools or reaction times of chemical plants. As shown in the figure, Company A provides its own data for analysis, and Company B provides its proprietary algorithm on a shared infrastructure for the analysis of Company A's data. Company A executes the data analysis process using Company B's algorithm and obtains only the results of data analysis from the shared infrastructure without either company viewing the other's intellectual property.

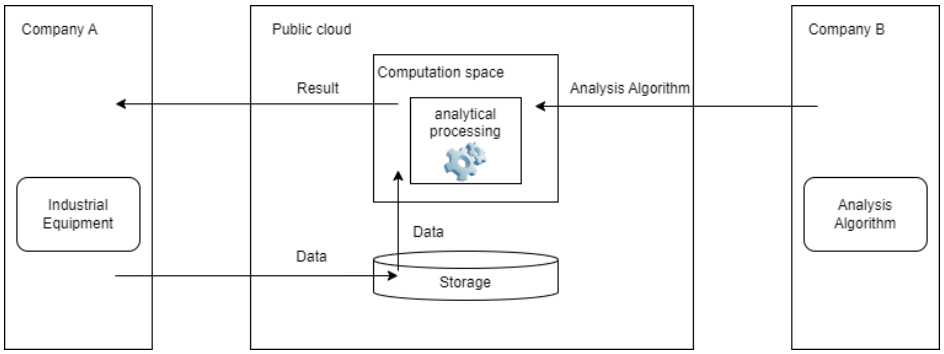

Figure 2.1-1: Providing high-value industrial analysis algorithms to other companies

2.1.2. Use case 2: Secure healthcare data analysis that combines personal vital data and medical record data

This is a use case in which Company A, a healthcare service provider, provides its customers' personal data to a specific hospital with the customers' consent while keeping the raw data confidential and limiting its usage and disclosure. The Hospital combines personal data with medical record data to obtain analysis results for research and development. Company A provides its own customers' vital data to a specific hospital with permission to use this data for a specific purpose. The hospital combines the vital data with its own medical record data and uses the public cloud, which has sufficient computing resources to perform large-scale calculations, to efficiently analyze the data and obtain analysis results.

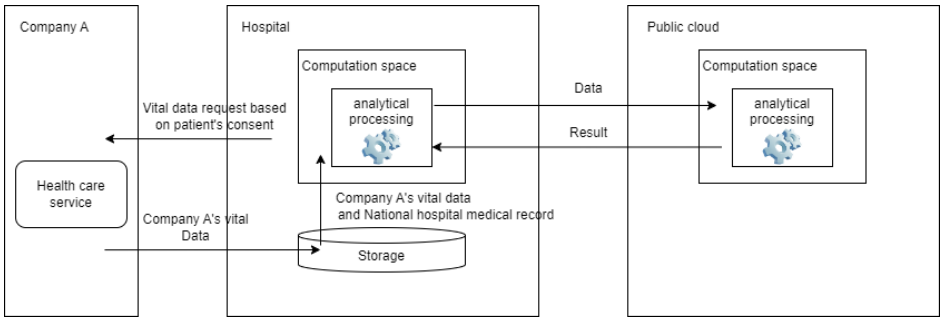

Figure 2.1-2: Analyzing that combines personal vital data and medical record data

2.2. Potential issues with today's practice

To achieve the use cases described above, the following issues must be addressed regarding data security.

- Data use not in accordance with policies: Although providers of data and algorithms such as Company A and Company B in use case 1 expect that the data they provide will be used in accordance with the usage policy agreed between Company A and Company B, current platforms allow operators of public clouds and users with access rights to use data in violation of the policy, and there is no technical guarantee that the data and algorithms they provide will be used in accordance with the policy. For example, if Company B in use case 1 has rights to access computation space, they can access to data of Company A on the cloud and transmit this data to locations that Company B has not agreed for it to be sent. Also, the Hospital in use case 2 can use the data of Company A for purposes other than those agreed upon.
- Malicious data attacks, especially in the shared infrastructure: In many of today's environments, storage, and communication channels are encrypted, but it is not common for the computation space to be encrypted. This means there is a risk of information leakage if an external attacker can penetrate the computation space regardless of whether it is on-premise or in a public cloud. Furthermore, system operators often have administrative privileges over the encrypted data even if it is encrypted and data on the platform is at risk of internal attacks by operators. For example, an internal attacker within a public cloud who has administrative privileges in use cases 1 and 2 can access to data in computation space on the public cloud, making this data vulnerable to theft or tampering. Therefore, it cannot be said that the data is protected against platform providers.

In this document, the requirements necessary to solve these issues are described in Section 3, the relevant existing technologies and the technical gaps to meet the requirements are clarified in Section 4, and the functional architecture of IOWN PETs is defined in Section 5. Section 6 describes examples of the use of IOWN PETs from the user's point of view. In addition, Section 7 discusses the user benefits of IOWN PETs.

3. Requirements for the IOWN PETs Architecture

This section describes the necessary features the IOWN PETs architecture requires to address the issues described in section2.

3.1. Data use in accordance with policies

The data on the IOWN PETs architecture should be stored, exchanged, and computed according to policies specified by a data owner. Policies may include, but are not limited to, the following.

- Location: Data shall be located in the allowed country/company/organization as the policy specifies.
- Devices: Data shall be sent/stored/computed in an appropriate device, e.g. firmware and hardware of the device are correctly produced by the chip vendor.
- Data access: Only authorized access as defined by the policies is allowed.
- Computed function: Computation shall be performed to the extent agreed upon with the data owner.
- Controllability of disclosed data: Disclosed data is limited to processed information if the data owner so desires (e.g., anonymously processed information, statistical information, etc.).

Users of the IOWN PETs architecture can obtain technical guarantees that the data they provide will only be used in accordance with the agreed-upon policies.

3.2. Confidentiality throughout the data lifecycle

The IOWN PETs architecture should support the computation of data as well as the storage and exchange of data. Therefore, a mechanism is needed that, at the very least, keeps the following pieces of information confidential, though there may be other functions such as integrity:

- Data confidentiality: Computation of data is done while keeping the data secret. In some cases, the output of the computation shall also be kept secret.
- Algorithm confidentiality: The computation of data is done while the algorithm for the computation is kept secret.

3.3. Non-functional requirements

The IOWN PETs also need to fulfil the following non-functional requirements.

Scalability

The IOWN PETs architecture must support the scale-up/scale-out capability required to implement distributed computing.

Processing Performance

The processing performance of the system that is achieved by IOWN PETs architecture is an important KPI for processing a large volume of data. Processing volume per unit of time and processing speed of data need to be considered.

Minimum Overhead

Due to the nature of the IOWN PETs, overheads are incurred to encrypt/decrypt data and enforce access rights policies. These overheads need to be minimized to provide an efficient environment for applications.

4. Existing Gaps Within Current Data Security

This section describes the existing state-of-the-art technologies related to IOWN PETs and the security gaps. This section also provides instructions on how these gaps can be filled.

4.1. State of the art of data security

4.1.1. Technologies for protection of data in motion

Technologies for protecting data in motion in the quantum computer era are described in Technology Outlook for Information Security [IOWNsec]. Post-quantum cryptographic communication technologies are maturing, and these technologies should be utilized in the IOWN era.

4.1.2. Technologies for protection of data at rest

Access control

Access control technologies are mainly used to protect data in the storage phase. Attribute-based access control, which controls access to data based on the attributes of access-controlled users and resources, and role-based access control, which controls access based on the roles and duties of users, are the mainstream solutions for this purpose. Attribute-based encryption, which enables fine-grained access control of encrypted data using different encryption keys depending on the attributes, can also be used. However, these technologies alone cannot control data use after granting access.

Storage encryption

Encryption techniques can provide robust security for data at rest. In database systems, there are three primary types, all of which are detailed in this section. Each method requires encryption keys to decrypt the data, adding an extra layer of security. Even if the attacker bypasses the storage access control measures, the correct encryption keys to access the data in the storage would still be needed. This makes encryption a formidable line of defense for securing data at rest.

Transparent Database Encryption (TDE): TDE is used to encrypt an entire database. The contents of the database are encrypted using a symmetric key, often referred to as a “database encryption key.” TDE encrypts all data, meaning no application modifications are necessary for TDE to function correctly.

Column-level Encryption: This method allows for individual columns within a database to be encrypted. This granularity of column-level encryption results in specific strengths and weaknesses compared to encrypting an entire database. Typically, applications need to be designed considering which columns are encrypted.

Application-controlled Encryption: In this type, the applications themselves are responsible for encrypting and decrypting data. Attribute-based encryption can be used in this context. However, encryption prevents several data processing functions, such as sorting and content-aware compression, from being performed within the database.

4.1.3. Technologies for protection of data in use

Information Rights Management (IRM)

IRM is a mechanism for encrypting files, granting operating privileges to each user, and managing their usage. Although IRM can even control how data is used after access is granted, it is a solution that specializes in the management of document files, images, etc. There are limits to the applications that can be used while under IRM management.

Privacy Enhancing Technologies (PETs)

Technologies called privacy enhancing technologies (PETs) are emerging to protect data in use that was traditionally unprotected. “PETs” is a generic term for technologies that enhance privacy protection and includes the following specific technologies detailed in this section. The first three are technologies that provide encrypted data processing, and the latter two are technologies that provide data obfuscation and distributed analytics, respectively.

Secret Sharing-based Multi-Party Computation (SS-MPC): Secret sharing (SS) is a cryptographic technique used to protect the confidentiality of a message by dividing it into pieces called shares. In SS-MPC, a message is shared among participating parties via SS, and the parties compute a function on the shared message while maintaining its confidentiality and obtaining shares of the function output. The output can be obtained using a message reconstruction algorithm of SS, taking all or a subset of the output shares as input.

Homomorphic Encryption (HE): HE is a type of encryption in which addition, multiplication, or a combination of these can be performed while encrypted. One promising application is outsourcing computation, where ciphertexts are handed over to a third party and the computation is performed by that party.

Trusted Execution Environment (TEE): The TEE is a secure area within a processor. It guarantees that the code and data loaded inside it are protected with respect to confidentiality and integrity. Essentially, TEEs provide a kind of ‘safe room’ for sensitive operations, ensuring that even if a system is compromised, the data within the TEE remains secure. TEEs operate by isolating specific computations, data, or both, from the rest of the device or network. This isolation is hardware-based, which makes it highly resistant to external attacks, including those from the operating system itself. Within a TEE, code can run without risk of interference or snooping from other processes [TEE].

Differential Privacy: Differential Privacy is a mathematical framework that enables the publication of statistical information about a data set while protecting the privacy of individual data in functions such as data publication, data analysis, etc. It typically reduces the privacy risk by adding Gaussian or Laplacian noise to the function, selected to make it unable to infer any individuals in the dataset.

Federated Learning: Federated learning (FL) is a distributed ML approach that trains ML models on distributed datasets. The goal of FL is to improve the accuracy of ML models by using more data while preserving the privacy and the locality of distributed datasets. FL increases the amount of data available for training ML models, especially data associated with rare and new events, resulting in a more general ML model [FI].

4.2. Gaps

Regarding data confidentiality, technologies to protect data in motion and data at rest are mature enough, and technologies to protect data in use are becoming increasingly used in actual services. Still, they do not cover data protection when transitioning between each state. In addition, only very limited solutions exist for policy-driven data usage controls, such as storage access control and IRM solutions. IRM is one of the few solutions that can ensure confidentiality and control usage throughout the data lifecycle by embedding encryption/decryption functions in the data usage control system. However, it has limitations on applicable data formats and is also not applicable to open data collaboration because it is a centralized mechanism.

In conclusion, although there are many existing technologies for the protection of data in each state, as described above, no mechanism that enables seamless protection of the entire data lifecycle and simultaneous data usage control has been realized for open data collaboration.

4.3. Directions to fill gaps

This document focuses on PETs, which can be a generic solution to protect data in use. It proposes an architecture that seamlessly connects technologies that protect data in each state. This architecture provides a data space that maintains confidentiality by connecting environments protected by PETs with direct secure communication through open interfaces. It simultaneously enables policy-driven data usage management to maintain data sovereignty throughout the entire data lifecycle for open data collaboration.

5. IOWN PETs Basic Functional Architecture

This section describes the basic functional architecture of IOWN PETs from the user's perspective and from the deployer’s perspective.

IOWN PETs within the IOWN Global Forum architecture are defined as a common platform layer that provides confidentiality functions to meet the requirements described in section 3 for applications, just as the IDH (IOWN Data Hub) provides fundamental data access functions for applications. IOWN PETs provide applications with a solution that provides secure virtual spaces in which data confidentiality and governance are technically guaranteed throughout the data life cycle in accordance with policy. The solution is referred to as the PETs Space and its provider is known as a PETs Space provider. IOWN PETs includes the necessary resources provided by DCI which provides applications with a distributed and heterogeneous computing and networking environment that spans end-to-end, i.e., across clouds, edges, and customer premises.

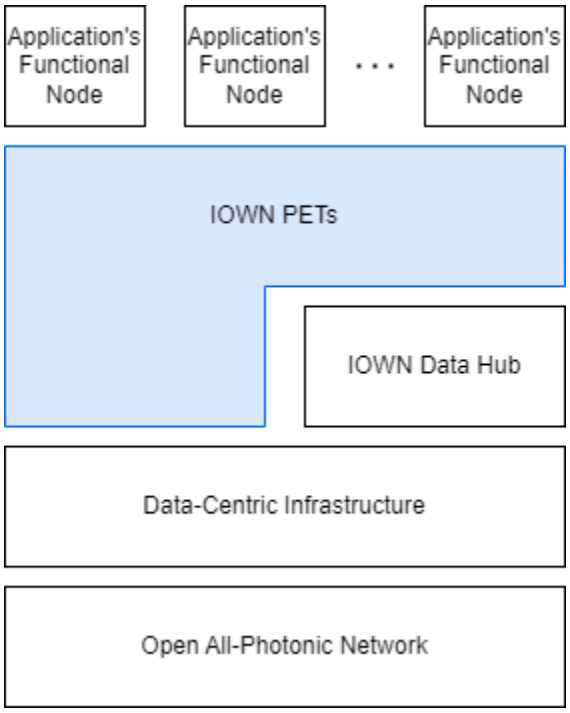

Figure 5-1: IOWN Global Forum architecture incorporating IOWN PETs

Definition of Terms

The table below shows the definition of basic terms used in this section.

Table 5-1: Definition of Terms

TERM	DEFINITION
Data	Arbitrary data to be handled in this architecture.
Application	Programs that process data in this architecture.

Result	Among data, result data calculated using data provided by data owners and applications provided by application owner. Some are eligible for protection, and some are not. For example, statistical data may not need to be protected.
Role	Named set of executable operations that can be performed by the user.

5.1. Functional architecture from view of the user

5.1.1. Object model

The figure below shows an object model for the IOWN PETs from the user's point of view using UML notation.

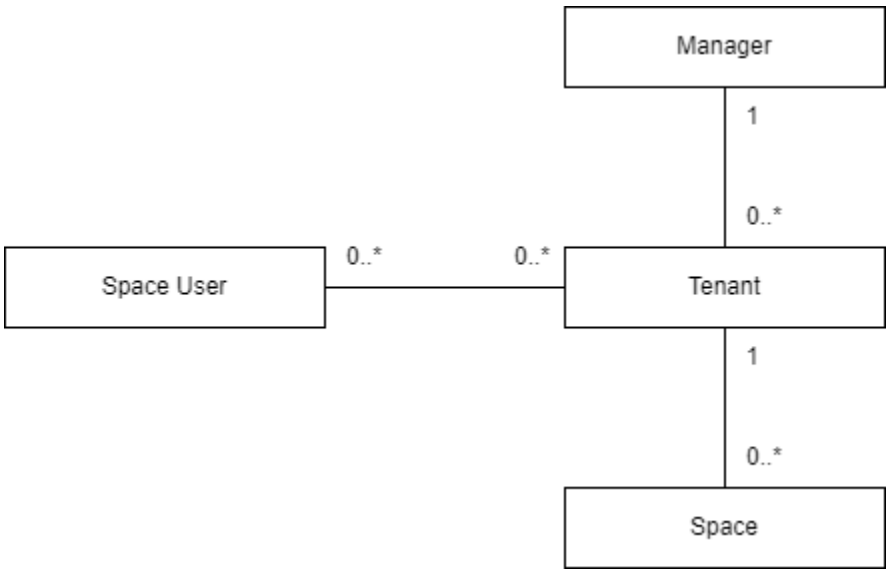

Figure 5.1-1: Object model for the IOWN PETs from user view

The definitions of each object are given below.

PETs Space: A space where data protection is technically guaranteed by PETs Space providers and governance over the data is realized, no matter what state the data is in. PETs Spaces are expected to be available not only within a single data center, but also across multiple data centers, organizations, and countries. PETs Space has the following minimum features.

- Seamless confidentiality across Data at rest, Data in motion and Data in use
- Policy-based data and application usage control
- Space User: A user who inputs/outputs data to/from the PETs Space or operates and monitors the PETs Space. Basic roles of a Space User: Data Provider / Data Consumer / Application Provider / Space Administrator
  - Data Provider: A user who provides data for analysis to the PETs Space.
  - Data Consumer: A user who retrieves and uses the analysis results from the PETs Space.
  - Application Provider: A user who provides applications for data processing and usage such as analysis to the PETs Space.
  - Space Administrator: A user who operates and monitors the PETs Space.
- Tenant: A Tenant is a unit that binds a set of Spaces. A Tenant can be associated with multiple PETs Spaces. A Space User who can use a Tenant can use all PETs Spaces associated with the Tenant.

- Manager: A system that manages Tenants. There are multiple Tenants under the Manager. How to create and associate Tenants and Space Users with the Manager is outside the scope of this document and is not specified.

5.1.2. IOWN PETs I/F from view of the user

The figure below shows IOWN PETs I/F from user's point of view.

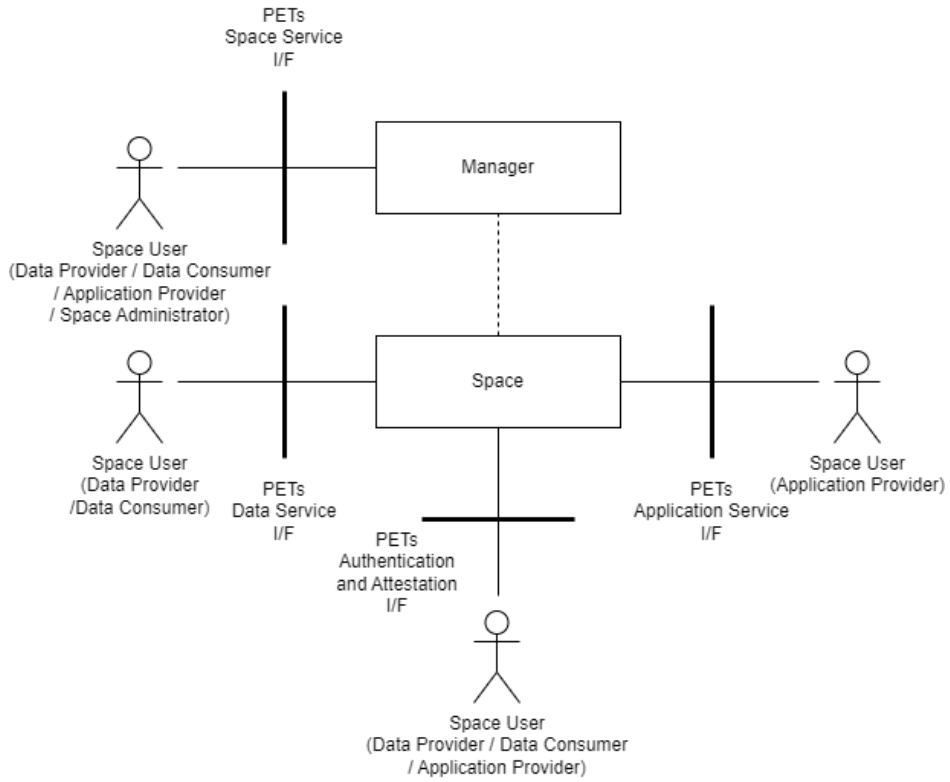

Figure 5.1-2: IOWN PETs I/F from view of the user

PETs Space Service I/F

Space Users can search an existing PETs Space or CRUD a PETs Space in the Tenant by using this I/F. The policy configuration for the use of data and applications provided to the PETs Space is also done through this I/F. This I/F also provides the operation and monitoring functions of PETs Space for Space Administrators.

Note: There are several possible variations in how policies can be handled, such as linking them to data or setting them to a PETs Space, but this document does not specify it in that level of detail. The handling of policies will be left to subsequent documents.

The minimum APIs provided by this I/F are described below at the semantics level.

- Discover Space: APIs to search for Spaces in the PETs world.
- Space Information: APIs to get meta-information on PETs Space (policies, participants, executable applications, etc.), PETs Space state, etc.
- Tenant CRUD: APIs that CRUD the Tenant.
- Space CRUD: APIs that CRUD the PETs Space that combines components, allowing arbitrary spaces to be created under a tenant.

- Policy configuration: APIs to configure usage policies for the Tenant, the PETs Space, and the data and applications handled within Space.
- Operation: APIs for PETs Space administrators to monitor and operate PETs Space.
- Lineage: APIs for PETs Space administrators to provide traceability functions such as logging and history management of data on the PETs Space.

PETs Data Service I/F

Space Users (Data Provider/Data Consumer) can input/output data using this I/F. This I/F is also used to acquire data from data lakes such as IDH.

The minimum APIs provided by this I/F are described below at the semantics level.

- Data exchange: APIs to input/output data to/from the PETs Space. It is also assumed to connect to IDH through this APIs.
- Execution: APIs to securely execute functions provided by applications on the PETs Space.

PETs Application Service I/F

Space Users (Application Provider) can CRUD applications via this I/F.

The minimum APIs provided by this I/F are described below at the semantics level.

- Deploy: APIs to deploy and undeploy applications.
- Operation: APIs to monitor and operate application status.

PETs Authentication and Attestation I/F

The minimum APIs provided by this I/F are described below at the semantics level.

- Authentication: APIs to provide the security process of verifying the identity of a user or application attempting to access the PETs Space.
- Attestation: Attestation: APIs to verify the configuration of the PETs Space. TEE's Remote Attestation, SBOM, etc. can be used to verify the hardware/devices used, OS and boot-time register information, on-board software, and their configurations (kernel modules, network settings, software settings, etc.). Space Users can verify the entire PETs Space without being aware of the individual components that make up the PETs Space.

  Note: The attestation in this document does not only refer to the RoT-based protocols used in TEE and TPM, but also in the broader sense of proof of configuration information.

5.2. Functional architecture from view of the deployer

5.2.1. Object model

The figure below shows an object model for the IOWN PETs from deployer’s point of view written in the UML chart.

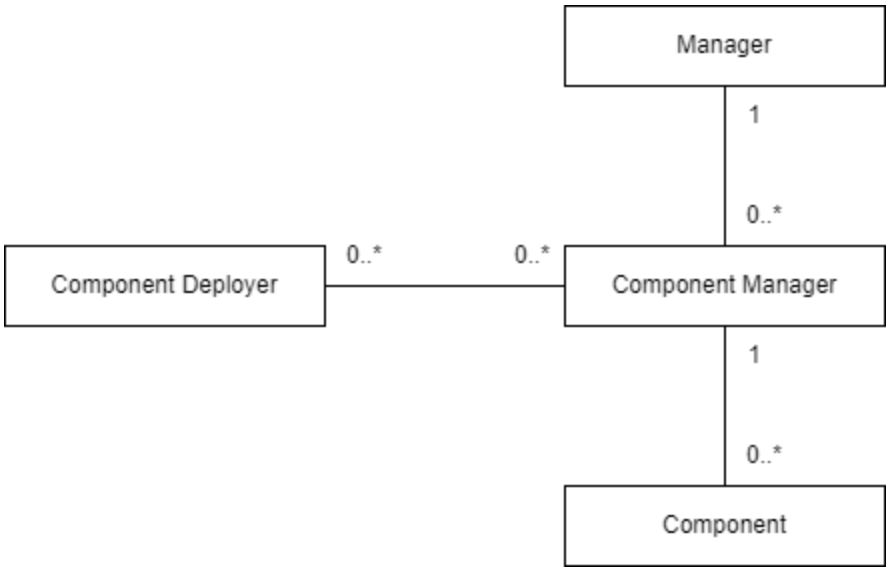

Figure 5.2-1: Object model for the IOWN PETs from view of the deployer

The definitions of each object are given below.

- Component (PETs Component): Component is a means of achieving a PETs Space, and a PETs Space consists of multiple Components.
  Component is a combination of hardware that provides a root of trust (RoT)-based, attestable trusted environment and software on that hardware, or software alone. Extending the definition of Confidential Computing by Confidential Computing Consortium [Confidential Computing Consortium], a PETs Component is defined as an attestable trusted environment that achieves one or more of the following: protection of data in use, protection of data in motion, and protection of data at rest. The Component must have attestability through its own or another component's functionality.

  NOTE: The definition of Confidential Computing by Confidential Computing Consortium [Common Terminology for Confidential Computing]

  “The Confidential Computing Consortium has defined Confidential Computing as “the protection of data in use by performing computation in a hardware-based, attested Trusted Execution Environment”.

  Examples of a component for protection of data in use

  A system that combines hardware (e.g., CPU TEE, GPU TEE, etc.) that provides an isolated environment and execution environment software that provides policy enforcement (usage control, access control, etc.) that runs on the hardware.

  SS-MPC software that provides a cryptographically protected execution environment.

  Example of a component for protection of data in motion

  MACsec/IPsec encrypted communication software that realizes quantum cryptography-resistant communication.

  Example of a component for protection of data at rest

  A system that combines hardware that provides an isolated environment with file system software that provides secure storage on top of the hardware.

- Component Manager (PETs Component Manager): A program that handles the PETs Component. Generally, PETs Component and PETs Component Manager are provided by the vendor as a pair.
- Manager: A program that sends instructions to PETs Component Manager according to instructions from the PETs User, combines PETs Components.
  Note: This Manager is identical to the Manager, which is a User View object, but the Manager has a functionality from the User View and a functionality from the Deployer View.
- Component Deployer: A person who manages the lifecycle of the PETs Component that makes up the PETs Space.

5.2.2. IOWN PETs I/F from view of the deployer

The figure below shows IOWN PETs I/F from deployer's point of view.

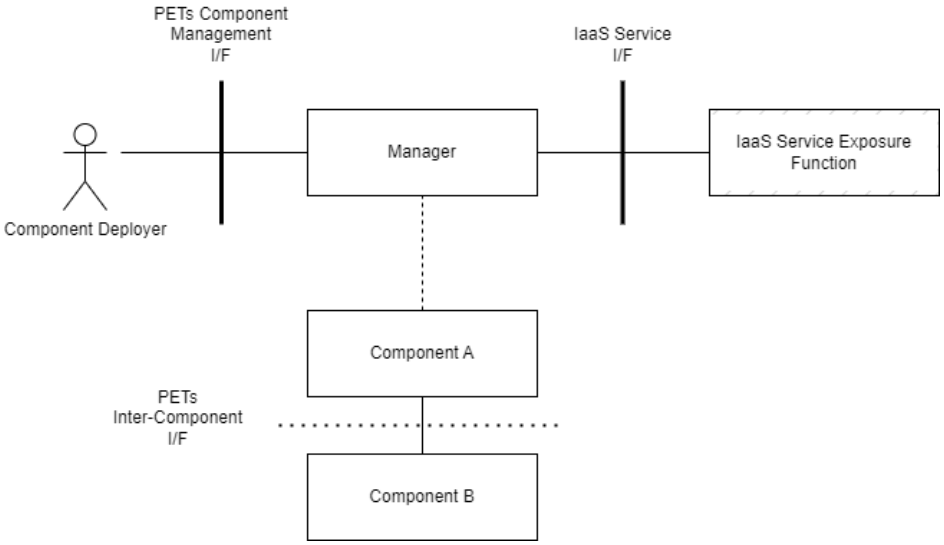

Figure 5.2-2: IOWN PETs I/F from view of the deployer

PETs Component Management I/F

The Deployer manages the lifecycle of the PETs Components that make up the PETs Space through this I/F.

The minimum APIs provided by this I/F are described below at the semantics level. Since this I/F is used only by the deployer of the component, no attestation APIs are required.

- Lifecycle Management: APIs for Component deployers to manage the lifecycle of Component, providing state management functions such as loading and unloading the Component on the system, starting, initializing, terminating, stopping, and getting the state.

IaaS Service I/F

The Manager uses the lower layer Service APIs to realize the CRUD of the Components that make up the PETs Space, as well as the connection between the Components. In particular, it is assumed that the DCIaaS APIs are used when PETs are built on DCI. The IaaS Service Exposure Function is a service function that enables the provisioning of hardware resources.

PETs Inter-component I/F

Components for protection of data in motion constituting a PETs Space communicate with each other securely through this I/F. This I/F can connect multiple different components to form a PETs Space. The PETs Space, which is a set of

multiple components, also seamlessly maintains the confidentiality of the whole space, and has the attestability of all components.

The minimum APIs provided by this I/F are described below at the semantics level.

- Inter-Component: APIs for secure data exchange between PETs Space components.
- Attestation: APIs to verify the components that will trust each other and make up the PETs Space. TEE's Remote Attestation, SBOM, etc. can be used to verify the hardware/devices used, OS and boot-time register information, on-board software, and their configurations (kernel modules, network settings, software settings, etc.).

5.3. Trust model of IOWN PETs

This section describes the trust model of the IOWN PETs architecture. The basic PETs architecture trust model is that the user trusts the PETs Space provider, who provides PETs Space by organizing multiple Components that make up the PETs Space and an attestation of all Components, and the providers of each Component. By trusting the PETs Space provider in addition to the Component providers, the user can obtain guarantee of data confidentiality throughout the entire PETs Space. Specifically, the PETs Space Users verify the confidence of PETs Space through the attestation of PETs Space provided by the PETs Space provider without being aware of the individual heterogeneous hardware, software, and operators that make up the PETs Space. This trust model also allows for open data distribution that is not dependent on specific vendors or operators while maintaining confidentiality.
The trust model of the IOWN PETs architecture can be broken down into the following two parts.

Trust model for PETs Users and external systems

PETs Users trust PETs Space by verifying information provided by the PETs Space (i.e., attestation). The confidence of a specific PETs Space is verified through the authenticity and the transparency of PETs Space as a set of Components. The verification of the authenticity of PETs Space requires that all Space Components are attestable. The functionality provided by a PETs Space to provide an attestation of all Components that make up PETs Space is called Space Attestation in this document. The transparency of PET's Space here means that the configuration and the provider of each component has been made public somewhere. The transparency of PETs Space is achieved by the registration and the disclosure of information (version, provider, etc.) of the hardware and software that make up the Component at some registry. Verification of authenticity can confirm that the components that make up a PETs Space have not been tampered with, but it cannot guarantee that each Component is honest. Similarly, transparency cannot prevent malicious or compromised Components, but it can hold the Component provider accountable.

Trust model between PETs components
Each component performs mutual authentication and attestation and forms a PETs Space by trusting each other.

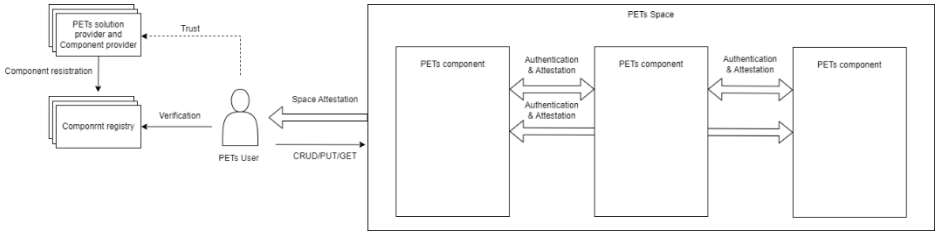

Figure 5.3-1: High level trust model of IOWN PETs

Assurance classes for PETs Space

There are several possible variations for the verifiability of authenticity. This document defines the following three classes of assurance for the PETs Space. PETs Space must provide one of them to PETs Users.

PETs Space assurance class 1 (PAC 1)

In PAC1, PETs Users verify the authenticity of PETs Space by comparing the PETs Space configuration information presented by the PETs Space with the component information with digital signatures added by the component provider to protect from tampering previously published in the component registry or elsewhere. This Component information at Component registry is added digital signatures by the component provider to protect from tampering. The PETs Space configuration information or its values for verification of its authenticity provided by the PETs Space may be signed with software-protected keys such as those stored in an access-controlled keystore by the PETs Space provider own for providing verifiability of its integrity as well.

Entities that PETs Users need to trust to get a guarantee of authenticity for a PETs Space in PAC 1: PETs Space providers and Component providers.

PETs Space assurance class 2 (PAC 2)

In PAC2, the PETs User can verify the integrity of the PETs Space configuration information guaranteed by a third party and then check it against publicly available information with digital signatures added by the component provider to protect from tampering at a registry to verify its authenticity. This class requires the PETs User to trust this third party additionally. Examples of integrity-verifiable information provided in this class include digital signatures by a trusted third party.

Entities that PETs Users need to trust to get a guarantee of authenticity of a PETs Space in PAC 1: PETs Space providers, Component providers and trusted third parties.

PETs Space assurance class 3 (PAC 3)

In PAC3, the PETs Space configuration information presented by PETs Space providers is given information that can verify its integrity based on the RoT. This class requires the PETs User to additionally trust the vendor of the hardware that provides the RoT. An example of means to verify integrity provided in this class is a remote attestation that is signed with a signature key stored in the RoT of a TPM or a TEE.

Entities that PETs Users need to trust to get a guarantee of authenticity of a PETs Space in PAC 1: PETs Space providers, Component providers and hardware venders providing RoTs.

Specific trust model for PAC 1

At PAC 1, PETs Users trust PETs Space by verifying that the PETs Space configuration information provided by the PETs Space provider as PETs Space attestation is the same as the publicly available information. The PETs Space provider and Component provider register and disclose their own software in advance. Then the PETs Space provider provides a PETs User with the PETs Space configuration information as a PETs Space attestation. The PETs User verifies the authenticity of the PETs Space configuration information using disclosed information at registries after verification of integrity of the PETs Space configuration information with a signature signed by the PETs Space provider. Each Component performs mutual attestation with information about configuration signed by Component providers as the same as a PETs User does to a PETs Space and forms a PETs Space by trusting each other.

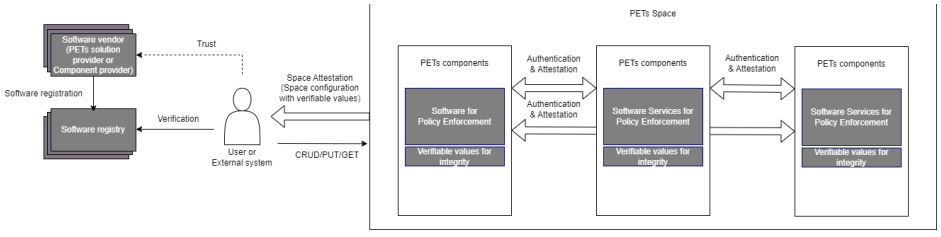

Figure 5.3-2: Trust model of IOWN PETs for PAC 1

Specific trust model for PAC 2

At PAC 2, PETs Users verify the integrity of the configuration information guaranteed by a third party before confirming that the PETs Space configuration information provided by the PETs Space provider is the same as the publicly available information through a PETs Space attestation. PETs Users verify the integrity of PETs Space's configuration information based on a mechanism provided by a third party such as a digital signature. Each Component performs mutual attestation with information about the configuration, whose authenticity is guaranteed by a third party and forms a PETs Space by trusting each other.

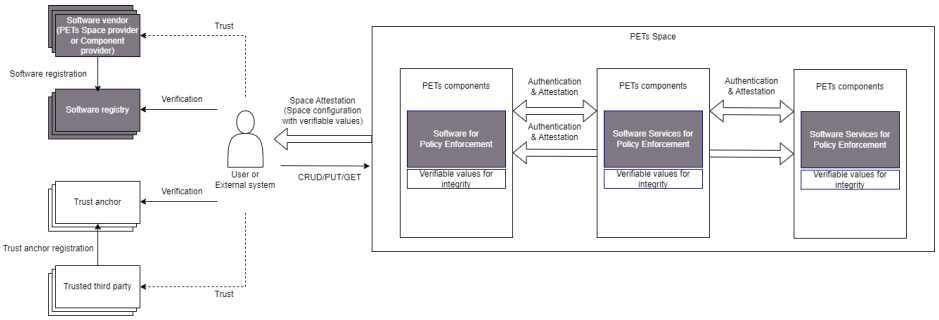

Figure 5.3-3: Trust model of IOWN PETs for PAC 2

Specific trust model for PAC 3

At PAC 3, the PETs Space provider provides the PETs User with a PETs Space attestation that can validate the entire PETs Space configuration based on the RoT. The PETs User can verify the authenticity of the PETs Space configuration by trusting the hardware vendors that provide the RoT. PETs Users verify the integrity of PETs Space's configuration information based on trust anchors provided by the hardware vendor. The authenticity is then verified by confirming that the components comprising the PETs Space are identical to the hardware and software previously registered in the registry. Each Component performs mutual attestation in environmental units based on the same RoT and forms a PETs Space by trusting each other.

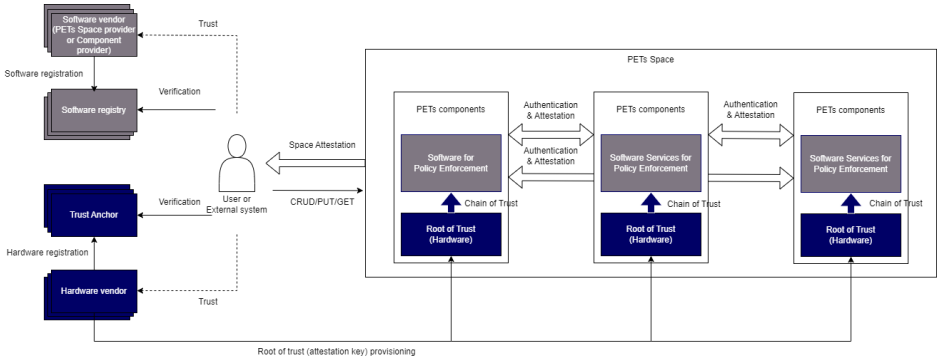

Figure 5.3-4: Trust model of IOWN PETs for PAC 3

6. Examples of IOWN PETs Procedures and Usage

This section describes examples of basic IOWN PETs usage from user's point of view. The following examples are common for each PAC defined in section 5.3.

Example 1: Simple data I/O (from user's point of view)

Figure 6-1 shows the procedure for a user to input data to the PETs Space. Space User searches for Space X, exchanges keys for cryptographic communication with Space X, and then sends encrypted data to Space X.

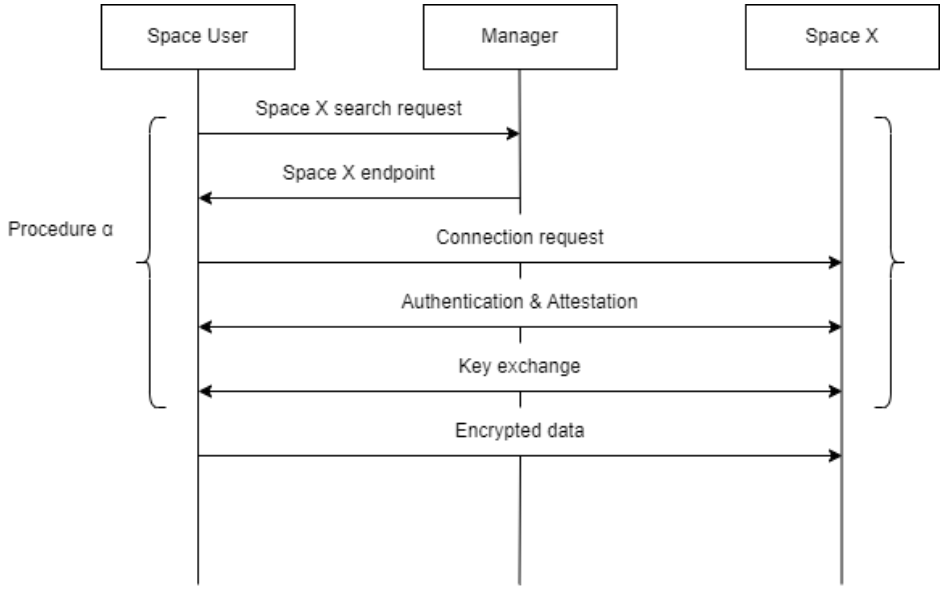

Figure 6-1: Simple Data I/O procedure

Example 2: Simple secure computation (from user's point of view)

Figure 6-2 shows the procedure for multiple users to perform secure computation using the PETs Space. Space User A sends an application to Space X. Space User B sends data and request execution of the application to Space X. Space User C gets the result from Space X.

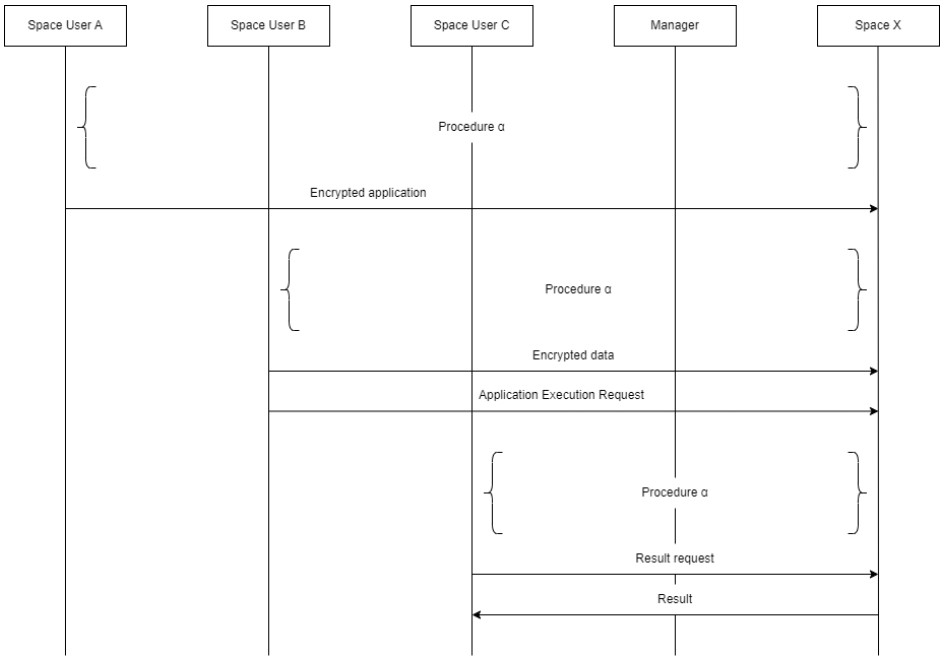

Figure 6-2: Simple Secure Computation procedure

Example 3: New space creation (from user's point of view)

The figure 6-3 shows the procedure for a Space User to create a PETs Space that combines heterogeneous resources. Space User makes a PETs Space creation request to the Manager, specifying the Tenant and the PETs Space creation method as arguments. After Space X (in this example, the PETs Space consists of executing environments on CPU and GPU) is created, the Space User sends data to the Space X (CPU). Space X performs computational processing using the CPU and GPU in coordination.

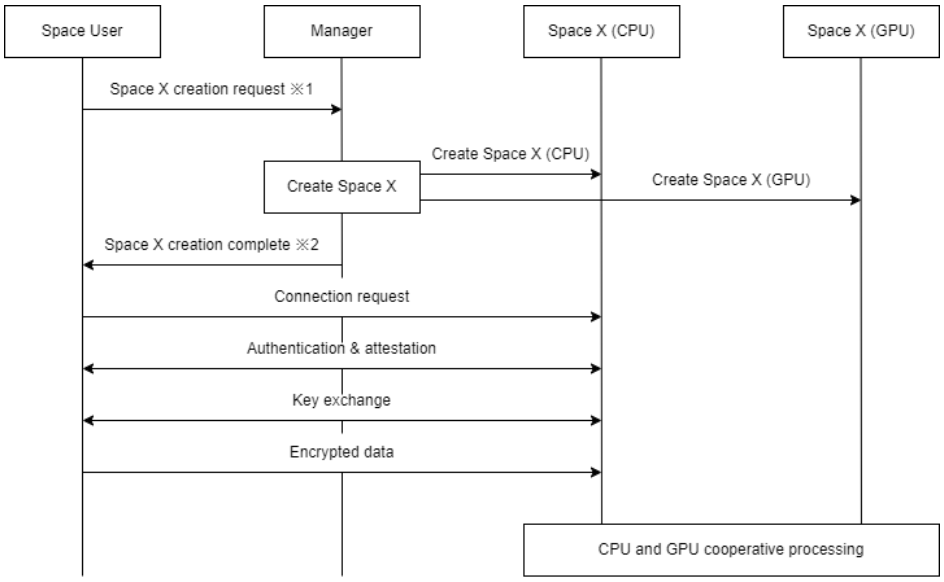

Figure 63: New Space creation procedure

*1 Example of the request

Create a PETs Space that combines VM resources using AMD SEV-SNP and GPU resources using NVIDIA H100 under Tenant A.

*2 Example of the return value

Identifier and endpoint of Space X.

7. Benefits to Users of the IOWN PETs Architecture

From here, the following two solutions to the issues in Section 2 that can be achieved by introducing IOWN PETs on top of the IOWN GF infrastructure are described.

- Advanced secure computational PETs Space supporting multiple users with different rights and roles
- Distributed Sovereign Hybrid Cloud

7.1. Advanced secure computational space supporting multiple users with different rights and roles

IOWN PETs can be used to implement a secure computation space supporting multiple users with different rights and roles as described in Section 2.1. This space provides data confidentiality and data governance across the entire data lifecycle, which existing technologies such as IDSA/GAIA-X have not been able to achieve, even if it is data collaboration among multiple users with different rights and roles.

The points to be improved by IOWN PETs are below.

- Protection throughout the data lifecycle from data ingestion to data processing and data use. The IOWN PETs provide protection on the system level for data processing and its application as well. Therefore, all data exchanges among a data provider, an application provider and a data consumer are protected.
- Improved data sovereignty. In PETs Space, all sharing and use of data and applications are controlled by policies that can be specified by users. Therefore, multiple stakeholders including a data provider and an application provider can manage the confidentiality of their data and application more flexibly.

7.2. Distributed sovereign hybrid cloud

A combination of the IOWN infrastructure and the IOWN PETs creates the distributed Sovereign Hybrid Cloud where various types of users exchange and process their confidential data flexibly and securely using public clouds as well as private and domestic clouds for storing data. Moreover, the distributed PETs Space across multiple DCs provides secure data distribution throughout their lifecycle even if data providers and data consumers are geographically distributed.

The points to be improved by IOWN PETs are below.

- An open mechanism that is independent of specific vendors or operators, making it easier to scale PETs Spaces (e.g., distributed sovereign hybrid cloud) where data distribution is possible while maintaining data sovereignty.
- The ultra-fast connectivity and disaggregated computing provided by IOWN GF's Open APN and DCI can be used together with IOWN PETs to form a secure virtual closed space across multiple DCs on a global scale.

8. Conclusion

This document provides an IOWN PETs architecture that can guarantee the technical protection of data according to the data protection policy of the data owner consistently throughout the data lifecycle (data in motion, data at rest, and data in use) to promote active data distribution on the IOWN infrastructure. By following this architecture, many hardware and software vendors can participate in the PETs World to achieve solutions such as advanced secure computational space supporting multiple users with different rights and roles and distributed sovereign hybrid cloud. We expect that these solutions will enable active data collaboration on IOWN infrastructures.

References

[IOWN Data Hub]	IOWN Global Forum, “Data Hub Functional Architecture,” 2023. https://iowngf.org/wp-content/uploads/formidable/21/IOWN-GF-RD-Data_Hub_Functional_Architecture-2.0.pdf
[IOWN Open APN]	IOWN Global Forum, “Open All-Photonic Network Functional Architecture,” 2022 https://iowngf.org/wp-content/uploads/formidable/21/IOWN-GF-RD-Open-APN-Functional-Architecture-1.0-1.pdf
[IOWNSec]	IOWN Global Forum, “Technology Outlook of Information Security,” 2022. https://iowngf.org/wp-content/uploads/formidable/21/IOWN-GF-RD-SEC-Technology_outlook_of_Information_Security.pdf
[TEE]	Confidential Computing Consortium, “Basics of Trusted Execution Environments (TEEs): The Heart of Confidential Computing – Confidential Computing Consortium”, 2018.
[FL]	https://aws.amazon.com/jp/blogs/machine-learning/reinventing-a-cloud-native-federated-learning-architecture-on-aws/
[IDSA]	https://internationaldataspaces.org
[Catena-X]	https://catena-x.net/en/
[Confidential Computing Consortium]	https://confidentialcomputing.io/
[Common Terminology for Confidential Computing]	Confidential Computing Consortium, “Common Terminology for Confidential Computing“, 2022 https://confidentialcomputing.io/wp-content/uploads/sites/10/2023/03/Common-Terminology-for-Confidential-Computing.pdf

Abbreviations

A

AMD SEV-SNP, AMD Secure Encrypted Virtualization-Secure Nested Paging

API, Application Programming Interface

C

CPU, Central Processing Unit

CRUD, Create, Read, Update, Delete

D

DC, Data Center

DCI, Data-Centric Infrastructure

DCIaaS, Data Centric Infrastructure as a service

F

FL, Federated learning

G

GPU, Graphics Processing Unit

H

HE, Homomorphic Encryption

I

IDH, IOWN Global Forum's Data Hub

IDSA, International Data Space Association

I/O, Input/Output

IOWNSec, IOWN Global Forum’s Security

IRM, Information Rights Management

P

PETs, Privacy-enhancing technologies

S

SBOM, Software Bill of Materials

SS-MPC, Secret sharing-based MPC (Multi-party computation)

STRIDE, Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege

T

TDE, Transparent Database Encryption

TEE, Trusted Execution Environment

V

VM, Virtual Machine

O

OpenAPN, Open All-Photonic Network

OS, Operating System

History

Revision	Release Date	Summary of Changes
1	October 2024	Initial Release