CYBERNETICA

JOCONDE D1.1 Usage Scenarios and System Requirements

Technical report

Version 1.0

20.01.2025

D-16-455

Public

Project JOCONDE (Joint On-demand COmputation with No Data Exchange)
https://cros.ec.europa.eu/joconde

Project Managers:	Fabio Ricciato (Eurostat)
	Baldur Kubo (Cybernetica)
Authors (Cybernetica):	Riivo Talviste,
	Kert Tali,
	Mihkel Haav,
	Hendrik Eerikson

Cybernetica AS, Mäealuse 2/1, 12618 Tallinn, Estonia.

E-mail: info@cyber.ee, Web: https://www.cyber.ee, Phone: +372 639 7991.

Date	Version	Description
23.09.2024	0.2	Advanced draft
01.11.2024	0.3	Revised draft
20.11.2024	0.4	Revised draft
03.12.2024	0.5	Revised draft
20.01.2025	1.0	Revised version for publication

Disclaimer

This document was prepared by Cybernetica AS as part of a procured project under Service Contract No ESTAT 2023.0400.

The opinions expressed in this document are those of the authors. They do not purport to reflect the opinions, views or official positions of the European Commission.

This is an intermediate report from the project. Parts of its content may be revised and changed in future version updates in the course of the project, also based on the feedback received by expert readers. The final version of the present report is planned to be released at the end of the project in March 2026.

We encourage readers to provide feedback. If you would like to suggest corrections or improvements, or simply ask for clarifications, we are happy to hear from you. Please contact us via email at joconde@cyber.ee. Your feedback may help to improve future versions of this document.

Table of Contents

1 Introduction

Eurostat produces European statistics in partnership with National Statistical Institutes (**"NSIs"**) and other national authorities (**"ONAs"**) in the European Union (**"EU"**) Member States. The partnership between Eurostat, NSIs and ONAs constitutes the European Statistical System (**"ESS"**). Eurostat also coordinates statistical activities at EU level and more particularly inside the European Commission (**"EC"** or **"Commission"**).

As the demand for new and timelier statistical and analytical information increases, the ESS is working towards the development of new statistical products based on the integration of multiple data sources, including new types of non-statistical data sources (e.g., privately held data). This will lead to a proliferation of cross-organisational processing instances where confidential data held by two or more organisations must be integrated, i.e., processed jointly, to produce the desired statistical indicator. This requires novel and stronger approaches to protect the confidentiality of the source data *during the processing stage*, in addition to usual means to prevent disclosure of individual information at the dissemination stage. Given the above background, Eurostat conceived the idea of a shared ESS infrastructure, based on modern privacy-preserving computation technologies, enabling ESS members and their partners to perform cross-organisational integration of confidential data. The initial conceptualisation of such infrastructure, under the name of Multi-Party Secure Private Computing-as-a-Service (**"MPSPCaaS System"**), benefited from the participation of Eurostat in the Input Privacy Preservation Project [1] coordinated by the United Nations Economic Commission for Europe (**"UNECE"**).

In order to move forward towards the development of such a system, in 2023 the Commission (Eurostat) issued an open call for tender for the *specification, feasibility analysis and prototype demonstration of a multi-party secure private computing system for processing confidential sets of micro-data across organisations in support of statistical innovation*[1] (**"Procurement"**). The tender was awarded to Cybernetica AS, an IT company specialised in security and privacy technologies from early stage of research to production deployments (**"Cybernetica"**). Cybernetica's core expertise is in mission-critical systems, operational technology, information security, cryptography, and protocol analysis. Following the successful finalisation of the Procurement procedure, in 2024 the project titled "JOCONDE" (**"Project"**)[2] was launched between Eurostat and Cybernetica. The Project's main goal is to develop a detailed set of specifications for the envisioned System and to demonstrate its feasibility through the implementation of a prototype version based on these specifications (**"Prototype"**).

The Prototype is going to be an early version of a future production System conceived to allow ESS members and their partners to perform on-demand Secure Private Computing (**"SPC"**) tasks on their respective data without sharing the data in intelligible form, neither with each other nor with an external Trusted Third Party (**"TTP"**).

The envisioned System (and its Prototype) will leverage Privacy-Enhancing Technologies (**"PETs"**) to allow the users to perform computations and extract insights from the integration of multiple confidential data sets without revealing the underlying data to any party, including the System owners and administrators. The System will be designed to prevent the disclosure of source

[1] Tender reference number ESTAT/2023/OP/0004. For further information, please see the TED eTendering website: https://etendering.ted.europa.eu/cft/cft-display.html?cftId=12503

[2] Project website: https://cros.ec.europa.eu/joconde/

data, leakage of any intermediate data, execution of unauthorised computations, and delivery of any other information beyond the final results of the predefined computation.

1.1 Purpose and scope

The work in Project JOCONDE is structured into six distinct tasks:

- Task 1 – Usage scenarios and system requirements
- Task 2 – Technology analysis
- Task 3 – Legal aspects
- Task 4 – System specifications and architecture
- Task 5 – Demonstrator prototype and functional testing
- Task 6 – Trust building plan

This document, *D1.1 Usage Scenarios and System Requirements*, is the first deliverables from Task 1. It covers key parts of system analysis that will guide the legal analysis, system design and implementation to be further elaborated in subsequent deliverables from the relevant Project Tasks. More specifically, the usage scenarios (Chapter 5) provide input to the Project's legal analysis work package as well as prototype implementation and technical tests. The security goals (Chapter 3), System description and processes (Chapter 4), usage scenarios (Chapter 5) and requirements (Appendices A and B) guide the System architecture and implementation.

1.2 Overview

D1.1 starts off in Chapter 2 by addressing the business vision for the envisaged System by synthesising the business requirements based on the Procurement and discussions with the Project representative of Eurostat. Chapter 3 establishes the security goals for the System and formalises the initial System requirements regarding security. The description of the System, along with processes and all other System requirements follow in Chapter 4. Lastly, use case descriptions are presented in Chapter 5.

Throughout this document, the established requirements are first introduced in-line with the main content, supported by the text that precedes it. This enhances traceability and clarity by connecting the requirement with the original reasoning for it. The in-line requirement block is structured as follows:

> **<Category>-<Group ID>.<Requirement ID>** ***Requirement*** *(<Group>)*
> *<Requirement text>*

The full set of requirements are presented within the appendices – business requirements in Appendix A and system requirements in Appendix B – along with supplementary details.

2 Vision

The Vision chapter introduces the key concepts, requirements and objectives of the JOCONDE System. Readers will gain insight into the challenges associated with traditional statistical production methods and how these are addressed by the System. The concept of servitising Secure Private Computation technologies is introduced. This chapter also defines the stakeholder roles that are used throughout the rest of the document.

The rules and requirements introduced in this chapter form the foundational principles that will guide the development of the JOCONDE System and ensure that subsequent work is aligned with the Project's goals.

2.1 Problem statement

Traditional statistical production involves a single authority in charge of collecting and analysing all data. When integration of multiple data sets are needed, they are often exchanged between organisations, with agreements on allowed usage and security. This approach relies heavily on trust and may not suffice in future scenarios where the integration of large-scale and highly granular micro-data poses increased risk and therefore requires proportionally stronger protection measures. Secure Private Computation (**SPC**) technologies offer an alternative to the traditional approach of sharing intelligible data. SPC technologies, also known as Input Privacy technologies, aim to enable the computation of desired statistics without revealing the underlying input data to any entity except their original holder [1, 2].

The JOCONDE Project focuses on cross-organisational computation scenarios where the desired computation task takes in input data held by multiple parties. For such scenarios, the most relevant SPC technologies are Secure Multi-Party Computation (**MPC**) and Trusted Execution Environments (**TEE**). For an up-to-date overview of the current state of the art of these technologies see *D2.1 Technology Survey and Analysis*.

The concept of Secure Private Computation (SPC) has applications beyond just statistical analysis, extending to various different scenarios. However, SPC alone does not specify the operational processes of a complete functional system. To utilize SPC technologies within the field of statistics, a broader ecosystem is required, including user interfaces, operational processes, organisational structures and more.

Integrating Secure Multi-Party Computation (MPC) and Trusted Execution Environments (TEE) into a practical system requires careful consideration and thorough analysis. These technologies have unique assumptions and requirements that may not align with conventional information systems, for example, ensuring the presence of multiple parties that do not collude or maintaining the control of source data throughout its usage. Therefore, the implementation process might not be straightforward, necessitating the design of novel workflows. The objective of the JOCONDE Project is to develop a well-rounded SPC system that includes comprehensive processes, specifically tailored for the statistical setting, while maintaining the robust privacy and security assurances provided by the SPC technologies.

The consultation with field experts and prospective users conducted during the UNECE Input Privacy Preservation Project [1] show that adoption of SPC technologies in statistical production is impeded by several factors, among which are the development and deployment costs of SPC solutions and the lack of SPC specialists and other relevant skills among potential adopters.

Adopting a servitisation model, inspired by the NIST Definition of Cloud Computing [3] in which a shared infrastructure deployment is employed to serve the data analysis needs of many institutions, could represent an effective approach to deal with the cost and skill barriers [1]. By servitising SPC, specialist resources can be pooled to address technical and organisational complexities in the development phase, so as to offer a robust and streamlined experience in the subsequent usage phase to a multiplicity of prospective users. This manner of front-loading hard engineering problems has the potential to enhance the quality (utility, security) and at the same time drive down the costs of using SPC technologies compared to dedicated, ad-hoc solutions. The term **JOCONDE System** (or just **System**) is used hereafter to describe a system that offers SPC services on demand, with the least possible compromises to the ease of use compared to traditional computation.

A service dedicated to provisioning SPC must be specifically tailored to the needs and constraints of the domain it is intended to serve. In this project, the System will be designed around the needs, capabilities and legal context of the ESS. As a consequence, the System might not fit without modification to other domains (e.g., other public sectors) for reasons such as differing scale, jurisdiction or technological capabilities. Nonetheless, we believe that several concepts, design aspects and components of the System may be reused in other contexts and the System may serve as a reference or inspiration for SPC provisioning developments in other domains.

The System is envisioned as a single instance within the ESS domain to serve all NSIs and their partners, to be used for the development and production of European and national statistics. Note that the System is not designed to be used for disseminating the statistics – dissemination shall be handled by the users who obtain computation results from the System.

Since the System exploits an as-a-Service model, up-front mapping of business processes of all prospective users is not viable even in the constrained context. For this reason, the business rules and processes associated with the service's users are provided by Eurostat/EC. Prospective users are expected to align the processes into their organisation on an individual basis.

2.2 Planes

The notion of "Computation Tasks" is central to the envisioned concept. The Computation Task represents the service unit offered by the System, i.e., an instance of secure private computing agreed by a small set of client organisations (at least two). By specifying a Computation Task, the involved organisations must agree on (i) what data shall be provided in input and by who, (ii) what computation function shall be executed by the System on this data and (iii) who will receive the final result(s) of the pre-defined computation function. The JOCONDE System offers its Clients the service of specifying, agreeing on, and executing Computation Tasks.

Taking inspiration from the architecture of connection-oriented telecom and computer networks, the System shall be organised into three distinct *planes*: the Management Plane, the Control Plane, and the Data Plane.

- **The Management Plane** gathers all horizontal functions that are independent from the Computation Tasks. It includes activities such as System setup and maintenance, System auditing, Member management, and enforcing System-wide policy.
- **The Control Plane** gathers all the functions necessary for setting up new Computation Tasks and tearing them down after completion. It includes all the functions involved in the workflow by which a group of Clients define, approve and launch new Computation Tasks to be executed by the System.

- **The Data Plane** leverages SPC technologies such as MPC and TEEs to execute the Computation Task established in the Control Plane. The Data Plane must ensure confidentiality and integrity of the data during the whole processing phase, allowing only the final result(s) of the desired computation function to be delivered (only) to the pre-defined stakeholder(s). The Data Plane must provide mechanisms to securely erase all exchanged data upon Computation Task termination.

Organising the functions of the System in the three different planes brings multiple benefits. First, such a structure helps to categorise the different activities (processes) and responsibilities (roles) in the System. Second, it allows designing, implementing and maintaining the software components related to the different planes independently. By establishing clear interfaces between the planes, they can be developed and evolved independently from each other, thus simplifying the maintenance of the overall System.

2.3 Roles

Each stakeholder interacting with the System assumes one or more roles. Each "role" represents a particular relationship between the stakeholder and the System. A single stakeholder may perform multiple roles even in the context of a single Computation Task (e.g., by providing input data and receiving output results). We have identified the five main roles described below.

- **Input Parties (IPs)** provide Restricted Data as input to the Computation Task[1]. The IP role may be assumed by NSIs, by other ESS members or by their partners, e.g. Private Data Holders or public administrations that do not belong to the ESS system but cooperate with their members. IPs are responsible for the preparation and submission of data to the System. They also take part in the specification and approval of the Computation Task that involves their data.

> **BUS-1.1** ***Requirement*** *(System roles)*
> *Input Parties shall make their Input Data available in protected form to the Computation Task.*

- **Output Parties (OPs)** define the parameters of the Computation Tasks in collaboration with IPs. OPs are responsible for initiating the Computation Tasks and retrieving the results upon completion. For example, NSIs or Eurostat may act as OPs for Computation Tasks intended to produce national or European statistics respectively.

> **BUS-1.2** ***Requirement*** *(System roles)*
> *Output Parties shall extract results of the Computation Task to use in a statistical product.*

- **Computing Parties (CPs)** execute the Computation Task. Each CP provides the IT resources for one Computing Node, enabling secure execution and warding off unauthorised access at its node. A single Computation Task requires cooperation by at least three mutually independent CPs in the Data Plane.

> **BUS-1.3** ***Requirement*** *(System roles)*
> *Computing Parties shall contribute resources to ensure availability of the System to carry out secure computations on demand.*

[1] The term "Restricted Data" denotes data that must be kept secret from other Members of the System and third parties due to regulatory (e.g. data protection or confidentiality requirements) or other reasons. Before uploading it to the System, it is made illegible to others (e.g., by encrypting or secret sharing it). This process is detailed in Section 4.4.2.

- **System Operator (SO)** manages and maintains the overall System. The SO oversees the overall IT infrastructure and manages the processes of onboarding new Members and revoking membership rights for past Members. It takes responsibility for ensuring that Computation Tasks comply with the usage policy as defined in the System Agreement. The SO ensures continuous monitoring of the System and arranges for independent auditing.

  BUS-2.1 ***Requirement*** *(Governance roles)*
  The System Operator organises the selection, onboarding, and attesting of Computing Parties, checking compliance with all legal and technical requirements before they are admitted to operate in the System.

  BUS-2.2 ***Requirement*** *(Governance roles)*
  The System Operator appoints three Computing Parties to act as the default for all Computation Tasks for which the System Members have not identified specific Computing Parties.

  BUS-2.3 ***Requirement*** *(Governance roles)*
  The System Operator appoints System Auditors.

  BUS-2.4 ***Requirement*** *(Governance roles)*
  The System Operator operates the System components responsible for collaboration and coordination mechanisms in a manner which, to a reasonable degree, rules out any possibility of attacks against System Members and their data.

- **System Auditor (SA)** acts in the interests of System Members. Given the right set of tools and access rights (e.g., to software code and execution logs), the SA shall periodically monitor the System to verify correct behaviour. The SA must expose and duly report any deviation from correct behaviours, including accidental errors and deliberate attempts to break or attack the System. The latter includes actual or potential actions by System Members, System Operator or external intruders aimed at deceiving the System, exceeding the given authorisation rights, and in general any violation of the System Agreement that ultimately could put Restricted Data at risk of exposure.

  BUS-2.5 ***Requirement*** *(Governance roles)*
  A System Auditor shall audit the System to detect misbehaving components or actors including the System Operator, System Members, Computing Parties, and external entities that could put Restricted Data at risk.

  BUS-2.6 ***Requirement*** *(Governance roles)*
  The System Auditor should be independent from the Computing Parties.

As system operations and functions are organised into three planes, it is convenient to represent the mapping of roles to planes. This is illustrated in Figure 1.

The five roles introduced above can be grouped hierarchically as depicted in Figure 2. This structure comes in useful when referring to operations, functions and responsibilities that involve a particular sub-group. We shall refer to the collection of all five roles as **System Participants**. They can be divided into System Providers and System Members.

System Providers include the System Operator and the System Auditor. These two roles are positioned in the Management Plane of the System (see Figure 1) and take care of System governance activities.

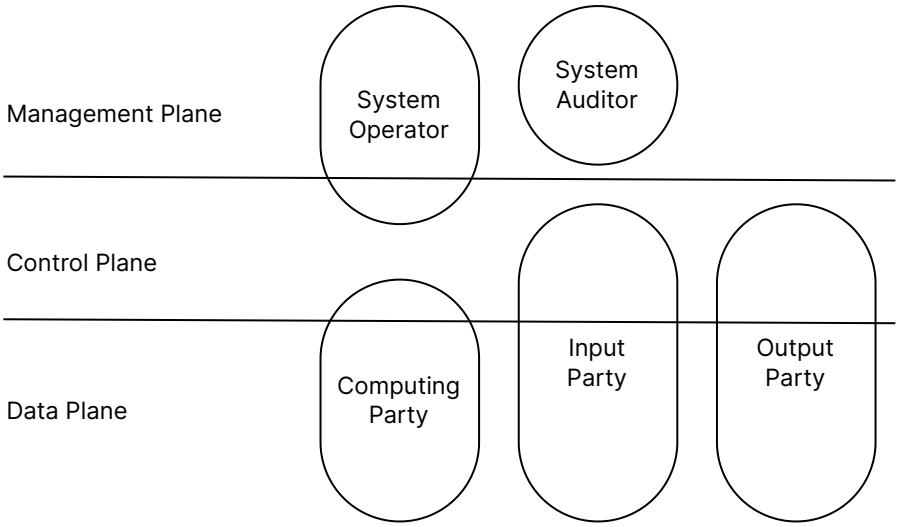

Figure 1. Mapping of System roles to System planes. The System Operator is mostly active on the Management Plane, and its involvement in the Control Plane is limited to signing off Computation Task Specifications together with Input and Output Parties. The System Auditor's functions are entirely contained in the Management Plane. Input and Output Parties create and manage Computation Task Specifications on the Control Plane and interface with Computing Parties on the Data Plane. Computing Parties are mostly active on the Data Plane, and their involvement in the Control Plane is limited to handling the Computation Task Agreements, i.e., signed versions of the Computation Task Specifications.

The System Operator manages access to the service. This involves organising the onboarding processes of new System Members by which the new stakeholders prove their identity and qualification, sign the System Agreement, and receive credentials for using the System.

> **BUS-2.7** ***Requirement*** *(Governance roles)*
> *System Members shall undergo a manual onboarding procedure with the System Operator for ensuring conformance with all legal and technical requirements set in the System Agreement before being able to use the System.*

The System Operator operates all System components (or subsystems) that are not under the responsibility of the Computing Parties (Computing Nodes). None of these subsystems handle Restricted Data and therefore the System Operator never comes into contact with it. These subsystems are characterised as supporting services that aid the creation and execution of Computation Tasks or enable access management. They are part of the Management Plane. For example, the Client Portal is a subsystem operated by the System Operator, it is described in more detail in Chapter 4.

Centralising the operation of Management Plane subsystems to a single stakeholder, the System Operator, improves the maintainability of the System without introducing a single point of trust.

> **BUS-2.8** ***Requirement*** *(Governance roles)*
> *The System Operator shall not be a single point of trust.*

The System Operator is not a single point of trust because all of the functions and components in the Control Plane and Data Plane that may influence the processing of Restricted Data are implemented in a distributed privacy-preserving manner among the System Members who are separate from the System Operator. In other words, the carefully designed combination of

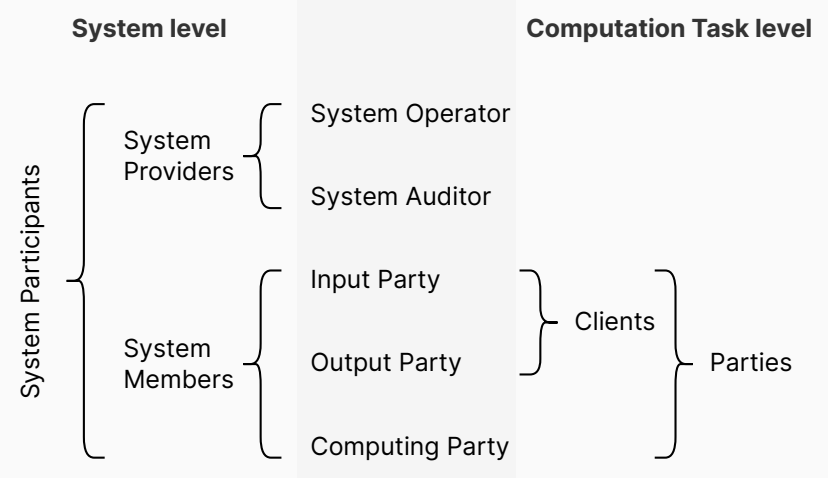

Figure 2. Hierarchy of roles. All System Members have gone through an onboarding process. Clients cooperate in planning new Computation Task that are then executed collaboratively by the Computing Parties.

centralised and distributed components, together with role separation, achieves a high level of system security while preserving system manageability.

System Members (also called simply **Members**) refer to stakeholders that interact with the System for the purpose of carrying out Computation Tasks. Input Parties, Output Parties, and Computing Parties are all System Members. In the context of the JOCONDE System, stakeholders taking the role of System Member will be legal persons with IT systems who have gone through a Member onboarding process coordinated by the System Operator[2]. All stakeholders that want to partake in a Computation Task must pass the Member onboarding process as a preliminary. The Member onboarding process is a function of the Management Plane. The Member onboarding process ensures that:

- their identity has been verified;
- they have agreed to the System Agreement;
- they have been granted access to the relevant System services;
- their credentials are recognised by the System and therefore also by the other Members.

The onboarding process is executed only once and may involve manual administrative and IT operations. This simplifies performing multiple future instances of Control Plane operations dedicated to specific Computation Tasks.

In the context of a specific Computation Task, a System Member is also called a **Party**. Such a dual taxonomy – Members at system level and and Parties at task level – is necessary because the System may support multiple parallel Computation Tasks, each with their own set of Input, Output and Computing Parties, and the same stakeholder may perform different roles in different Computation Tasks.

[2] Although System Members are considered as legal persons (i.e., organisations), it is important to note that natural persons (i.e., employees) are actually in charge of interacting with the System. The term System User is adopted in other Project documents to denote such natural persons authorised by the respective System Member.

In a context of a specific Computation Task, the Input, Output and Computing Parties are all Parties. Input and Output Parties represents the **Clients** of the service provided by the System (refer again to Figure 2). Each Computation Task is always initiated by a Client (initiator) that invites one or more other Clients to join the specific Computation Task. Within a single Computation Task, a single stakeholder may perform both the Input and Output Party roles at once.

> **BUS-1.4** ***Requirement*** *(System roles)*
> *Clients shall be able to independently initiate new Computation Tasks with minimal manual intervention by others to streamline operations and lessen personnel workload.*

> **BUS-1.5** ***Requirement*** *(System roles)*
> *Clients can invite other Clients to participate in their Computation Task as Input or Output Parties.*

The roles of Input and Output Party entail operation in the Control Plane and Data Plane (see Figure 1). In the Control Plane they negotiate and agree on the details of the computation task, i.e., they collaboratively produce the Computation Task Specification (CTS). Forming the CTS may involve multiple rounds of communication among the Clients, i.e., among the Input and Output Parties. We assume that this process takes place offline, outside the System. When the Clients have reached consensus, one of them submits the CTS to the System and invites all other Clients to review and approve it. Approval implies that all parties understand and agree on the computation function that defines the computation result. They also acknowledge and accept any potential residual risk associated with the agreed computation function (e.g., the final result may carry a small but non-zero disclosure risk under certain conditions). By formally approving the CTS, they also commit to participating in the Computation Task as their specified role according to the terms defined in the CTS. A Client expresses its approval of the CTS by digitally signing it. A CTS with signatures from all participating Input and Output Parties is called a Computation Task Agreement (CTA).

> **BUS-1.6** ***Requirement*** *(System roles)*
> *All Input and Output Parties must understand and approve the details of the planned statistical computation, forming a Computation Task Agreement, before it could be carried out to ensure the acknowledgement of any residual risk.*

In the Data Plane, Input Parties are responsible for preparing and providing the input data sets ahead of the computation. Data preparation involves bringing raw Restricted Data to conformance with the data model set in the CTS to ensure proper formatting. The data prepared in this way are then “protected”, i.e., made unintelligible to Computing Parties and external intruders, by applying the agreed privacy-preserving primitive (e.g., encryption or secret sharing) from among the primitives made available by the Control Plane. Finally, the Input Parties upload the Protected Input Data to the Computing Nodes.

At the end of the computation process, the Output Parties use the primitives made available in the Data Plane to fetch the Protected Output Data from the Computing Nodes and reveal them (e.g., by decryption or reconstruction from multiple shares).

As with the Input and Output Parties, The Computing Parties must also be onboarded as a preliminary. They hold no governing power in the System except for the responsibility of managing their Computing Node infrastructure and making it available to the Computation Tasks defined by the Clients. The Computing Node is a crucial component of the System as a whole, and in fact the Computing Party is the role that technically enforces MPC guarantees by assuring its independence from other Computing Parties.

As already said above, a single organisation (stakeholder) can take on several roles. For example, in the JOCONDE System, Eurostat could perform the role of System Operator but also act as Input Party and/or Output Party for some specific Computation Tasks. Furthermore, it could act as one Computing Party. The same organisation performing all of these roles does not jeopardise the security of the System. However, for transparency and building trust, the stakeholder taking the role of System Auditor should be separate and independent from those acting as System Operator and Computing Parties. And most importantly, as iterated above, the Computing Parties must be all independent from each other.

Vendors. In addition to the five roles, **Vendors** are external entities or suppliers who provide products or services to the stakeholders performing some of the above roles on the basis of a contractual relationship. Examples include:

- **Cloud Service Providers (CSPs)** offering hosting services to CPs, most probably as IaaS.
- **TEE technology providers** offer TEE hardware and the accompanying software kit for CPs or CSPs. Products from several different TEE vendors might be used in parallel in a single deployment of the System.
- **System Vendors**: offer software components of the System, including the MPC and TEE technology and other Data Plane, Control Plane, and Management Plane subsystem components.
- **Contractors** offer their services (e.g. reviewing code in Computation Task Specification) for IPs/OPs, but do not need to be System Members themselves.

System roles and their interactions with the System, Vendors and each other are illustrated in the concept map in Figure 3.

2.4 Business rules

2.4.1 Powers and capabilities

For a system to concurrently handle Restricted Data from multiple domains, it is essential to employ strict access control. Access rights to the System shall be designed around the principle of least privilege to ensure that Members can only execute actions which are explicitly allowed for them. This is evidently necessary in the Data Plane where technological measures have to be in place, for example, to allow anyone to only retrieve the Output Data of a Computation Task in which they serve as the Output Party.

> **BUS-3.1** ***Requirement*** *(Powers)*
> *System Members shall have their access rights associated with a given Computation Task, limited in accordance with the principle of least privilege to permit only the minimal set of actions necessary for their role(s) in the Computation Task at hand.*

The System may be optionally configured to require explicit approval by the System Operator (in addition to the Clients) for any new Computation Task. The decision to enable or disable this option is purely a matter of system governance, and may be dictated by legal considerations. We assume this option is enabled by default, since an additional layer of oversight on system usage by the SO on the Control Plane, besides Member onboarding on the Management Plane, appears to be a desirable feature in the ESS context, at least in the early usage phase. However, the System will offer the possibility to disable this option in order to let Clients create new

Computation Task without further supervision by the SO, should this alternative scenario be the preferred one within the ESS.

In addition to the aforementioned proactive measure, we provide the SO with the power to reactively abort any ongoing computation in case it detects potential violation of System Agreement or otherwise anomalous behaviour. Also the Input Parties are given the power to abort an ongoing Computation Task if they detect potential violations of the Computation Task Agreement.

BUS-3.2 ***Requirement*** *(Powers)*
By default, all Computation Tasks shall be approved by the System Operator before execution in order to enforce compliance to System Agreement. The System Operator shall be able to disable this option if the context of the deployment allows it.

BUS-3.3 ***Requirement*** *(Powers)*
The System Operator shall be able to terminate a Computation Task at any time to mitigate a System violation, abuse, or possible harm when detected.

BUS-3.4 ***Requirement*** *(Powers)*
Each Input Party shall be able to terminate an ongoing Computation Task at any time, in line with conditions outlined in the System Agreement, if it detects potential violation of Computation Task Agreement.

The System shall offer a default set of Computing Parties. However, Clients shall be able to select different Computing Parties for their Computation Task. In this way stakeholders acting as Input Parties have the possibility to act also as Computing Party for their tasks, and in this way exert direct control over task execution (jointly with the other CPs), at the cost of deploying more IT resources (i.e., a Computing Node) and additional burden in the onboarding phase. Moreover, in some practical scenarios the choice of alternative CPs may be dictated by legal or business-related considerations.

BUS-3.5 ***Requirement*** *(Powers)*
Input and Output parties shall be able to agree on a custom set of Computing Parties to use for a specific Computation Task.

The Parties involved in a Computation Task must be given the option to verify each other identities directly, without necessarily relying on the claim by the SO. This option is meant to serve as an additional measure against impersonation attacks involving the SO. While not mandatory, the activation of this option is strongly recommended.

BUS-3.6 ***Requirement*** *(Powers)*
Parties involved in a Computation Task shall be able to reliably verify the identity of each other and the Computing Parties to mitigate attacks involving impersonation.

A System Auditor is provided with necessary tools and access rights to all System components, including the subsystems operated directly by SO. The System Auditor can not obtain similar rights to the infrastructure of Computing Parties (i.e., Computing Nodes), but shall be given access to the logs produced by all Computing Nodes for auditing purposes.

BUS-3.7 ***Requirement*** *(Powers)*
A System Auditor should be able to audit the correctness of a Computation Task after its completion.

BUS-3.8 ***Requirement*** *(Powers)*
A System Auditor should have access to all Computation Task Specifications, Computation Task Agreements, Computation Task Logs and System Logs.

2.4.2 Law and contracts

The System must be designed and implemented in accordance with the general rules governing the use of SPC techniques (e.g., non-collusion of Computing Parties in MPC), the quality standards of European statistics (e.g., independence, relevance, accuracy, reliability, comparability, coherence) as well as all other applicable laws and regulations (e.g., European Statistics Regulation, General Data Protection Regulation, Data Governance Act).

On top of complying with mandatory rules, the System should be sufficiently flexible to enable its Members to encode specific aspects of their mutual relationship, agreement and interests – this is where contractual relations and private law step in (e.g., intellectual property rights, trade secret protection, data licensing).

In order to manage this legal and regulatory complexity, the Project shall provide a set of model agreements defining rights and duties of the Members. At the center of the model agreements there will be a set of general standard terms, governing all uses of the System and mandatory for all Members: the **"System Agreement"**.

The combination of the System Agreement and Computation Task Agreement defines the cross-organisational data governance policy for the Members, statically at the System level and dynamically at the Computation Task level. Such policy is technically enforced by the System. Legal requirements will be addressed in separate project deliverables, and specifically in *D3.1 Initial Legal Analysis*.

2.4.3 Policies

Two kinds of policies can be distinguished within the system: those inherent to the System itself, applicable to all Computation Tasks, and others defined by Clients during System operations and specific to a certain Computation Task. The former complements the latter, as the primary assurance of the System is that the Clients' policies are implemented. Client-defined policies are described in Computation Task Specification, expressed in a formal language as code and access restrictions.

BUS-4.1 ***Requirement*** *(Policies)*
Restricted Data access controls for the Computation Task shall be expressed in a non-proprietary formal language.

Only upon approval by all involved parties (i.e. the Clients serving as Input and Output Parties in the Computation Task at hand) will the contents take effect, forming a Computation Task Agreement.

From the overarching theme of protecting Clients' Restricted Data, the System shall ensure that no one could reveal the Input, Output, or any Interim Data other than the Client(s) specified in the Computation Task Agreement.

BUS-4.2 ***Requirement*** *(Policies)*
No single entity shall have the possibility to reveal the Protected Data (Input, Interim, or Output Data), unless explicitly stated in the Computation Task Agreement.

BUS-4.3 ***Requirement*** *(Policies)*
The Computation Task Agreement shall be legally valid and enforceable.

The System may only run computations under the exact terms detailed in the Computation Task Agreement. In MPC, Computing Parties can only execute a computation collectively, based on consensus. As the Computation Task Agreement is formed, Input and Output Parties delegate control to the set of Computing Parties assigned for that Computation Task. The System therefore relies on the assigned set of Computing Parties to strictly follow the Computation Task Specification in all actions.

BUS-4.4 ***Requirement*** *(Policies)*
A Computation Task shall be executed only after approval by all Input and Output Parties.

BUS-4.5 ***Requirement*** *(Policies)*
Computing Parties shall strictly follow the Computation Task Agreement for all decisions in order to counter any possibility of data being used outside of the pre-agreed context.

To split control to a reasonable degree, at least three Computing Parties should be assigned to any individual task. The risk of collusion is minimised by requiring that all Computing Parties assigned to the same Computation Task are fully independent legal entities from each other, not sharing any common authority or shared interest that would motivate collusion.

In principle, the more the Computing Parties the lower the risk of successful collusion, as more stakeholders would need to collude to gain access to the data outside the agreed terms. In practice, every additional Computing Party adds complexity, communication and computation overhead, and a greater likelihood for reducing the overall System availability. Three independent Computing Parties have been demonstrated to be suitable by MPC deployments working with real data and several Input Parties, balancing privacy and availability risks [4, 5].

BUS-4.6 ***Requirement*** *(Policies)*
There shall be at least three Computing Parties for any particular Computation Task.

BUS-4.7 ***Requirement*** *(Policies)*
A Computing Party shall be an independent legal entity from other Computing Parties.

BUS-4.8 ***Requirement*** *(Policies)*
No single entity shall have control over more than one Computing Party involved in a Computation Task.

The System must ensure that all Restricted Data used or produced within a Computation Task must adhere to a clear data lifecycle policy specifying when, or upon which condition, the data is deleted. Upon termination of Computation Task, or after a certain time deadline (specified in the Computation Task Specification) has elapsed, all Protected Input Data and Interim Data (if any) must be deleted or equivalently made permanently illegible. Moreover, the Output Data must be retained no longer than necessary – i.e. until retrieval by the Output Parties or upon reaching some pre-defined time deadline.

BUS-4.9 ***Requirement*** *(Policies)*
Protected Input Data and Interim Data connected to a Computation Task shall be securely deleted or rendered permanently illegible once the Computation finishes or the deadline is reached.

BUS-4.10 ***Requirement*** *(Policies)*
Output Data connected to a Computation Task shall be securely deleted or rendered permanently illegible once the result is delivered to Output Parties or the deadline is reached.

2.4.4 Supplementary rules

Users should be able to use the core functionality of the System without necessarily being proficient in SPC technologies. Complexity inherent to these technologies should be hidden, automated, or otherwise covered by comprehensive instructions and procedural guides. The System should enable Clients and their Contractors to assess the residual risk of partaking in a certain Computation Task.

BUS-5.1 ***Requirement*** *(Supplementary)*
All operations within the System shall be as simple and lightweight as possible for the Clients with only marginal costs in order to lower the barrier of entry for utilising PETs for statistics.

The System should adhere to the principles of as-a-Service platforms, particularly emphasising the aspects of on-demand, self-service and observability. Clients should be empowered to provision and manage Computation Tasks independently through a user-friendly interface with minimal support. Additionally, the System should incorporate observability features providing the Clients insight into the progress, status, errors, or necessary actions regarding a Computation Task. Such transparency will facilitate agile workflows, troubleshooting, and greater awareness.

BUS-5.2 ***Requirement*** *(Supplementary)*
The System shall provide Clients an up-to-date overview of active workflows and status updates for ongoing Computation Tasks.

The System shall expect Input Data in tabular format. Tabular data is the most fitting for a statistical scenario as it integrates well with existing workflows, operations, and data sets. Tabular data offers a flexible yet well-defined structure, allowing for precise specification of required formats and column criteria prior to data exchange, ensuring compatibility with the Computation Task. The tabular data structure, i.e., data model, is part of the Computation Task Specification as described in Section 4.4.1.

BUS-5.3 ***Requirement*** *(Supplementary)*
Computation Tasks shall expect Input Data in a well-defined tabular format.

The System should not be bound to a specific MPC and/or TEE technology. The Data Plane should support multiple MPC and TEE technologies, including proprietary and open-source technologies. This acts as a preventative measure against vendor lock-in. It additionally allows to evolve and adapt the System in light of changing requirements or advancing state-of-the-art. For example, the initial MPC technology might be replaced with one that targets a different point on the security-performance scale in order to keep up with either more strict compliance requirements or increased workload.

The Control and Management Plane components need to be open source, while the Data Plane components may make use of proprietary closed source code if necessary.

BUS-5.4 ***Requirement*** *(Supplementary)*
The Data Plane shall support a portfolio of multiple different MPC and TEE technologies.

BUS-5.5 ***Requirement*** *(Supplementary)*
All Management Plane and Control Plane components shall be open source.

In addition to software-based security measures, i.e. the MPC technology, the Data Plane shall make use of secure hardware technologies. It is critical that the layered solution, combining hardware and software security measures, would provide complementary security guarantees which, in conjunction, offer stronger protection of Restricted Data compared to solely using one or the other. See *D2.1 Technology Survey and Analysis* for an overview of how software- and hardware-based SPC technologies complement each other.

BUS-5.6 ***Requirement*** *(Supplementary)*
The System should incorporate security measures at both the hardware and software levels, complementing one another in order to secure the computation environment and achieve the highest possible degree of protection and trustworthiness.

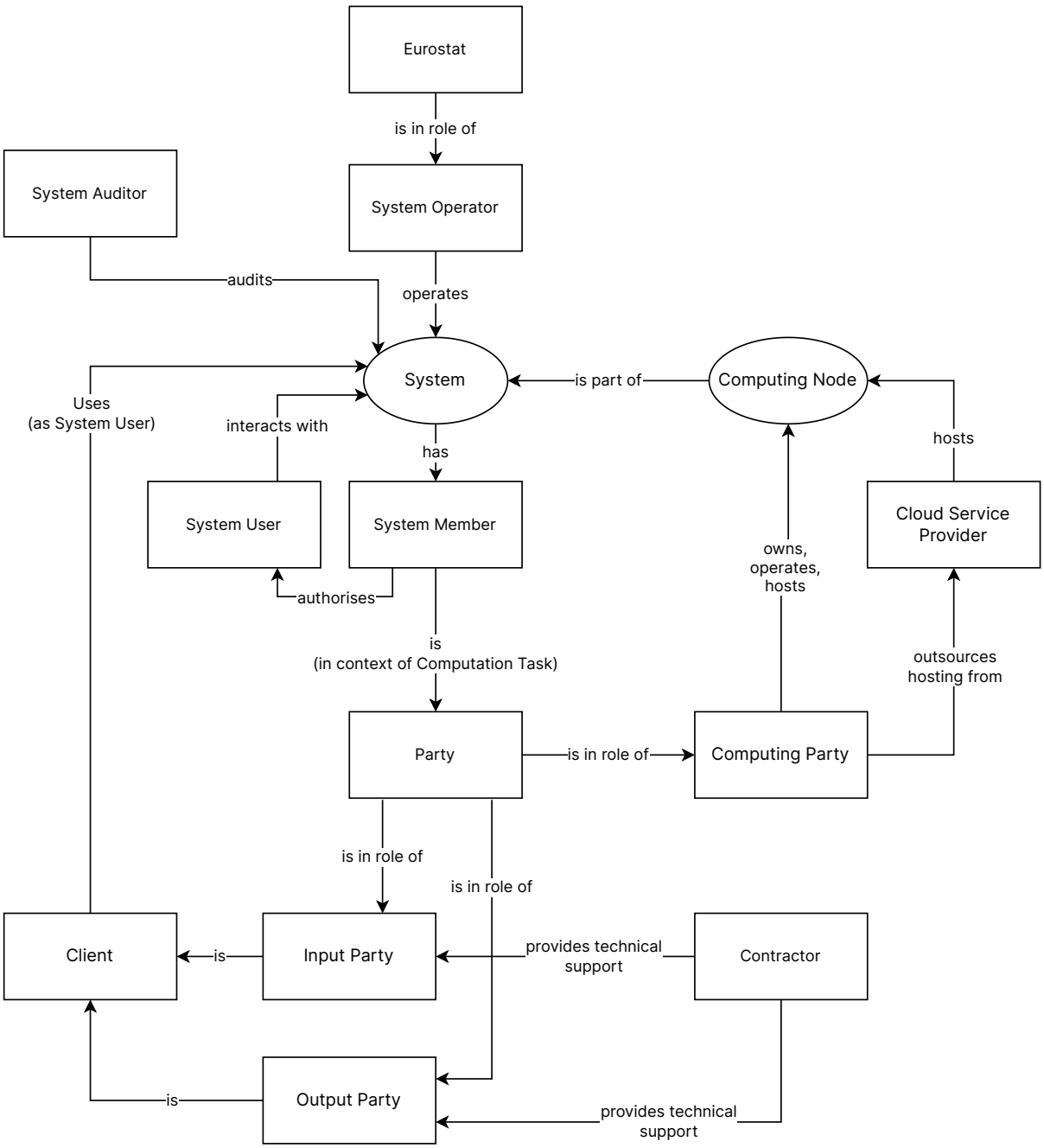

Figure 3. Concept map of roles and their interaction with the System (shown as ovals on the figure)

3 Security aspects

This chapter outlines the primary security goals of the JOCONDE System, emphasising that the security provided by the system must not be less than the *status quo*. The chapter also presents an analysis of the attack surface, including potential points of vulnerability, and the layered security strategies that are employed to mitigate security risks.

3.1 Security goal

In the usage scenarios of the System, the data require a very high level of protection at all stages of the Computation Task execution including transmission, storage and processing. To meet this requirement, Secure Private Computing (SPC) technologies are employed.

We define the security of the System relative to the *status quo* where the System is not used. The data sets processed within the System are never the primary copies of these data, the primary copies reside with the Input Parties who hold the data in the first place. The Input Parties upload the data (more precisely, the subset of data that are strictly necessary to perform the Computation Task at hand) in a protected form. The data remain protected throughout the whole lifecycle of the Computation Task, and are eventually deleted. Only the final result is revealed (i.e., leaves the protected form) to the intended Output Party.

In principle, an attacker interested in obtaining these data sets has a choice between attacking the System and attacking the data sources at the Input Parties' premises.

Given this scenario, the primary security goal of the JOCONDE System can be stated as follows: breaking the System must not be any easier than breaking the data sources, i.e., the Input Parties. In other words, using the JOCONDE System shall not be less secure than the *status quo*.

From this primary security goal, more follow. For example, before the Input Party data are uploaded in the System, they must be protected so that there exists no single point within the System where the data could be retrieved from. Therefore, it follows that the data must also be protected during the Computation Task. After the Computation Task is finished and the Output Parties have received their output, all inputs and other data must be deleted from the System.

Status quo approaches contain a single point of failure, the data origin, where one Input Party's data is stored, presumably protected by conventional measures such as access control or disk encryption. For the System to be harder to attack than the data origin, it must not contain any single point of failure. Therefore, a distributed system which is resistant against infiltration of more than one participant must be used. This requirement warrants the use of multi-party computation in the System.

A detailed list of system requirements related to the security of the System is presented in Section 3.3.

3.2 Attack surface of the JOCONDE System

The attack surface of the JOCONDE System consists of all points in the system where an attacker could compromise data confidentiality or integrity. The following section lists the points of the System which are relevant to the use of SPC technologies.

Data Input and Output. The Input Parties submit their data to the System using software from the JOCONDE System software provider. During data submission, the Input Party's data leaves their organisation. Thus, a data protection mechanism has to be applied before data submission. A conventional encrypted tunnel, using public key encryption, is not sufficient as the end point of the tunnel in the JOCONDE System must not see the data in plain (otherwise it would constitute a single point of trust).

Data output has a similar problem, as Output Data produced by the Computation Task must be submitted to the Output Parties in such a manner that no other parties, even the Computing Parties carrying out the Computation Task, can see the output in plain. An encrypted tunnel is not a sufficient protection mechanism as the entry point of the tunnel must not see the Output Data.

Data Storage. Input Data submitted to the system must be stored until all inputs of a Computation Task have been submitted and the Computation Task can be completed. The output of the Computation Task must be stored until all Output Parties have received it. Conventional encrypted storage is again insufficient as it would entail a single encryption key held by one party.

Computation Task Specification. An Input or Output Party defines a Computation Task using a programmable interface. The Computation Task can include data aggregation and statistical analysis. The output of the Computation Task is received by the Output Parties designated by the Computation Task. An incorrectly defined Computation Task can reveal too much about its inputs. The Computation Task Specification is a major opportunity for an attacker to hide a vulnerability in the system, thus it is important that technical and organisational measures are adopted to ensure that the Computation Task outputs only what is intended to the Output Parties.

Computation Task Execution. The computation on Input Data can not take place in plaintext as that would entail collecting all computation inputs into a single point. Instead, data protection mechanisms must stay in place during the computation. Furthermore, it must be verifiable that the correct Computation Task was executed and that nothing besides the agreed upon computation was done with the Input Data.

Identity Management. In a distributed system like the JOCONDE System, participants also rely on organisational measures to ensure that the security goals are met. For example, all Input Parties must agree on the Computation Task Specification including agreeing on the identities of the Computing Parties and Output Parties. Such organisational measures rely on robust identity management. Otherwise, an attacker could impersonate a Member of the System to, e.g. forge a signature on a Computation Task Agreement.

Other Security Considerations. Concerning availability, in a multi-party system like this, attacks on availability are trivial as a single Computing Party can refuse to take part in the computation and effectively halt the whole process. This risk is mitigated by using organisational controls to incentivise Computing Parties to participate. Other conventional attack on availability, such as denial-of-service attacks, are mitigated using cybersecurity measures such as firewalls. Therefore, we do not consider availability as one of the main objectives in this chapter and concentrate on confidentiality and integrity.

The risk of a quantum breakthrough which renders current public key cryptography vulnerable is mitigated by building a modular system where cryptographic components can be swapped out with newer technologies. The same controls that avoid vendor lock-in (see Section 2.4.4) also help against aging technologies such as obsolete secure tunnel protocols or vulnerable versions of public key infrastructure. As post-quantum secure versions of these components become available, incorporating them should be straightforward if the System architecture is designed in a modular way.

3.3 Proposed security measures

To meet the stated security goals and protect the attack surface of the System, SPC technologies are used in conjunction with conventional cybersecurity tools, practices and organisational methods.

Secure Multi-Party Computation (MPC) enables dividing data and computation between a number of participants such that no single participant can see the data. MPC entails using secure protocols for inputting, outputting, and storing data, and for computing algorithms and complex workflows on said data.

The technology underlying MPC works by splitting data into parts which are indistinguishable from random to any party holding one of these parts. However, by combining them, the original data can be reconstructed. Crucially, this data splitting must have a *homomorphic* property. In other words, random-looking parts of data can be used to compute functions such that the output is in the same split up representation. Recombining this split representation gives the output which is equal to as if the function had been evaluated in plain text. In short, MPC allows data protection to remain in place during computation.

Many specific methods exist for the data splitting, the most notable of which are arithmetic secret sharing, garbled circuits, and homomorphic encryption. In this document, data protected by splitting into random parts is referred to as simply Protected Input/Output/Interim Data[1].

The specific secure data representations and secure protocols for computing with that representation differ in the security guarantees that they offer. The main variables are: does the protocol protect only data privacy or also data integrity, how many Computing Parties are required for the protocol and how many of these Computing Parties can collude without revealing the Protected Data. A detailed discussion of the security of different multi-party computation methods will be given in *D2.1 Technology Survey and Analysis*.

To enable the secure processing of data with the highest sensitivity and granularity, very strict security requirements are set in place that restrict the choice of multi-party computation methods used within the System.

SYS-1.1 ***Requirement*** *(System security)*
The System shall use techniques that split the Input Data elements and/or the encryption keys across multiple nodes by applying secret sharing, multi-key homomorphic encryption or other MPC schemes in combination with secure hardware.

[1] In the Prototype, in most cases, but not all, the precise method of data protection is arithmetic secret sharing, but this detail is mostly omitted in the document.

SYS-1.2 ***Requirement*** *(System security)*
The technology and implementation of choice for any operation on Protected Data shall be robust to collusion of two out of three Computation Parties.

SYS-1.3 ***Requirement*** *(System security)*
The System shall be robust to intrusions at up to two out of three Computing Nodes.

SYS-1.4 ***Requirement*** *(System security)*
The System shall detect attacks on data integrity during the execution of a Computation Task, causing it to halt; an attack shall not cause the deliverance of an incorrect result which cannot be distinguished from a correct result.

SYS-1.5 ***Requirement*** *(System security)*
Upon detecting an attack on data integrity the System should reveal the party at fault.

Trusted Execution Environment (TEE) enables the protection of executing software and its working set (memory, storage, networking and other resources) against inspection and manipulation, essentially providing confidentiality and integrity of data during computation. Although there exist software-based TEEs (e.g., isolated environments provided by virtualisation hypervisors[2]), hardware-based TEEs provide stronger protection and are more common in cloud computing. In the JOCONDE System, only hardware-based TEEs are considered and a detailed overview is given in *D2.1 Technology Survey and Analysis*.

With hardware-based TEEs, the CPU itself manages the protection of the code and working set in a secure enclave, such that even the operating system or hypervisor are unable to break the confidentiality or integrity guarantees. The CPU uses additional secrets to provide remote attestation and data sealing features. Remote attestation allows a remote party to verify that the software running inside the TEE is the expected software. Remote attestation is also used to create a secure tunnel between a remote party and the secure enclave, for example for inputting data. Sealing is used to extend the protection of the data created by the enclave beyond the lifetime of the enclave, such that state can be persisted across restarts, or large swaths of data outsourced to a larger storage medium. Together, a TEE provides protection for data in transit, during computation and at rest.

In the JOCONDE System, both MPC and TEE technologies provide protection of data in transit, during storage and during computation. To increase the level of protection these two technologies are layered on top of one-another, i.e. the software executing within the TEE is the multi-party computation protocol. The remote attestation of the TEE allows the Members of the System to verify that the correct MPC software is executing in the System. In Section 3.4, the added protection of the overlaid technologies is illustrated.

Additionally and crucially, TEEs enable the secure deletion of data which is required to minimise the residual security risk after the JOCONDE System has been used.

The robustness of the System can be further increased by using a diverse selection of TEEs from different hardware providers at the different Computing Parties. With such a strategy, a vulnerability in any one TEE does not affect the security of the overall System. Furthermore, overlaying MPC over TEE brings additional benefit, as elaborated in *D2.1 Technology Survey and Analysis*.

[2] Microsoft. *Virtualization-based security (VBS) enclaves*, https://learn.microsoft.com/en-us/windows/win32/trusted-execution/vbs-enclaves. Last accessed: October 2024

SYS-1.6 ***Requirement*** *(System security)*
The System should incorporate security measures at both the hardware and software level to secure the computation environment.

SYS-1.7 ***Requirement*** *(System security)*
The Computing Nodes shall employ secure hardware technologies, e.g. TEE with hardware isolation.

SYS-1.8 ***Requirement*** *(System security)*
The System shall be robust against side-channel attacks.

SYS-1.9 ***Requirement*** *(System security)*
The System shall be robust against software and hardware attacks.

SYS-1.10 ***Requirement*** *(System security)*
The System should use a combination of multiple technologies and security/privacy layers with complementary security guarantees to achieve the highest possible degree of protection and trustworthiness.

Security measures in the Control Plane. Thanks to the SPC technologies adopted in the Data Plane (i.e., the combination of MPC and TEE) the System only reveals the final result to the Output Parties specified in the Computation Task Specification. It is the responsibility of the Control Plane to ensure that the Computation Task Specification is created and handled exactly as intended by the Clients.

Using the tools and workflows inherent in the Control Plane, the Clients verify the data analysis workflow that will be computed by the System, as well as the identities and roles of all the Parties involved. These details are consolidated in the Computation Task Agreement to be technologically enforced by the Computing Nodes.

The enforcement relies on a local component – authorisation of each Client action towards one Computing Node with respect to the Computation Task Agreement – and the fact that a Computation Task can only be executed collaboratively by several independent Computing Parties as described in Section 2.4.3. Effectively, as long as at least one Computing Node honours the Computation Task Agreement, no entity can successfully execute a request to the MPC infrastructure (e.g. to receive Protected Output Data) for which they are not authorised.

To protect against a compromised or malicious System Operator, Clients are enabled to cross-validate any information presented to them by the System Operator. If the System Operator were to attempt to deceive a Client into approving a maliciously crafted Computation Task Specification, the Client can detect the attempt by comparing their view with the views presented to their peers. The most prominent example of this would be the System Operator impersonating a Client in a Computation Task with the goal of learning Restricted Data. This manner of impersonation can be circumvented if the Client validates the identities using communication channels not under the control of the System Operator – i.e. out-of-band – before approving.

SYS-2.1 ***Requirement*** *(Trust)*
The System shall enable Users to verify identities of other Members contained in the Computation Task Specification locally. The verification shall not rely on trust in the System Operator.

The approval of a Computation Task Specification in the Control Plane implies the Client's agreement to deploy a Computation Task under the exact conditions contained within. For this reason

it is important that the mechanism of approval guarantees integrity and non-repudiation. It is reasonable to assume that Clients express their approval by digitally signing the Computation Task Specification.

> **SYS-2.2** ***Requirement*** *(Trust)*
> *The System shall use Computation Task Specification approval mechanism that provides integrity and non-repudiation.*

As the Computation Task is deployed into the MPC infrastructure, each action with it (e.g. uploading of Protected Input Data) should be preceded by an integrity verification: a check ensuring that each Computing Node is actively following the Computation Task Specification approved by the Clients. Given that the Computation Task execution is collaboratively handled by several Computing Parties, it would suffice if all Computing Nodes report the hash of the active Computation Task Specification that matches the one signed by the Clients.

> **SYS-2.3** ***Requirement*** *(Trust)*
> *The System shall allow Input and Output parties to verify the integrity of their Computation Task Agreements deployed in Computing Nodes directly.*

3.4 Analysis of potential attacks

The list of attacker profiles that must be considered when designing the security measures of the System include:

a) an active attacker against the Computing Parties, Input Parties, Output Parties and System Operator;

b) a compromised or malicious Output Party, Input Party or Computing Party;

c) a compromised or malicious cloud provider;

d) compromised or malicious internet infrastructure;

e) a malicious author of the Computation Task Specification;

f) a compromised or malicious System Operator; or

g) a colluding clique of multiple Members of the System together with external attackers.

In this section, four attacks against the JOCONDE System are illustrated, showing how the proposed security measures provide protection against the attacks.

Attack Against Computing Parties. In this scenario, let's assume that an attacker has already obtained login credentials to the Computing Party's device that is hosting the JOCONDE System. The attacker's goal is to obtain Input Data, the Computation Task output or any Interim Data produced by the Computation Task. The Computation Task is executed by the MPC software inside of a trusted execution environment (TEE). Storage and communication are protected by the TEE's secret key and computation takes place within the TEE. So the attacker needs to retrieve the TEE's secret key or tamper with the software executed inside the TEE.

To tamper with the software within the TEE, or switch it out with software containing a backdoor, an attack against TEE attestation is needed. Using attestation, Members of the System verify that the hash of the code executing within the TEE matches with the hash of the expected code. To attack the attestation, the attacker must either find a vulnerability within the attestation process itself or spoof every Member who performs attestation. The spoofing entails delivering tampered

MPC software to every attester such that they do not detect the tampering and will compare the software within the TEE to the already tampered software delivered to them.

The other attack vector is to retrieve the TEE's secret key, which requires exploiting a vulnerability in the TEE hardware.

Since the software executing inside the TEE is the MPC software that computes on Protected Data, exploiting a single TEE is not sufficient to obtain the private data. Security properties of the chosen MPC software dictate how many TEEs at respective Computing Parties the attacker has to breach in order to break confidentiality of private data. This task is more difficult when the Computing Parties use a variety of different TEE hardware.

Attack Against an Input Party. Suppose an attacker targets one of the Input Parties. Looking from the outside, the Input Party sends their Protected Input Data to the Computing Parties through encrypted secure tunnels. An attacker seeing only this communication between the Input Party and Computing Parties would have to break the encryption of all three secure tunnels to obtain the Input Data.

Suppose an attacker exploits another vulnerability to gain administrator access to the Input Party. Since the data inputted to the System already exists within the premises of the Input Parties, the Party's use of the JOCONDE System makes no difference to such an attack, and the security goal stated in Section 3.1 would be still met

Side-channel Attack Using Physical Access to a Computing Party. An attacker with physical access to a Computing Party's hosting device has no meaningful advantage compared to an attacker with remote administrator access.

An attacker with physical access to the TEE hardware can attempt to learn about the data inside the TEE by measuring the TEE chip's power draw, thermal profile, memory access pattern, cache misses, or other side-channels. The JOCONDE System is better protected against side-channel attacks than systems where TEEs are used without MPC. Most side-channel attacks against TEEs require running the code within the TEE multiple times or interrupting its execution between every code instruction. Since the program executing within the TEE is a multi-party computation protocol, there are other parties waiting for messages from the party under attack. Thus, interruptions or re-running of the protocol code would be detected by the other Computing Parties, meaning that the attacker's presence is also detected.

Compromised or malicious System Operator. A compromised System Operator can launch attacks on either the Management Plane or the Control Plane. On the Management Plane, it can impersonate one or more Parties and sign a Computation Task Specification on their behalf. On the Control Plane, a compromised System Operator can alter the view of Computation Task Specification contents or its state for a Party, for example, to deceive one into signing something that they do not want to.

Both of these attacks are mitigated by the fact that, by design, none of the System's central components are unconditionally trusted by System Members. Parties are enabled and encouraged to verify critical steps offline: Members' identities can be verified out-of-band and Computation Task Specifications are expected to be double checked and signed on premises.

4 System description

4.1 High-level overview

The JOCONDE System uses MPC to preserve data confidentiality throughout the lifecycle of a statistical Computation Task. The architecture is based on three planes that represent groups of logically connected functions: the Data Plane, the Control Plane, and the Management Plane. These planes operate jointly to ensure that data can be securely processed without compromising privacy.

- The Data Plane includes all data operations within a given Computation Task, including Input Data preparation, protecting Input Data for upload, and execution of the Computation itself, utilising cryptographic techniques like MPC and TEEs to maintain confidentiality.
- The Control Plane deals with the creation and lifecycle management of Computation Tasks. The functions in this plane ensure that all involved Parties agree on the Computation Task Specification before any Computation is executed.
- The Management Plane gathers all functions that are independent from specific Computation Tasks. It handles System maintenance, auditing, and Client onboarding, ensuring that the System functions smoothly and remains secure.

With this approach the System aligns with the vision outlined in Chapter 2 of providing an on-demand, privacy-preserving computation service aimed to be as easy to use as possible, while providing strong security guarantees to meet the data protection requirements. The use of TEE technology enhances security by providing isolated environments for Computation.

In parallel to the planes, which describe a logical composition of the System, there are some technical components foreseen as vital in the System – which, in this document are aptly named as Management Plane Subsystems and Control Plane Subsystems, hosted and managed by the System Operator. The Management Plane Subsystems enable the centralised management of Members. As part of the System Operator's Management Plane activities, it is responsible for onboarding and offboarding Members. With it, the System Operator maintains the catalogue of Members that are allowed to interface with the System.

The Control Plane Subsystems serve to coordinate and oversee the admission of Computation Tasks. By Requirements BUS-3.2 and BUS-3.3, the System Operator must have an overview of, and the ability to prevent Computation Tasks running in the System. Moreover, the Control Plane Subsystems make sure that Computation Tasks adhere to System Agreement before they could be deployed to the MPC infrastructure. It is expected that Computing Nodes utilise machine-to-machine interfaces with the System Operator's infrastructure, providing admission control and telemetry for Computation Tasks.

A standout component of the Control Plane Subsystems is the Client Portal, offered by the System Operator. The Client Portal is a web interface, broadly accessible over the network, for interacting with the Control Plane. The goal of the Client Portal is to simplify the consolidation, agreement, and deployment of Computation Tasks for Clients.

> **SYS-3.1** ***Requirement*** *(Using the System)*
> *Preparation, configuration, and execution of a Computing Task in the System should be as simple and lightweight as possible for the Clients and should involve only minimal marginal costs.*

Any subsystems operated by the System Operator are however clearly and purposefully separated from the Data Plane. The Data Plane will require separate software tools to securely transport Protected Data directly between the Client and the MPC infrastructure. A comprehensive specification of the tools and components will be presented in the architecture deliverables of the Project, starting with *D4.1 System Specification and Architecture (first version)*.

The roles and responsibilities in the core workflow are as follows:

- An Output Party conceptualises and proposes a Computation Task.
- The System Operator provides the service (Client Portal and Control Plane Subsystems) for approving the Computation Task Specifications and deploying the Computation Tasks in the System.
- Input Parties prepare and validate the Input Data and apply protection to Input Data before distributing it among Computing Parties.
- Computing Parties validate the Computation Task requests before executing the tasks. If successful, they return the Protected Output Data to the Output Parties. Finally, the Computing Parties securely erase all Protected Data.

An Output Party starts by conceptualising a new Computation Task. Output Parties and Input Parties formalise the task into a detailed Computation Task Specification, communicated outside the System.

The Computation Task Specification, when formalised, is put up for consolidation between all involved Clients (OPs and IPs) via the Client Portal for signing the Computation Task Agreement based on the Computation Task Specification.

After the Computation Task Agreement has all the signatures needed, and has additionally been signed off by the System Operator, IPs prepare their Input Data, and after validation the Protected Input Data is delivered to Computing Parties.

Computing Parties validate the submissions. If validation fails, the task is rejected and subject to corrections by the Input Parties. When Input Data is accepted, the Computing Parties execute the given tasks.

Upon completion, the Output Parties retrieve the Protected Output Data from the Computing Parties and remove the protection locally to obtain the resulting Output Data. See Figure 4 for a high-level overview of the core workflow process and Figure 5 for a concept map of tasks and data flows.

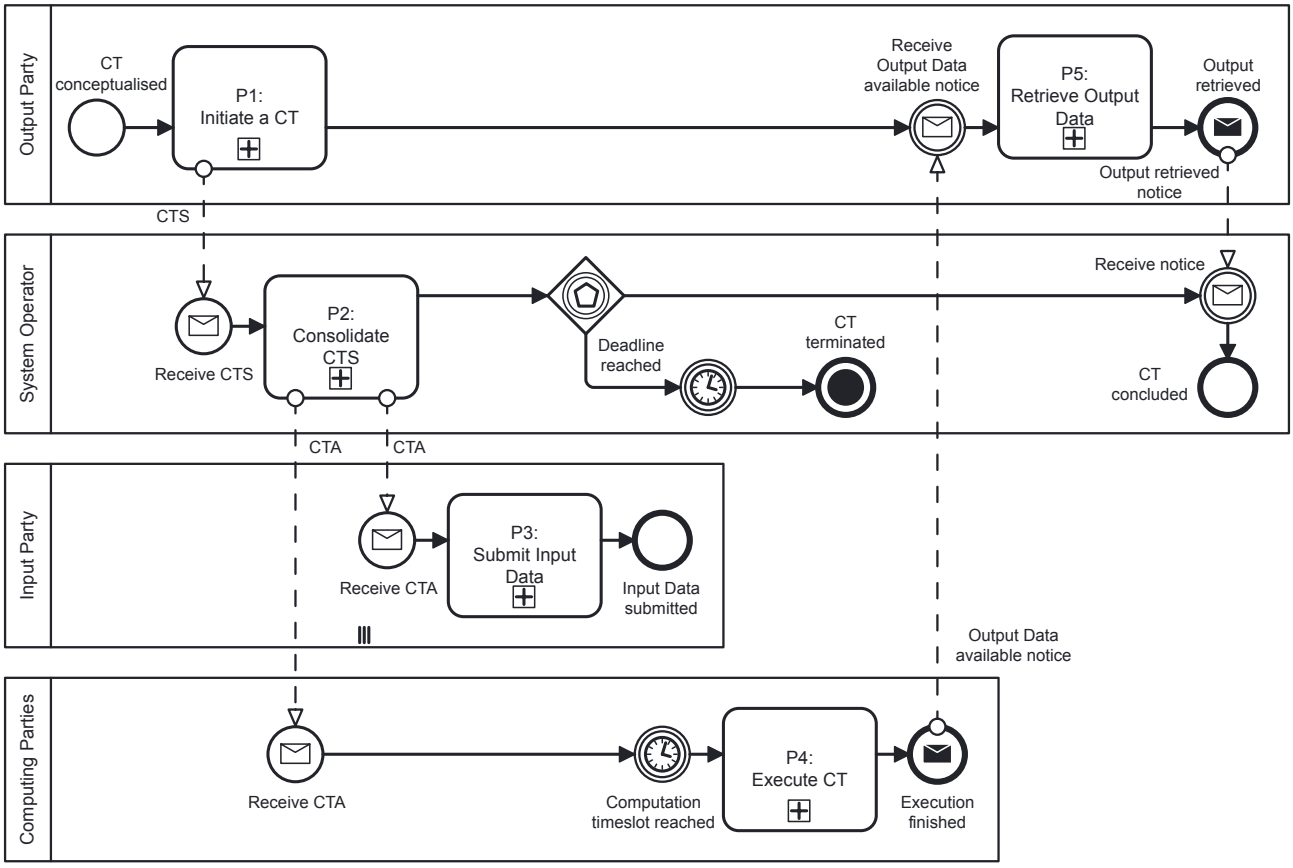

Figure 4. High-level diagram of the core workflow process, illustrating the lifecycle of one Computation Task. An exploded view and explanation of each sub-process (a self contained process marked with a process identifier "Pn") are given throughout Sections 4.4.1 to 4.4.4. The secure deletion of Protected Data is part of sub-processes P4 and P5. The process assumes a simple distribution of Client roles: there is a single Output Party, who is disjoint from the set of Input Parties. On this and subsequent sub-process diagrams, the pool *Computing Parties* expresses the set of Computing Parties of a Computation Task as one abstract stakeholder. This should be interpreted as the Computing Parties operating in unison with respect to each other, i.e. as if it were one machine. Abbreviations used: Computation Task (CT); Computation Task Specification (CTS); Computation Task Agreement (CTA).

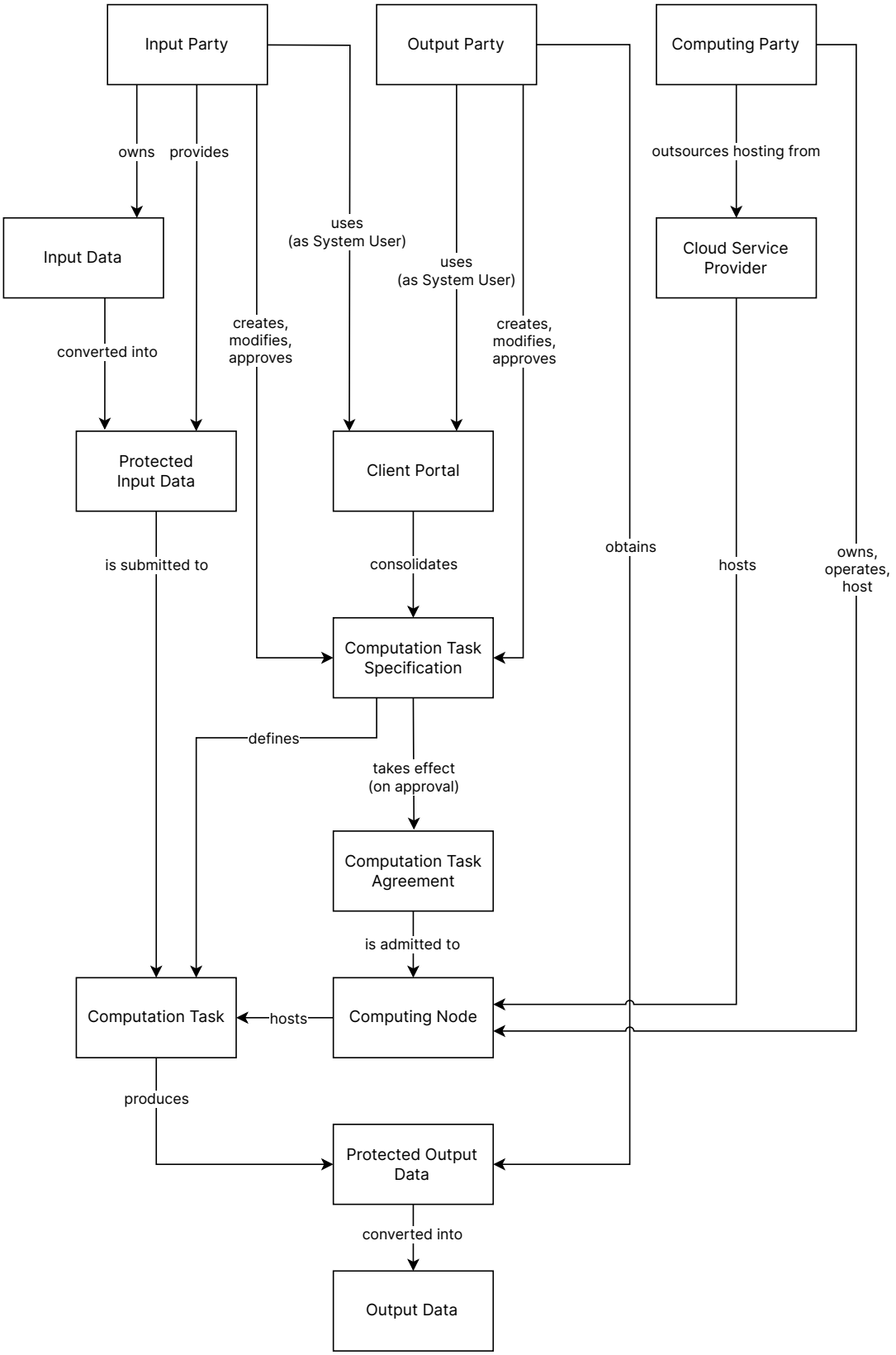

Figure 5. Concept map of tasks and data flows, illustrating how Parties interact with different components, states of the Computation Task, and data in the System.

4.2 System setup

Identity and Access Management (IAM) is a mandatory Management Plane subsystem. The System Operator is responsible for maintaining a robust IAM framework for authorisation and authentication of Members in the System. Credentials used to interface with the System are provisioned to Members during the onboarding process carried out by the System Operator. This ensures that external actors who are not authorised by the System Operator are not able to harness the System's resources.

> **SYS-4.1** ***Requirement*** *(Managing the System)*
> *The System shall provide the System Operator with Identity and Access Management (IAM) for Member management and access.*

Care must be taken to secure the Client Portal, and Management and Control Plane Subsystems against the common threats regarding online services – while the aforementioned components never come into contact with Restricted Data, they do handle documents and operational information that can be confidential. It is vital that the Client Portal is highly available; as it colloquially is the front-end for the whole JOCONDE System, it can aid in communicating and repairing Data Plane outages (which are foreseen as more likely) whilst Clients would still have access to most of the System functionality necessary for their work.

The System must involve and onboard prospective Computing Parties during the initial setup. The notion of a *default set of Computing Parties* ensures the System capability to serve Computation Tasks on-demand in a "one size fits all" configuration. Selection of Computing Parties for the default set must follow careful consideration with respect to their mutual independence (BUS-4.7), interests and motivation to participate, technological capability to maintain infrastructure, and most critically, their compatibility with the legal setting and perceived public trust. Following these criteria, the System Operator shall assess the fitness of a set of candidates to fulfil most Clients' needs, hence supporting the largest amount of Computing Tasks without necessarily onboarding new Computing Parties. From a system security perspective, nothing prevents the stakeholder in charge of acting as System Operator to act also as Computing Party. There needs to be at least three Computing Parties in the default set to successfully employ the MPC paradigm (BUS-4.6).

> **SYS-5.1** ***Requirement*** *(System)*
> *The distributed secure multi-party computing infrastructure of the System shall consist of at least three distinct Computing Nodes.*

The System setup is completed when at least three Computing Parties are onboarded, with their Computing Nodes fully operational and interfaced with the Control Plane Subsystems, ready for the deployment of incoming Computation Tasks.

4.3 Member onboarding

System Members are introduced to the System through standardised onboarding procedures. Well-defined and rigorous onboarding procedures are key to mitigating risks and privacy concerns. Onboarding is a Management Plane activity arranged by the System Operator. The process involves direct interaction between the prospective Member and the System Operator; it can be initiated by either entity.

The onboarding process must ensure that all Members are securely integrated into the System and shall guarantee compliance with the System's security, technical, and legal standards. The

System allows Members to take on multiple roles such as IP, OP, or CP at the same time, depending on their verified capabilities. The onboarding process combines offline administrative actions with online authentication processes for identification. If possible, it is advisable for the System to reuse already existing, state-backed credentials (e.g. those based on cryptographic key pairs and public key infrastructure). In particular, alignment with the European Digital Identity Regulation No 910/2014[1] would allow both identity verification and authentication.

The specifics of the onboarding process are dictated by whether a prospective Member is onboarded as a Computing Party or a Client. Nevertheless, both of these roles share the following onboarding steps:

1. **Identity verification:** Member candidates shall comply with any request from the System Operator to verify their true identity and affiliation. The System Operator should only proceed with the onboarding after validation of these claims. At this stage the System Operator may reject the candidacy if the criteria for becoming a Member are not met.
2. **Signing the System Agreement:** Member admission is formalised by signing the legal contract(s) part of the System Agreement. An important part of the System Agreement is a list of rules, which outlines the System Members' responsibilities and the conditions under which they can interact with the System. As the bare minimum, it contains the following:
   - The JOCONDE System is for ESS members. This means that at least one Client in every Computation Task must be an NSI or Eurostat. ONAs may replace NSIs in this role, but only with the NSI's approval. This condition is verified by the System Operator during the onboarding phase, likely through administrative procedures.
   - Each Member remains responsible for complying with all national and EU laws that apply to its membership in and use of the System. For example, the prospective Client may have to consult their national regulatory or supervisory authority or conduct a fully-fledged DPIA preliminarily to agreeing on the Computation Task. The technical and legal documentation about the System developed in the Project will support the prospective Client in performing such duties. By signing a Computation Task Specification, the Clients who are Parties to that specific Computation Task assume responsibility that the Computation Task Specification is compliant with all applicable laws.
   - Clients must accept that the analysis code from the Computation Task Specification will be made available to the public. This transparency rule is in line with the European Statistics Code of Practice[2], improves public acceptance and contributes to increase security by allowing independent scrutiny by external experts.
   - Clients must accept that upon termination of the Computation Task all Interim Data, Protected Input and Protected Output Data will be securely erased or made permanently illegible (see requirements BUS-4.9 and BUS-4.10).
   - Clients accept that logs produced by a Computation Task (Computation Task Logs) will be stored permanently and made available to all participating Parties (see requirement SYS-5.3) and the System Auditor.
   - System Logs that transcend any specific Computation Task will be made available to the System Auditor and other relevant auditing and reviewing authorities.

   A more detailed list of such rules is part of the System Agreement draft that will be provided as part of deliverable *D3.2: Draft of reference DPIA and model agreements*. All Members –

[1] eIDAS Regulation. https://digital-strategy.ec.europa.eu/en/policies/eidas-regulation (30.10.2024)

[2] European Statistics Code of Practice – 2017 edition. https://ec.europa.eu/eurostat/web/quality/european-quality-standards/european-statistics-code-of-practice

whether CP, IP, or OP – must adhere to the System Agreement. Candidate Members must acknowledge the presented terms and either accept or decline. Upon acceptance, the candidate Member's onboarding process will continue; upon refusal, the process is terminated. Both candidate Members and, after successful onboarding, full Members must provide evidence and supporting documents that the mandatory provisions contained in the System Agreement are fulfilled. The System Operator is responsible for ensuring that all Members remain compliant with these terms throughout their participation in the System by means of conducting regular audits and compliance checks.

Subsequent steps in the onboarding process depend on the target role, varying in terms of technical prerequisites and compliance requirements.

4.3.1 Computing Party onboarding

Computing Parties play a pivotal role in the execution of Computation Tasks, and their onboarding process is designed to ensure that they can provide secure, isolated environments for Computations. The onboarding process includes:

1. **Infrastructure Setup:** Computing Parties must deploy the necessary infrastructure with properly configured hardware and software security measures. Computing Parties, aided by documentation and guidance from the System Operator, provision the necessary resources and integrate their Computing Node to the System. Computing Parties are free to choose whether to host their Computing Node on premise or outsource its hosting to a cloud service provider (CSP) as long as they remain in control of the Computing Node and the independence requirement (see BUS-4.7) of Computing Parties is fulfilled.
2. **Compliance Verification:** Computing Parties must comply with security standards, which are asserted by the System Operator and periodically audited by the System Auditor. This includes ensuring that Computing Parties are independent from each other to prevent collusion, as well as verifying that the deployed infrastructure meets all technical requirements.

By ensuring that Computing Parties are properly onboarded, the System guarantees that Computation Tasks are executed securely, in conformance with the MPC paradigm.

4.3.2 Client onboarding

The onboarding of a new Client is initiated by the applicant. Clients can initiate or participate in new Computation Tasks as Input Parties and Output Parties only after having successfully completed the onboarding process. The onboarding process involves the following steps in addition to the general steps outlined in Section 4.3:

1. **Account creation:** The System Operator creates an account for the Client and either provides the Client with respective credentials or registers already existing credentials (e.g., cryptographic key pair) with the System. The account created for the Client is meant strictly for using the Client Portal – or more generally, improve the user experience of interacting with the System. Only those credentials, which are not managed by the System Operator or any other single point of trust, shall be utilised for interacting with the Data Plane.
2. **Provisioning Tools:** Some subsystems may require special software to use. This is highly likely to be the case for interactions with the Data Plane, which is expected of the Client throughout the Computation Task lifecycle. The System Operator's role is to prepare new Clients, providing or guiding them to the necessary software and instructions.

SYS-3.2 ***Requirement*** *(Using the System)*
Technical requirements for Clients of the System in the role of Input Party and/or Output Party should be minimal – without the need to install specialised hardware.

This up-front onboarding process aligns with the vision of maintaining strict compliance assurances (BUS-2.7) while minimising the complexity of the core workflow (BUS-1.4). Already onboarded Clients can jump straight to planning new Computation Tasks as described in Section 4.4.1.

4.4 Computation Task lifecycle

A Computation Task is the central element in the core workflow. While conceptually the Computation Task is a transaction that turns Input Data into Output Data, in a technical sense it is a finite state machine residing in the Computing Nodes. This abstraction, illustrated in Figure 6, is convenient for reasoning about the Clients' interactions with the System while at the same time fulfilling certain criteria with respect to data lifecycle policies (e.g. requirements BUS-4.9 and BUS-4.10). In other words, the Computation Task is the link between the Control and Data Planes ensuring that all rules – Computation Task Specification and System Agreement – are followed exactly as they were defined and agreed on beforehand.

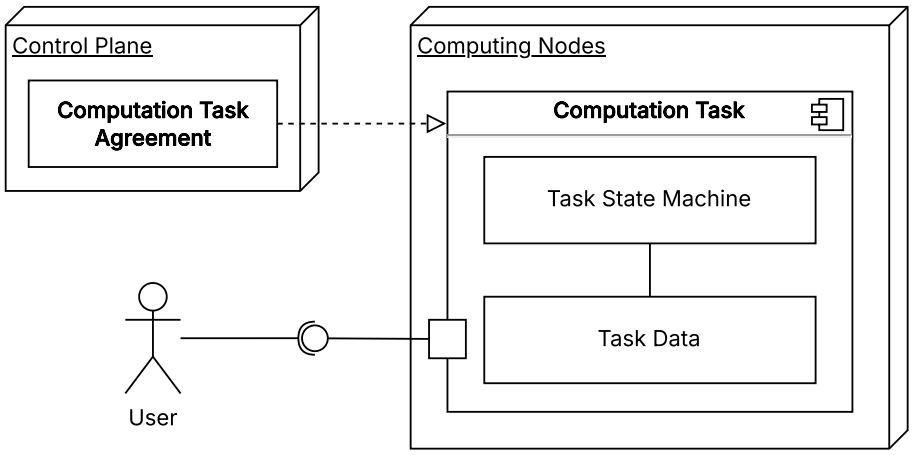

Figure 6. The Computation Task is deployed through admission of the Computation Task Agreement into the distributed MPC infrastructure

SYS-6.1 ***Requirement*** *(Task and data lifecycle)*
The System shall associate all data with its respective Computation Task to enforce data lifecycle policies.

In order to streamline collaboration and ensure timely Computation Task execution, data lifecycle deadlines are enforced. These deadlines are critical to avoid delays or discrepancies between Members:

- **Computation Task Specification signing deadline:** Input and Output Parties must sign the Computation Task Specification within a predefined time-frame since its submission in the Client Portal. Failure to meet this deadline results in the cancellation of the Computation Task Specification, annulling all existing signatures, and requiring the Computation Task Specification to be reinstated. This is to motivate agile collaboration and avoid lingering Computation Task Specifications.

- **Input Data preparation deadline:** Input Parties are required to prepare and submit data within a set time-frame after the Computation Task Agreement is concluded. This ensures that all the required Input Data is available for a timely execution of the Computation Task.
- **Computation timeslot:** A specific timeslot is allocated for the execution of the Computation Task. All Parties must be ready by this point, ensuring that the Computing resources are optimally utilised or available at that time.
- **Data retention:** Any Protected Output Data that is not retrieved by the given deadline will be automatically erased along with the Computation Task to prevent retention of Restricted Data beyond its necessity.

> **SYS-6.2** ***Requirement*** *(Task and data lifecycle)*
> *The Computation Task shall expect time deadlines for task lifecycle stages, specifically manual I/O operations including Protected Input Data upload and Protected Output Data retention.*

As established, a Computation Task implements the terms set in a well-defined Computation Task Specification contained within the Computation Task Agreement. Hence before a Computation Task can exist, Clients must engage with the System and among each other to define the Computation Task Specification. The processes impelled by the actions of Clients starting from the conception of a Computation Task Specification and ending with receiving the outputs constitute the Computation Task lifecycle. The following section presents a detailed description of the lifecycle stages which were briefly outlined as the core workflow in Section 4.1.

4.4.1 Initiation and consolidation of Computation Tasks

The main goal of the System is to allow Clients to collaboratively analyse data under MPC with minimal effort. It begins with an Output Party having a need or idea for a new Computation Task that requires input from several Clients. In fact, any Client can propose a new Computation Task, but we see that in most cases it is the Output Party as they are the one interested in the results of the analysis.

The System provides a central component – the Client Portal – to specify and consolidate such proposals as Computation Task Specifications and invite other participants to cooperate throughout the Computation Task lifecycle until analysis result (Output Data) is produced. We however see that in practice this process may actually begin with discussion and several rounds of drafting by the participants outside the System. In the end it is only important that at some point the new Computation Task idea is formalised as a Computation Task Specification in the Client Portal (see Figure 7 for the Computation Task initiation process outline and accompanying Tables 1 and 2 describing the process steps and data elements).

Table 1. P1: Process steps

ID	Name	Roles	Description	Trigger	Goal
P1.1	Draft a Computation Task Specification	OP	An OP defines the details of the Computation Task by drafting a Computation Task Specification	The OP has conceptualised a Computation Task	Create a Computation Task Specification that compliant with the System

Table 1. P1: Process steps *(continued)*

ID	Name	Roles	Description	Trigger	Goal
P1.2	Submit the Computation Task Specification	OP	The OP submits the Computation Task Specification to the System Operator via the Client Portal	A Computation Task Specification exists	The System Operator can use the Computation Task Specification in the subsequent consolidation process.

Table 2. P1: Introduced data elements

ID	Name	Holder	Description
E1.1	Computation Task Specification	OP, SO	A Computation Task Specification consolidating both human- and machine-readable artefacts: • chosen MPC technology; • data quality assurance algorithms; • data analysis algorithm; • Input Data specification; • Client-to-role assignments; • Members' identities; • assignment of Computing Parties; • legal contracts; • Computation Task lifecycle deadlines.

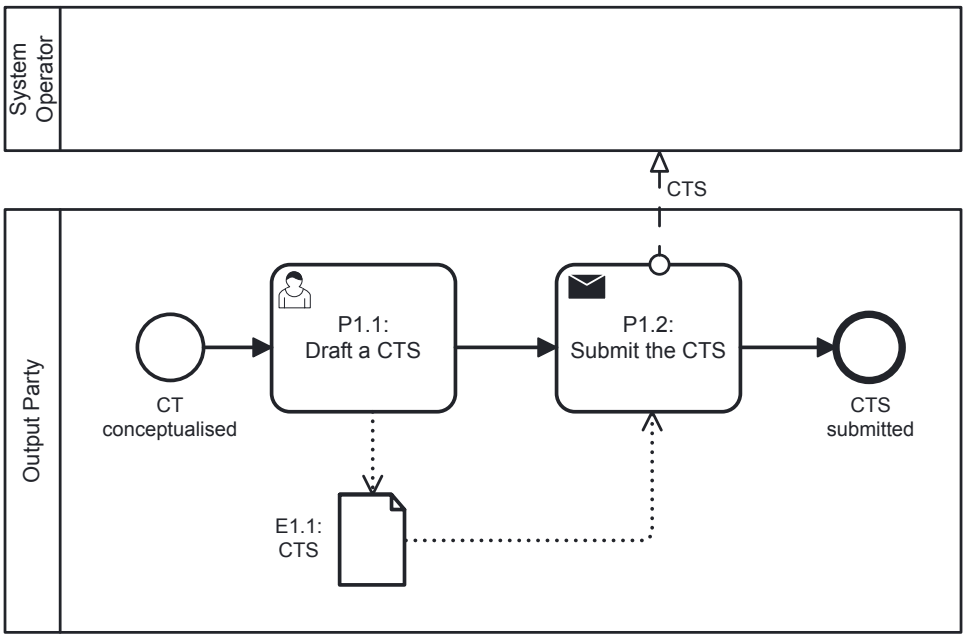

Figure 7. Sub-process P1: Initiate a Computation Task (CT). One Output Party drafts and submits a Computation Task Specification (CTS) based on their concept of the Computation Task.

The Computation Task Specification exhaustively specifies all details concerning a Computation Task. Technically, the Computation Task Specification is a structure that consolidates both human- and machine-readable artefacts. Owing to requirement SYS-3.1, A simple and intuitive

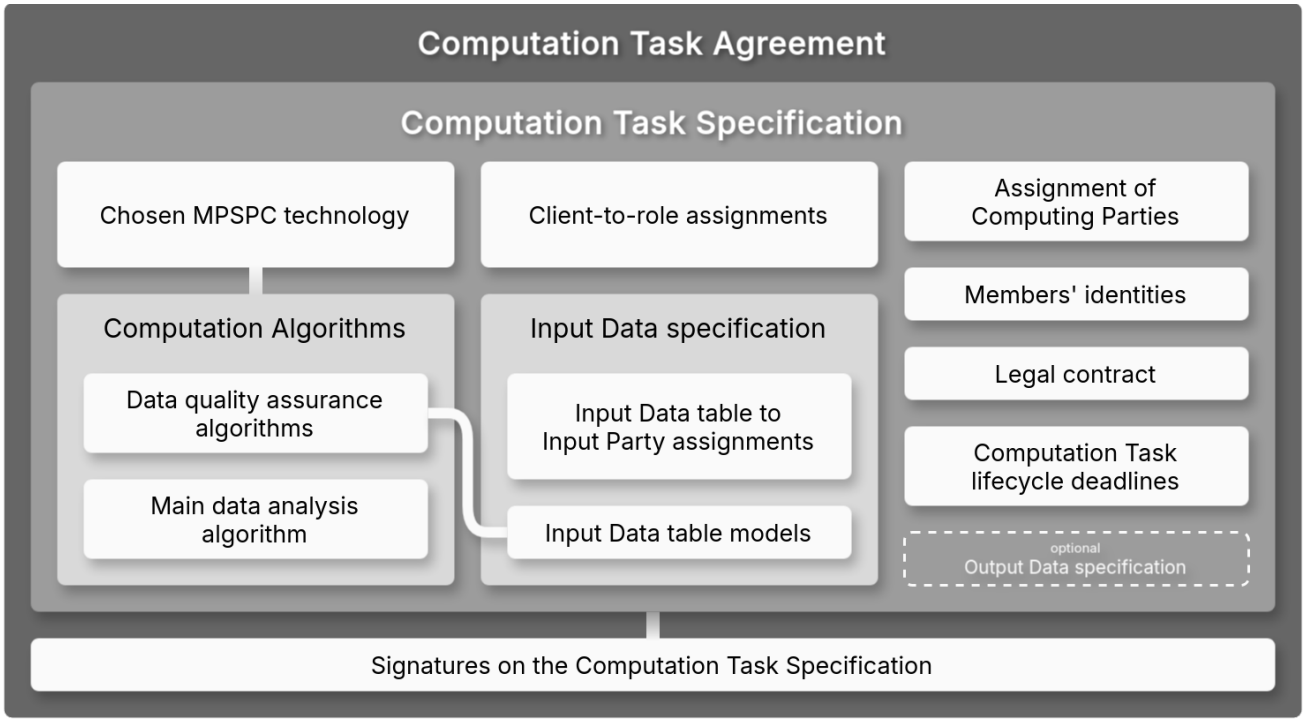

Figure 8. Taxonomy of the Computation Task Agreement.

user interface aids Clients in the creation of a valid Computation Task Specification. In this section we expand on the functional role of the Computation Task Specification, its elements, and the Client's interactions with it; technical details (e.g. with respect to the encoding of all contained items) will be specified in deliverable *D4.1 System Specification and Architecture (first version)*. An overview of items comprising a Computation Task Specification can be found in Figure 8. A detailed description of each item follows.

Definition of the data analysis algorithm delineating the function(s) applied to the Input Data to produce Output Data. The definition details the exact algorithm to be executed by the Computation Task. It has to be both machine- and human-readable, the former to facilitate its automatic translation into a format that is executable by the MPC technology and the latter to facilitate reviewing the definition for correctness and potential data leaks. Therefore, this document should consist of computer source code written in a programming language that is easy for analysts to follow and review.

One such programming language is SecreC[3] [6, 7] which enforces the separation of public and private (protected) data flows. Moving data from the protected to the public domain can only be carried out by a single function and the same is true for Computation output (Protected Output Data). Therefore, it is easy for reviewers to search for code segments where a possible data leak might occur. SecreC comes with a substantial standard library[4] that allows for quick privacy-preserving application development, also by non-cryptographers. Furthermore, the language and its standard library are open-source and protocol polymorphic, i.e. independent of the actual MPC Engine implementation, following the System design requirement to avoid vendor lock-in.

[3] The SecreC language. https://github.com/sharemind-sdk/secrec
[4] SecreC Standard Library. https://github.com/sharemind-sdk/secrec-stdlib

However, requiring the use of a special programming language to specify analysis code for the Computation Task Specification introduces a new challenge for the involved Clients. Output Parties have to learn a new language to express their algorithms and Input Parties (or their Contractors) have to learn the language to review the proposed analysis code. To alleviate this and be more appealing for new Clients, the System may also support specifying analysis code in a more domain-specific language such as SQL (e.g. see SECRECY [8], Senate [9], SCQL [10]) or R (e.g. see Rmind[5] [11]). This also makes it easier to port existing analysis code to be compatible with the JOCONDE System. Such alternative languages can be either supported directly by the Computation Task Specification or there can be tools to transcode programs written in such languages into the Computation Task Specification language of choice. In the latter case, the transcoding have to be verifiable.

As specified in the Procurement[6] the System must support "all combinations of elementary private set operations (private set union, private set intersection, private set difference) along with simple computations on the items of the resulting set (e.g., counting of items fulfilling some arbitrary equality or inequality conditions on the variable values, aggregation of variable values, computation of binned histograms of variables values, linear regression between variables)." The support for exact record matching based on common keys (that can also be concatenations of variables' values) is also required, while probabilistic record matching is deemed highly desirable.

SYS-7.1 ***Requirement*** *(Computational capabilities)*
The operations on private Input Data shall include all combinations of elementary private set operations (private set union, private set intersection, private set difference) along with simple computations on the items of the resulting set.

SYS-7.2 ***Requirement*** *(Computational capabilities)*
The System shall support exact record-matching based on common keys (identifiers or concatenation of variables values).

SYS-7.3 ***Requirement*** *(Computational capabilities)*
The System should support additional operations (e.g. probabilistic record-matching).

List of Clients, assigned roles, and identities clearly defines *who does what* in the scope of the Computation Task. The list contains only those Clients who are participating in the Computation Task in Input or Output Party roles, giving Clients a clear overview of the participants involved. The configuration of roles must be flexible to accommodate a wide range of envisioned scenarios: an Output Party who may also want to provide Input Data takes both roles; one Computation Task may have several parties interested in the result, requiring multiple Output Parties.

SYS-3.3 ***Requirement*** *(Using the System)*
The System shall allow for flexible configuration of Computing Tasks serving different Clients, i.e. in the role of Input Party, Output Party or both at the same time.

SYS-8.1 ***Requirement*** *(Task I/O)*
The System shall support Computation Tasks with at least two Input Parties and at least one Output Party.

[5] The Rmind data analysis tool. https://docs.sharemind.cyber.ee/sharemind-mpc/2023.09/development/rmind.html

[6] Tender reference number ESTAT/2023/OP/0004. For further information, please see the TED eTendering website: https://etendering.ted.europa.eu/cft/cft-display.html?cftId=12503

SYS-8.2 ***Requirement*** *(Task I/O)*
The System should support Computation Tasks with more than one Output Party.

Machine-readable or cryptographic Client identities embedded in the Computation Task Specification serve a dual purpose. First, they serve as an aid for Clients to reliably identify their peers. As discussed in Section 3.3, Clients should not rely on the view presented by the Client Portal lest it become a single point of trust. Following requirement SYS-2.1, Clients can establish direct trust over out-of-band (OoB) communication channels (e.g. email), exchange identities and cross-check. Second, it serves as access control within the MPC infrastructure – for example, a Computing Node only authorises a request to retrieve Protected Output Data if the Output Party supplies proof of identity.

Input Data specification is to be specified in minute detail to improve communicability of expected data structures and to avoid runtime errors caused by ill-formatted Input Data. Input Data may only be provided in well-defined tabular formats. Data models in the Computation Task Specification must contain, for each Input Data table, the number of columns and the data type (e.g. integer, floating-point number, string) of each column. The adherence of uploaded data to the specified data model is subject to local and server-side validation, explained in Section 4.4.2.

SYS-8.3 ***Requirement*** *(Task I/O)*
The Computation Task shall support Input Data in tabular format.

SYS-8.4 ***Requirement*** *(Task I/O)*
The Computation Task shall expect a detailed data model for each Input Data table as a part of the Computation Task Specification.

Furthermore, each Input Data table must be uniquely identified and associated with the Input Party who is responsible for providing the table. This is necessary for Computing Nodes to authorise uploading of Input Data, ensuring that no other Client could submit a table on behalf of the intended Input Party. An Input Party should not be limited to providing a single Input Data table as multifaceted Computation Tasks may require more intricate aggregations across heterogeneous tables.

SYS-8.5 ***Requirement*** *(Task I/O)*
The Computation Task should support multiple Input Data tables per input party.

Output Data specification is optional. It is not necessary for a minimal solution where Output Data constitutes a single set of pre-agreed values accessible to all Output Parties. In such case the Output Data is not explicitly specified in the Computation Task Specification but is derived from the analysis program code. The code must clearly indicate the values which are to be revealed to the Output Party[7]. Output Data comprises one or more instances of scalar or vector values.

SYS-8.6 ***Requirement*** *(Task I/O)*
The Computation Task shall support Output Data values in scalar and vector formats.

However, it is possible that some prospective use cases may require assigning specific subsets of Output Data to only be revealed to certain Output Parties within a single Computation Task. For

[7] For example, the SecreC language provides a simple yet flexible *publish* command for this.

such use cases, Computation Task Specification must contain an explicit declaration of Output Data specification, reminiscent of the Input Data specification. The Output Data specification must encode the association between the Output Party and the computed Output Data value; the MPC infrastructure must technically enforce the associations as part of access control.

> **SYS-8.7** ***Requirement*** *(Task I/O)*
> *The Computation Task shall support multiple Output Data values.*

> **SYS-8.8** ***Requirement*** *(Task I/O)*
> *The Computation Task should allow configuring individual Output Data values to only be retrieved by specific Output Parties.*

Data quality assurance algorithms are optional items that impose further user-defined restrictions on uploaded Protected Input Data, in addition (and extension) to data models, for more involved tests. The algorithms, if present, are specified in similar fashion to the main data analysis algorithm, i.e. the same programming language. This is because the data quality assurance step is itself an invocation of MPC on the Protected Input Data of one Input Party. It does not however extract or return any Output Data, but contains only the necessary assertions to verify the fitness of the data ahead of the Computation Task execution. Data quality assurance is further described in Section 4.4.2.

List of Computing Parties assigns the Computation Task to a specific set of Computing Parties. As with the list of Clients, this item should provide meaningful and reliable information for the Input and Output Parties to gain assurance w.r.t the identities of the Computing Parties. That is, Clients are able to cross-check these facts by contacting a Computing Party directly. Additional details such as the IP addresses or fully qualified domain names of the Computing Nodes are essential for the client application to communicate with the MPC infrastructure in the following stages.

In the process of conceptualising the Computation Task, Clients might eliminate the System's default set of Computing Parties as a viable choice. This might be the case for legal restrictions, or simply due to a desire to increase security or trust with a custom selection of Computing Parties. To this end, the requirement BUS-3.5 introduced in Section 2.4.1 states that Clients are able to choose a custom set of Computing Parties for a specific Computation Task. Moreover, Clients are also free to specify themselves as the Computing Parties of the task, given they have previously passed the Computing Party onboarding procedure (Section 4.3.1).

> **SYS-3.4** ***Requirement*** *(Using the System)*
> *The System shall support Input Parties and/or Output Parties taking the role of a Computing Party in Computation Tasks which they are a part of.*

Deadlines for Computation Task lifecycle stages are embedded in the Computation Task Specification to strictly define the allotted timeframes for Client actions. The components of this item are as described in Section 4.4 and provided in a date-time representation.

MPC technology of choice would need to be declared in the Computation Task Specification to indicate to the Clients the security characteristics of the Computation Task, and to configure the Data Plane of the MPC infrastructure accordingly. This item is applicable only if the Sys-

tem supports a modular Data Plane with multiple production-ready MPC technologies that are selectable on per-Computation Task basis.

> **SYS-7.4** ***Requirement*** *(Computational capabilities)*
> *The System should allow Members to choose the MPC technology for each Computation Task and specify the chosen technology in the Computation Task Specification.*

Legal documents concerning the data analysis are incorporated in the Computation Task Specification to tie the contractually binding and technologically enforced terms together into a single package. Legal requirements will be identified in the deliverable *D3.1 Initial Legal Analysis*.

Once the Computation Task Specification is drafted, it is shared with all Clients of the Computation Task for review and approval. See Figure 9 for the Computation Task Specification consolidation process and accompanying Tables 3 and 4 for description of process steps and data elements. This collaborative process ensures that Clients are aligned on the task details before proceeding. Upon approval, i.e. signing by all Clients of the Computation Task, the Computation Task Specification is converted into a Computation Task Agreement (CTA) – specifically as shown in Figure 8. Note that the signatures on the Computation Task Agreement are legally binding, so if the System uses custom identities (see Section 4.3) then CTA is the document that binds these to the real identities of the Parties. CTA formalises the roles, responsibilities, acknowledged residual risks, and legal obligations of all Parties involved, fostering transparency and accountability.

At this stage the Computation Task is deployed into the specified Computing Nodes in an initial state: waiting for Input Data. It is critical that Computing Nodes admit only Computation Task Agreements established within the Client Portal.

> **SYS-5.2** ***Requirement*** *(System)*
> *Computing Nodes shall only accept and enforce Computation Task Agreements sourced by trusted means.*

Table 3. P2: Process steps

ID	Name	Roles	Description	Trigger	Goal
P2.1	Consolidate the Computation Task Specification	SO, Client (IP, OP)	The Computation Task Specification is distributed among Clients and Client signatures are collected; if the signing deadline defined in the Computation Task Specification is reached before receiving all signatures, the Computation Task is terminated	Receipt of a new Computation Task Specification	Inform Clients of a new Computation Task Specification for review and signing; collect the signatures for formalising the Computation Task Agreement

Table 3. P2: Process steps *(continued)*

ID	Name	Roles	Description	Trigger	Goal
P2.2	Review the Computation Task Specification	Client	All Clients review the Computation Task Specification and decide to approve or reject it; rejection by any Client means that the Computation Task Specification will not be consolidated	Notification of a new Computation Task Specification	Give Clients the opportunity to thoroughly examine and reject the proposed Computation Task Specification
P2.3	Sign the Computation Task Specification	Client	If the Client approves, they sign the Computation Task Specification	Decision to approve	The Client gives a legally binding signature signifying their willingness to proceed with the Computation Task under the conditions outlined in the Computation Task Specification
P2.4	Submit the Computation Task Specification signature	Client, SO	The signature is delivered to the System Operator via the Client Portal	Client has signed the Computation Task Specification	Have the Client's signature admitted to the System for finalising the consolidation
P2.5	Sign the Computation Task Specification	SO	If all Clients have submitted their signature, the System Operator gives their own signature; this is a manual process including the act of review and approval by the System Operator if it is required (see requirement BUS-3.2), but otherwise it is an automated activity	Signatures received from all Clients of the Computation Task	Express System Operators approval, making it known that the Computation Task is sanctioned by the System Operator to proceed
P2.6	Formalise Computation Task Agreement	SO	The System Operator formalises the Computation Task Agreement by combining the Computation Task Specification, the received signatures, and their own signature; the Computation Task Agreement is subsequently sent to all Parties	The System Operator has given their signature	Compile the Computation Task Agreement, which could be distributed and deployed as a Computation Task

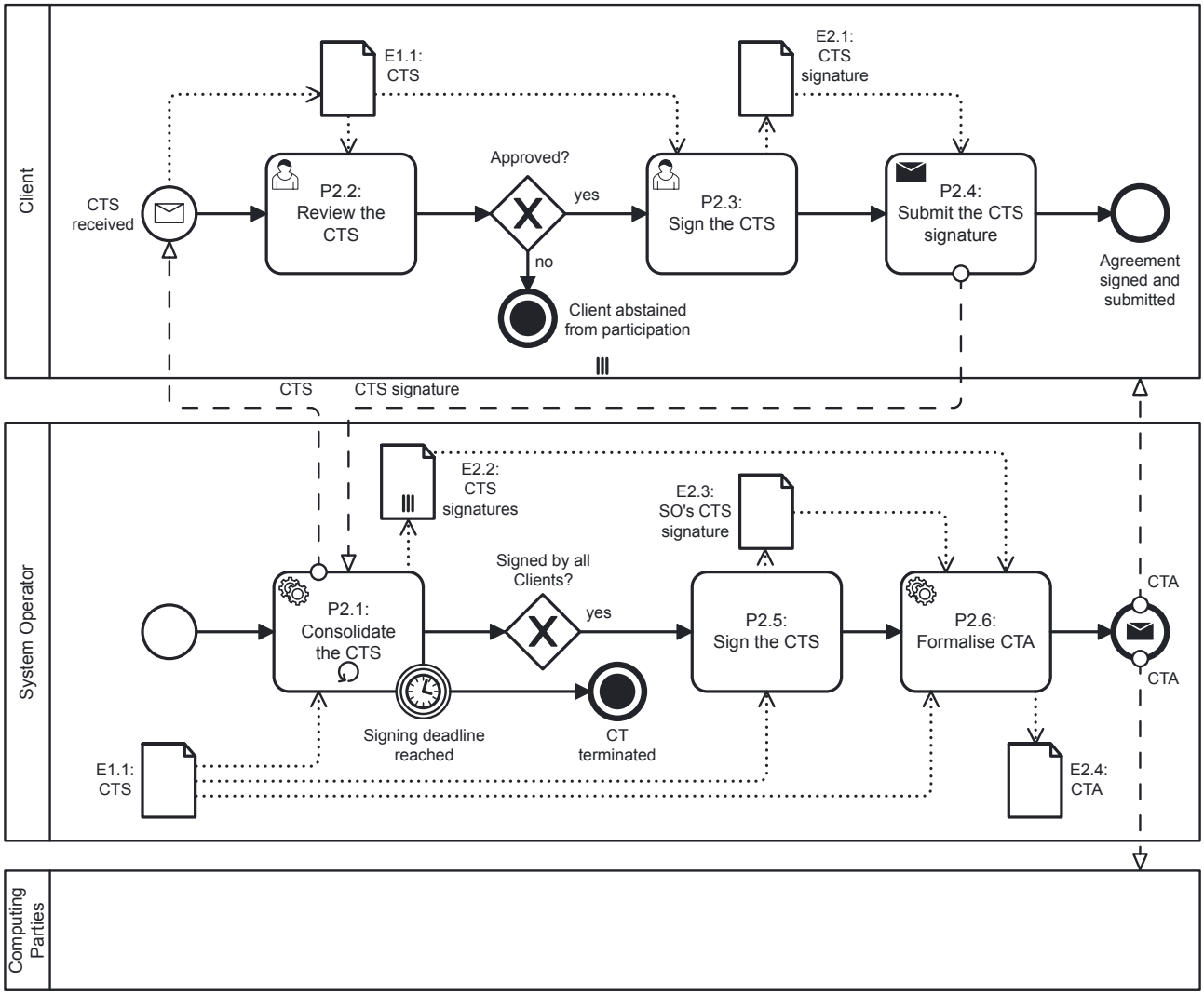

Figure 9. Sub-process P2: Consolidate Computation Task Specification (CTS). The System Operator facilitates the consolidation workflow via the Client Portal. Clients of the Computation Task, comprising the Input and Output Parties, review and sign the Computation Task Specification. A timely receipt of all signatures will result in the Computation Task Agreement (CTA) being formalised and distributed to all Parties of the Computation Task.

Table 4. P2: Introduced data elements

ID	Name	Holder	Description
E2.1	Computation Task Specification signature	Client	Legally binding signature of one Client on the Computation Task Specification
E2.2	Computation Task Specification signatures	SO	Instances of E2.1 from all Clients
E2.3	System Operator's Computation Task Specification signature	SO	System Operator's given signature on the Computation Task Specification

Table 4. P2: Introduced data elements *(continued)*

ID	Name	Holder	Description
E2.4	Computation Task Agreement	SO, Clients, Computing Parties	Document comprising the Computation Task Specification (E1.1), Client signatures (E2.2), System Operator's signature (E2.3)

4.4.2 Input Data preparation and distribution

From this point forward, Clients interact directly with the MPC infrastructure, i.e. the Computing Nodes in charge of the Computation Task, for providing Input Data or retrieving Output Data. Since the following workflow involves orderly and timely actions from multiple Input Parties and Output Parties in an opaque multi-party infrastructure, coordination is the key to fostering a smooth experience. To aid in the coordination of subsequent actions, Clients should have access to up-to-date telemetry about the progress of the Computation Task.

> **SYS-6.3** ***Requirement*** *(Task and data lifecycle)*
> *The System should provide telemetry about the progress of a Computation Task to Input and Output Parties in order to help coordinate their actions (e.g. if the task is waiting for input from a specific Input Party).*

Before the Computation Task can be executed, Input Parties prepare their Input Data locally. This involves extraction of the relevant dataset from their records, application of any necessary transformations for it to conform to the data model set in Computation Task Specification, and validation of its data quality.

Protection is then applied in order to form Protected Input Data using cryptographic techniques. The Protected Input Data is distributed among the Computing Parties, ensuring that no single party has access to the complete dataset.

> **SYS-9.1** ***Requirement*** *(Privacy)*
> *The System shall not disclose Input Data values to anyone other than the Input Party, unless explicitly stated otherwise in the Computation Task Specification.*

> **SYS-9.2** ***Requirement*** *(Privacy)*
> *No single entity shall have the ability to learn any Input Data values of any individual computation task, unless explicitly stated in the Computation Task Specification.*

An Input Party is provided with a software component that applies protection to Input Data and handles the distribution for Protected Input Data. It is also possible that the software component interfaces directly with the Input Party's existing database system, eliminating the necessity for the manual data export step mentioned above.

In addition to local data quality validation, the Computation Task Specification also provides means to optionally include a separate data validation step that is done right after uploading the Protected Input Data to the System. This serves several goals. First, since this validation algorithm is run on Protected Input Data with MPC technology, it brings out possible data formatting and conversion issues that might arise from the exporting and protection of the original dataset. Second, this data validation step, being part of the Computation Task Specification, is agreed upon by all participating Clients. Therefore, it also provides guarantees for the other participants that the Input Party in question cannot provide a maliciously-crafted Input Data that might po-

tentially threaten the confidentiality of other Input Parties' data. The Protected Input Data upload process fails if the server-side data validation step fails.

> **SYS-8.9** ***Requirement*** *(Task I/O)*
> *The Computation Task should support server-side data validation or data quality checks during Protected Input Data upload based on custom data quality assurance algorithms, part of the Computation Task Specification.*

An overview of the Input Data preparation and submission process is shown in Figure 10 with descriptions of steps and data elements given in Tables 5 and 6.

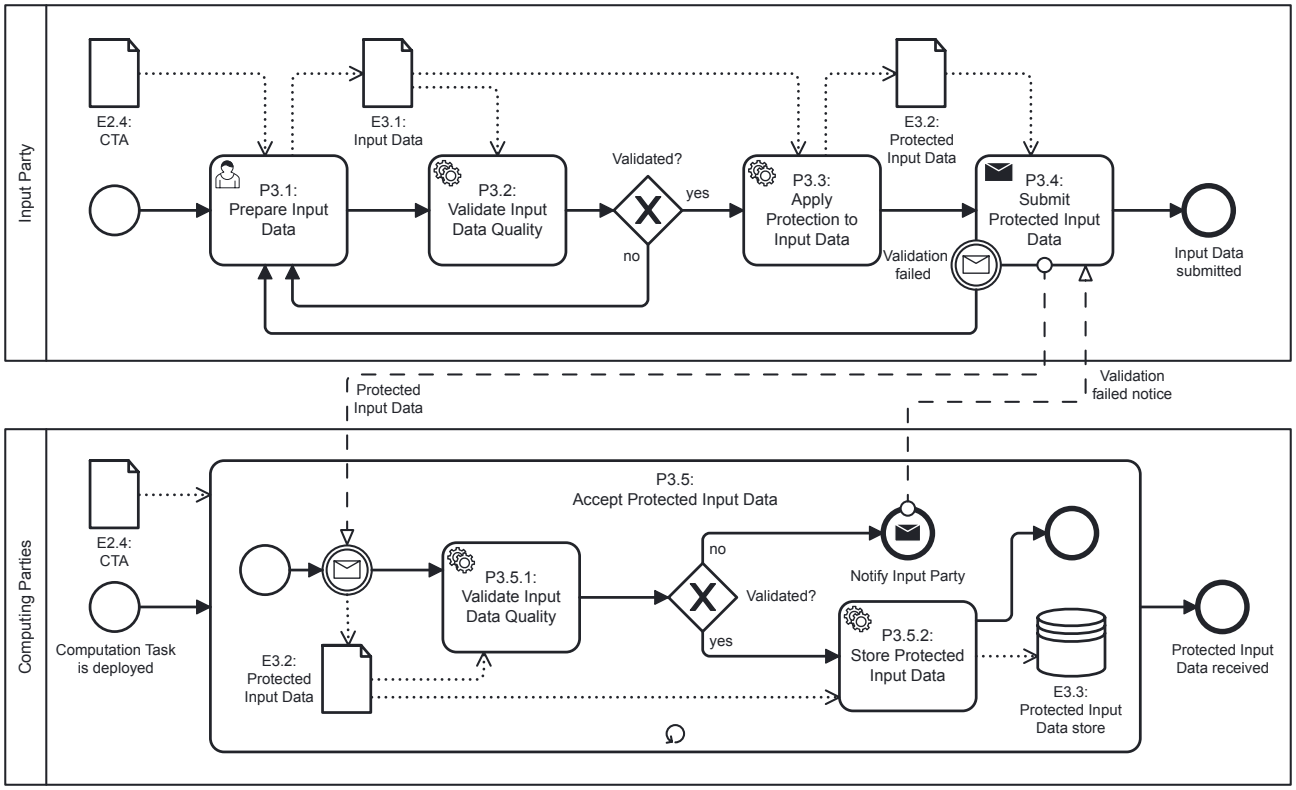

Figure 10. Sub-process P3: Submit Input Data. This sub-process depicts the procedure of submitting Input Data by one Input Party, who received the Computation Task Agreement (CTA) after sub-process P2, as can be seen on Figure 4. As the Computation Task Agreement is also received by Computing Parties, the Computation Task is deployed in the initial state of *waiting for input*.

Table 5. P3: Process steps

ID	Name	Role	Description	Trigger	Goal
P3.1	Prepare Input Data	IP	Prepare (e.g. extract, transform) local data to conform to the Input Data specification in the Computation Task Agreement	Receipt of a concluded Computation Task Agreement, signifying the start of the Computation Task	Obtain the data to be used in the Computation in the expected format

Table 5. P3: Process steps *(continued)*

ID	Name	Role	Description	Trigger	Goal
P3.2	Validate Input Data Quality	IP	Client tools run checks to validate that the Input Data is prepared correctly; if validation fails, the Input Party repeats the preparation process	Input Data is prepared	Ensure adherence to the Input Data specification and data quality criteria locally before submission
P3.3	Apply Protection to Input Data	IP	Protection is applied to the validated Input Data using cryptographic techniques, forming Protected Input Data	Input Data validation succeeds	Make Input Data conform to the expected SPC-specific data format expected by the Computation Task
P3.4	Submit Protected Input Data	IP, CPs	The Protected Input Data is submitted to the set Computing Parties assigned for the Computation Task; if the Computing Parties reject the submission due to a validation failure, the Input Party will repeat the preparation process	Protected Input Data is created	The set of Computing Parties obtain the Protected Input Data of one Input Party to be used in the subsequent Computation
P3.5	Accept Protected Input Data *(sub-process)*	CPs, IP	The Computing Party awaits submissions of Protected Input Data in accordance with the expected Input Data specification in the Computation Task Agreement; the contained activities are invoked on each submission, finishing after the Input Data is stored	Receipt of a concluded Computation Task Agreement, signifying the start of the Computation Task	Computing Parties collect the Protected Input Data to be used in the Computation
P3.5.1	Validate Protected Input Data Quality	CPs, IP	Optionally, validation is executed on Protected Input Data by Computing Parties using the MPC technology; if validation fails the corresponding Input Parties are notified	Protected Input Data has been received	Protected Input Data adheres to Data Quality criteria detailed in the Computation Task Agreement
P3.5.2	Store Protected Input Data	CPs	Protected Input Data of one Input Party is stored	Protected Input Data passes validation	Protected Input Data is present in the Computing Nodes ahead of its use in the Computation

Table 6. P3: Introduced data elements

ID	Name	Holder	Description
E3.1	Input Data	IP	Input Data prepared locally for a given Computation Task according to the Input Data specification in the Computation Task Agreement
E3.2	Protected Input Data	IP, CPs	Protected Input Data for a given Computation Task ready for submission to the Computing Parties
E3.3	Protected Input Data store	CPs	Store of all Protected Input Data for a given Computation task, i.e. the Input Data tables submitted by the Input Parties; this data element is initialised alongside the receipt of the Computation Task Agreement by the Computing Parties, i.e. as the Computation Task becomes effective, and is erased as the Computation finishes (or is prematurely terminated)

4.4.3 Computation Task execution

Once the Computation Task has been populated with all the expected Input Data tables, the data analysis steps – or Computation Task execution – may proceed. Only the exact data analysis function that was previously approved may run.

> **SYS-9.3** ***Requirement*** *(Privacy)*
> *The System shall not execute any other computing functions than the one approved in the Computing Task Agreement.*

The execution may be initiated in one of several ways: manually invoked by an Input or Output Party; scheduled for a specific time; or automatically run once all Input Data tables are received. It should be noted that the exact method of initiation has no foreseeable security or privacy implications; the most suitable candidate may emerge during the specification of System architecture.

> **SYS-6.4** ***Requirement*** *(Task and data lifecycle)*
> *The Computation Task shall support one or more of the following initiation methods: manual; scheduled (time-based); upon receiving all Protected Input Data.*

> **SYS-6.5** ***Requirement*** *(Task and data lifecycle)*
> *The System shall employ technical measures ensuring that all Protected Input Data and Interim Data is permanently and securely deleted or rendered permanently illegible immediately after completion of the Computation Task or upon reaching the deadline.*

> **SYS-9.4** ***Requirement*** *(Privacy)*
> *Any Interim Data produced in the System during the computation shall not be disclosed to any party.*

Executions may be long-running, taking anywhere from a few seconds to a few days to complete if the data sizes and/or the analysis function complexity grows. For the most part this is unavoidable due to the use of advanced cryptographic techniques used to hide the data in use. The problem can be mitigated, however, by opting for a more efficient MPC technology (requirement SYS-7.4), trading off security guarantees. As a baseline target figure, the candidate MPC technology should be able to perform basic operations (e.g., set intersection with exact matching) on 100 million records within three days.

SYS-7.5 ***Requirement*** *(Computational capabilities)*
Operations shall be able to run on pairs of Protected Input Data sets in tabular format of size up to 100 million records (rows) with up to 40 variables (columns) in less than 72 hours using commercially available hardware.

An overview the Computation Task execution process is shown in Figure 11 with descriptions of its steps and data elements in Tables 7 and 8. The deliverable *D4.1 System Specification and Architecture (first version)* details how MPC and TEE technologies are used together to provide a secure environment for the Computation.

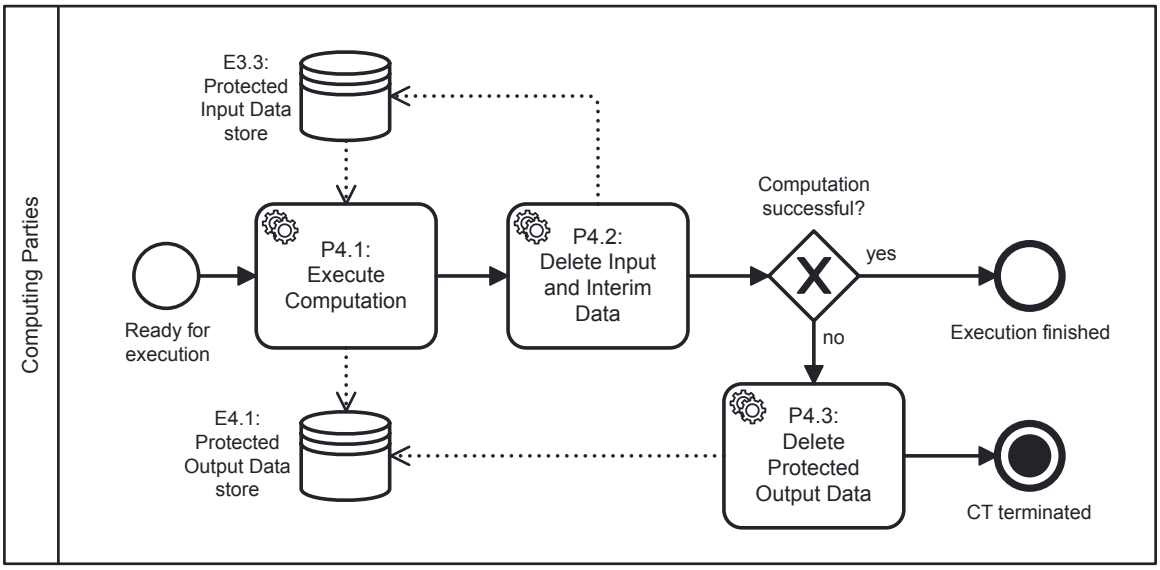

Figure 11. Sub-process P4: Execute Computation Task.

Table 7. P4: Process steps

ID	Name	Roles	Description	Trigger	Goal
P4.1	Execute Computation	CPs	Computation Task is executed, running the previously agreed the data analysis function and storing the resulting Protected Output Data	All expected Protected Input Data are received by the Computing Parties	Produce Protected Output Data
P4.2	Delete Input and Interim Data	CPs	Any Protected Input Data and Interim Data is securely deleted or otherwise rendered permanently illegible	Computation Task execution has finished	Ensure that Input and Interim Data is rendered unusable as soon as it is no longer required by the Computation Task

Table 7. P4: Process steps *(continued)*

ID	Name	Roles	Description	Trigger	Goal
P4.3	Delete Protected Output Data	CPs	Securely delete or otherwise render unusable any Protected Output Data that may have been created, preemptively terminating the Computation Task	The Computation finishes erroneously	Ensure that Protected Output Data from an erroneous Computation is discarded

Table 8. P4: Introduced data elements

ID	Name	Holder	Description
E4.1	Protected Output Data store	CPs, OP	Protected Output Data from the Computation, subject to be retrieved by Output Parties

4.4.4 Retrieval of Computation results

Once the Computation is complete, the Output Parties are notified and must retrieve their results in a given timeframe (specified by the retention deadline in the CTA, see Section 4.4 for details). Once all Output Parties have retrieved the Protected Output Data, all data connected to the Computation Task is deleted. This marks the successful finalisation of a Computation Task. If, however, any of the Output Parties fail to retrieve the Protected Output Data on time, the Computation Task along with the data will be erased.

> **SYS-6.6** ***Requirement*** *(Task and data lifecycle)*
> *The System shall employ technical measures ensuring that all Protected Output Data is permanently and securely deleted or rendered permanently illegible immediately after retrieval by all Output Parties or upon reaching the deadline.*

The retrieval process is coordinated through the Client Portal by issuing notifications to the Output Parties regarding required actions. As with the uploading of Input Data, the Client employs vendor-provided tools to download the Protected Output Data from the Computing Nodes and remove protection in order to reconstruct the final Output Data. The tools handle the authentication of the Client – Computing Nodes verify the Client's identity against the Computation Task Agreement, authorising only the pre-agreed Output Parties to receive the results.

> **SYS-9.5** ***Requirement*** *(Privacy)*
> *The System shall not deliver any information to Output Parties other than the final result predefined in the Computation Task Agreement.*

> **SYS-9.6** ***Requirement*** *(Privacy)*
> *The final result shall be disclosed only to the intended stakeholder(s) identified as Output Parties in the Computation Task Agreement.*

An overview the Computation Task output retrieval process is shown in Figure 12 with descriptions of its steps and data elements in Tables 9 and 10.

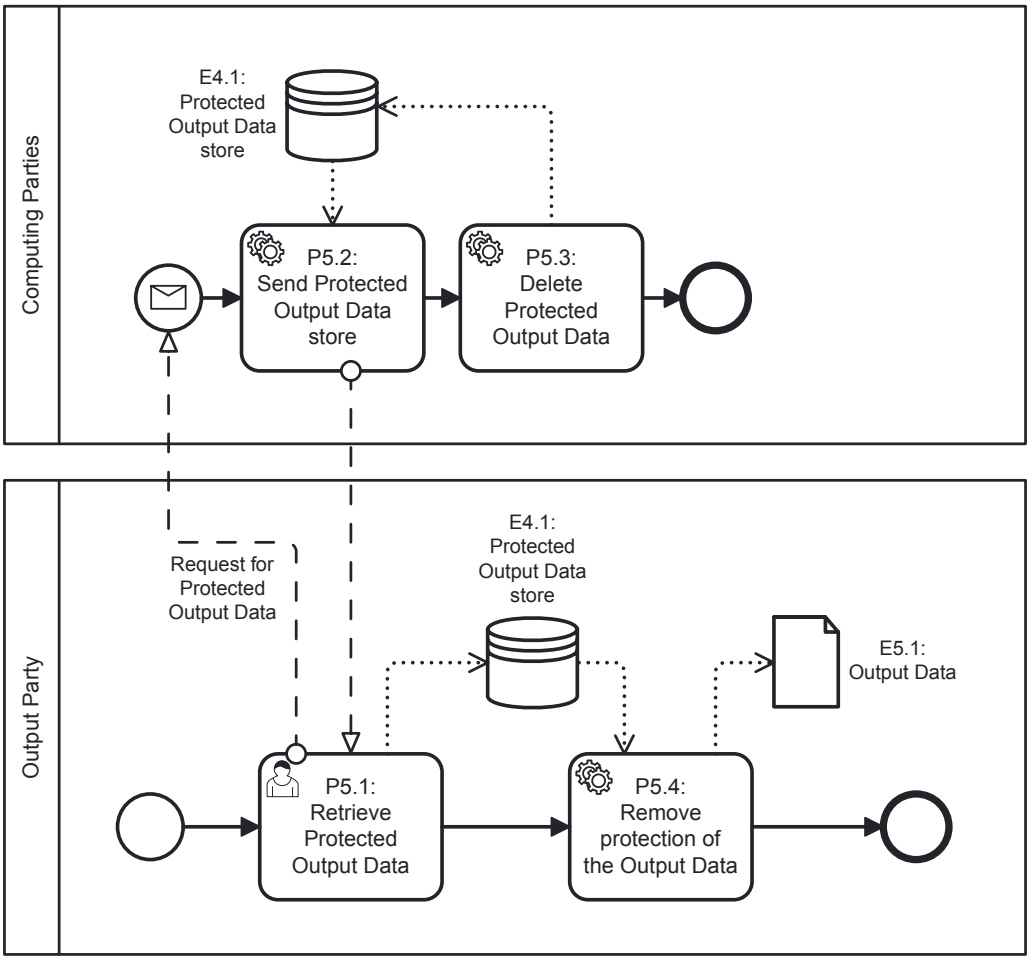

Figure 12. Sub-process P5: Retrieve Output Data.

Table 9. P5: Process steps

ID	Name	Roles	Description	Trigger	Goal
P5.1	Retrieve Protected Output Data	OP	The Output Party requests the Protected Output Data store from the Computing Parties	Notice of Protected Output Data available from the Computing Parties	The Output Party obtains the results of the Computation
P5.2	Send Protected Output Data store	CPs	Computing Parties send the Protected Output Data store to an authorised Output Party	Retrieval request from the Output Party	The Output Party obtains the results of the Computation

Table 9. P5: Process steps *(continued)*

ID	Name	Roles	Description	Trigger	Goal
P5.3	Delete Protected Output Data	CPs	Protected Output Data is securely deleted or otherwise rendered permanently illegible	All Output Parties have received their copy of Protected Output Data	Ensure that there is no copy of Protected Output Data left after all Output Parties have retrieved their copies
P5.4	Remove protection of the Output Data	OP	Protection is removed to obtain legible Output Data	Protected Output Data is present	The Output Party is able to use the Output Data as intended

Table 10. P5: Introduced data elements

ID	Name	Holder	Description
E5.1	Output Data	OP	Legible representation of Output Data; result of removing the protection of E4.1 Protected Output Data store

4.5 Logging and engagement of the System Auditor

To provide transparency and auditability, the System keeps two kinds of logs:

- **Computation Task Logs** contain everything that happens in the context of a specific Computation Task. First, they contain all of the Computation Task state changes, starting from when a new Computation Task Specification is created, when its signed by respective Parties, when Input Parties provide their Input Data, when the Computation in executed on the MPC infrastructure, etc. Therefore, the telemetry connected to a Computation Task is also part of the Computation Task Logs. As stated in the System Agreement summary (see Section 4.3), these logs are made available to all Parties participating in a given Computation Task as well as to the System Auditor.
- **System Logs** contain everything else that is not specific to any one Computation Task. This includes, for example, logs pertaining to the Management Plane Subsystems and Control Plane Subsystems, but also logs about System availability, configuration, etc. System Logs are made available to the System Auditor.

> **SYS-5.3** ***Requirement*** *(System)*
> *The System shall maintain a log of the proceedings of each Computation Task, containing Computation Task lifecycle events, and make it available to all Parties of the Computation Task.*

More detailed information about the exact contents of the logs and their retention will be provided in *D4.1 System Specification and Architecture (first version)*.

To maintain trust and ensure compliance with security and privacy policies, the System Operator appoints an independent System Auditor who conducts regular audits to assess whether the System is functioning as expected, particularly focusing on security controls, compliance

with legal regulations, and adherence to privacy guarantees, ensuring that the System remains reliable, compliant, and secure over time.

> **SYS-10.1** ***Requirement*** *(Auditing)*
> *The System shall provide auditing facilities in order to allow ex-post verification and detection of errors, attacks, and workflow deviation attempts.*

The auditor is responsible for:

- **Ex Post review of Computation Task execution:** The auditor checks that all Computation Tasks have been executed in compliance with the Computation Task Agreements. This includes verifying that data was handled securely and that no unauthorised access occurred, ensuring that no deviations occurred that could compromise confidentiality or integrity of the results.
- **Audit System components:** The auditor has access to System Logs, administration interfaces, Computation Task Specifications, and Computation Task Agreements to detect any irregularities or misbehavior by any Party, including the System Operator. The System maintains an audit trail that logs significant events, including data submission, CTA signing, CT execution, and output retrieval. The System Auditor ensures that these records are complete and that no tampering or unauthorised access has occurred.
- **Ensure data lifecycle compliance:** The auditor ensures that all Input and Output Data is deleted or rendered illegible once the Computation Tasks are complete, in line with the System's data lifecycle policies.

> **SYS-10.2** ***Requirement*** *(Auditing)*
> *The System shall maintain an audit trail that logs significant events, including the signing of Computation Task Agreements, Input Data submission, Computation Task execution, and Output Data retrieval.*

This auditing process provides an additional layer of security and trust, ensuring that the System operates in accordance with its privacy-preserving objective. Note that as a part of their auditing tasks, System Auditor is never given access to any Restricted Data. The latter is true as Restricted Data is not in legible from in any of the System Components that the System Auditor oversees. It only exists in the form of Protected Data in the independent Computing Nodes to which the System Auditor does not have direct access to.

Note that by default, the System Auditor role does not include auditing or reviewing the *software code* of System components. This remains as a task for the System Operator who is responsible for the System setup in general. However, the SO is free to delegate this responsibility, including to the same stakeholder that is in the role of the SA. More details on the secure deployment of the System will be provided in *D6.1 Trust Building Plan*.

4.6 Member offboarding

In a scenario where Members leave or are removed from the System, they are first removed from the IAM by the System Operator. The System Operator must consider whether or not to take action with respect to previously existing and already deployed Computation Tasks associated with the Member. For example, a termination of a Computing Party under the suspicion of being malicious must be effective immediately throughout the whole System, i.e. also in the MPC infrastructure to prevent further possible damage following the Computing Party's execution of subsequent Computation Tasks. Hence the System Operator must exercise their power

to abort the relevant Computation Tasks (see requirement BUS-3.3). The Control Plane Subsystems ensure that new Computation Tasks that contain offboarded Members could not be consolidated.

5 Use cases for prototype demonstration

In the previous chapters we have outlined the features of the envisioned System, whose specifications are developed in the JOCONDE Project. In order to demonstrate the feasibility of these specifications, the JOCONDE Project also includes development and testing activities involving a Prototype. The production System and the demonstration Prototype are two different objects: while the Prototype is developed in the JOCONDE Project, the System will be procured and developed in follow-up projects (subject to the successful completion of JOCONDE). While in principle both the System and the Prototype are based on the same set of specifications, the Prototype should be understood as a "lighter" version of the System with a reduced set of features.

In this chapter we focus on the Prototype and detail a small set of use cases to be implemented *for demonstration purposes* within the JOCONDE Project. The use cases presented in this chapter are selected to be illustrative of the real-world use cases of the future production System. We remark that the use cases presented hereafter are designed to serve the testing activities in the Project, not official statistics production. In other words, they serve as illustrative examples for testing the Prototype implementation, and should not be interpreted as detailed representations of official statistics products.

The use case descriptions in this chapter are written to be short and understandable to non-specialist readers. The descriptions emphasise what data attributes are being used as inputs, what is calculated and *most importantly: what output information (the final* statistics*) is provided to the agreed Output Party.* The content of this chapter will serve as input to the implementation and demonstration activities in the JOCONDE Project

5.1 Use case 1: Intersection of country population registers

Each EU country maintains a list of registered residents in the country. The National Statistical Institutes (NSI) of two EU countries (C1, C2) want to analyse dual residency[1]. To achieve this they need to learn how many people are jointly registered in both countries at some given point in time. We assume the two NSIs have compiled two sets of records, whereby each record corresponds to an individual person registered in their countries. Each record (row) contains certain input attributes (columns) detailed below. The two NSIs are interested to compute the number of records at the intersection between the two sets, whereby the intersection operation is defined by some matching criteria defined below. We assume the goal of the Computation Task is to disclose only the number of intersection records, while their identities must remain protected. In other words, we want to discover *How many, not Which ones* have dual residency.

Input attributes:

1. Birth date: day, month, year in format `dd.mm.yyyy`
2. Name (Standardised): string of characters from the English alphabet
3. Sex: (M, F, others) because the countries can have different encodings of the attribute it needs standardisation
4. Country of birth (Standardised)

[1] For a discussion of dual residency in EU, see https://emifast.com/blog/the-pros-and-cons-of-dual-residency-in-europe-a-comprehensive-guide/

Matching criteria by attribute

To better evaluate System performance[2], this use case is split into two versions – UC-1a and UC-1b – that differ on how attribute matching is accomplished.

UC-1a is based on exact matching on all attributes:

- **Birth date**: exact match
- **Name**: exact match
- **Sex**: exact match
- **Country of birth**: exact match

Recalling SYS-7.5 we require that, for this use case, the computation run in less than 72 hours when with Input Data tables C1 and C2 containing 50 million rows (around 100 million records in total).

UC-1b is based on a mixture of fuzzy matching for *Name* and *Sex* attributes and exact matching for the remaining attributes:

1. **Name:** matching based on Levenshtein distance (or other similar string distance metric) falling below a given threshold. The rationale for considering this criterion is rooted in possibly imperfect name standardisation due to alphabet differences across various EU languages (e.g. a person registered as "Köbler" in Germany may be registered as "Koebler" or "Kobler" in Italy).
2. **Sex:** matching based on the truth table below. The rationale for this criterion is that a non-binary person may have the possibility to declare herself/himself as "non-binary" in one country but not in the other, e.g., due to perceived risk of discrimination in the other country, or simply because the non-binary option is not available therein. The following truth table for the equality function is proposed (T – matching: true; F – matching: false):

Sex	Male	Female	Non-binary
Male	T	F	T
Female	F	T	T
Non-binary	T	T	T

Output result. For both versions UC-1a and UC-1b, the following indicator will be computed and delivered to the Output Parties:

- **N** – number of residents common in C1 and C2 registers (size of set intersection).

5.2 Use case 2: Roaming Mobile Network Operator data

This use case was inspired by the work conducted in the parallel project Multi-MNO[3].

[2] While the fuzzy matching used in UC-1b needs to be developed for the operational system, to be robust to collusion of, and intrusion into, two out of three computing parties, the testing in the Prototype may be conducted with lowered security assumptions and working times extrapolated for predicting those of the final system.

[3] Multi-MNO project page: https://cros.ec.europa.eu/landing-page/multi-mno-project. See specifically the *Deliverable D2.2 – Methodology framework: high-level architecture, requirements, use cases and methods*.

Let us consider a sample EU country X with 3 MNOs $X1, X2$ and $X3$. During a reference period of $W = 6$ months, $N = 10$ million mobile users from other countries come to visit country X and roam across the mobile networks of the three MNOs. These mobile users represent *inbound roamers* from the perspective of the visited MNOs. We consider three flows of inbound roamers:

Tourists: Each tourist k arrives to and departs from the country at times

$$ta_k$$

and

$$td_k = ta_k + d_k,$$

wherein d_k represents the *length of stay*, from less than one day to multiple weeks. For simplicity's sake we assume a single visit per tourist, i.e. tourists do not return to country X after the first visit.

Border residents: They live in border areas and roam into the network for short periods of time due to differences in network reception in places they move in. Their appearance in operator data is sporadic.

Long-term roamers: They stay in the country continuously during the whole observation period.

During their stay, the visitors generate *signalling events* across multiple MNOs, according to some distribution. These events are recorded by the MNOs and summarised into digests for each visitor. The digest represents a summary of the signalling events seen by each MNO for the individual roamer k in the 24 hour time window. Digest contains the following attributes.

Input Attributes:

1) **Mobile Country Code** (MCC) of the inbound roamer.
2) **Mobile Network Code** (MNC) of the inbound roamer.
3) **Pseudonym** of the mobile user.
   Important: We assume the MNOs adopt a coordinated pseudonymisation function so that they all assign the same pseudonym to the same mobile user.
4) **Temporal distribution** *Bitmap* representing the temporal distribution of the events seen by the MNO for the individual inbound roamer. We assume the 24-hour period is slotted into one hour intervals. For each generic slot, the bit is set to 1 if at least one event was seen in that slot for user k, 0 otherwise. The maximum length of the bitmap (when it encodes the whole 6-month period, i.e., 180 days) is therefore $24 \times 180 = 4320$ bits (0.53 kilobytes). Since the distribution of the *length of stay* is skewed towards a shorter stay (tourists will spend between 1 day and 2 weeks in the country), more efficient encodings are possible (e.g. encode explicitly the first and last time slots, and limit the bitmap to encode only the interval between these two).

Record matching

- MCC, MNC, pseudonym – exact matching

For each visitor k, generate a *combined bitmap* by performing bitwise OR on the bitmaps from the different input parties. Identify and filter away *Border residents* and *Long-term roamers*, based on heuristic rules. When designing these heuristic rules, it is important to keep in mind that the chosen algorithm is to be implemented for an MPC system. Therefore, a known heuristic that is used for plaintext data (e.g., finding the longest substring of repeating ones in the given bitstring) might not be performant enough. For MPC systems, simple computations that take advantage of the "single instruction, multiple data" principle, are preferred. For example, in this

specific case, it is possible to divide the whole bitstring into y-bit slices and count (in parallel) how many of those slices are all ones. This also carries information on whether someone stayed in the network for y contiguous hours.

During the calculation of aggregated outputs per user indicators need to be calculated first. Please note that this data is not Output Data but is used in further computations in protected form. For each remaining tourist k (after filtering out *Border residents* and *Long-term roamers*), generate descriptive indicators:

a) Pseudonym;

b) Number of nights spent in the country;

c) Total length of the stay;

d) Date of arrival in the country: $min()$ operation between the first event seen by each MNO;

e) Date of departure from the country: $max()$ operation between the last event seen by each MNO

Output result. For UC-2 the following indicators will be computed and delivered to the Output Parties:

a) Number of border residents;

b) Number of long-term roamers;

c) Histogram of quantified length of stay;

d) Histogram of number of nights spent in the country;

e) Histogram of date of arrival;

f) Histogram of date of departure

Bibliography

[1] *Input Privacy-Preservation for Official Statistics Project: Final Report*. United Nations Economic Commission for Europe (UNECE), 2023. URL: https://statswiki.unece.org/x/mQCQFw.

[2] Fabio Ricciato. "Steps Toward a Shared Infrastructure for Multi-Party Secure Private Computing in Official Statistics". In: *Journal of Official Statistics* 40.1 (2024), pp. 3–15. DOI: 10.1177/0282423X241235259.

[3] Peter M Mell and Timothy Grance. "SP 800-145. The NIST Definition of Cloud Computing". In: *National Institute of Standards & Technology, Gaithersburg, MD, United States, Tech. Rep* (2011).

[4] Peter Bogetoft et al. "Secure Multiparty Computation Goes Live". In: *13th International Conference of Financial Cryptography and Data Security. FC'09*. 2009, pp. 325–343.

[5] Dan Bogdanov et al. "Students and Taxes: a Privacy-Preserving Study Using Secure Computation". In: *PoPETs* 2016.3 (2016), pp. 117–135. URL: http://www.degruyter.com/view/j/popets.2016.2016.issue-3/popets-2015-0019/popets-2016-0019.xml.

[6] Dan Bogdanov, Peeter Laud, and Jaak Randmets. "Domain-Polymorphic Programming of Privacy-Preserving Applications". In: *Proceedings of the Ninth Workshop on Programming Languages and Analysis for Security*. PLAS'14. Uppsala, Sweden: ACM, 2014, pp. 53–65. URL: http://doi.acm.org/10.1145/2637113.2637119.

[7] Jaak Randmets. "Programming Languages for Secure Multi-party Computation Application Development". PhD thesis. University of Tartu, 2017. URL: http://hdl.handle.net/10062/56298.

[8] John Liagouris et al. "SECRECY: Secure collaborative analytics in untrusted clouds". In: *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. Boston, MA: USENIX Association, Apr. 2023, pp. 1031–1056. ISBN: 978-1-939133-33-5. URL: https://www.usenix.org/conference/nsdi23/presentation/liagouris.

[9] Rishabh Poddar et al. "Senate: A Maliciously-Secure MPC Platform for Collaborative Analytics". In: *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Aug. 2021, pp. 2129–2146. ISBN: 978-1-939133-24-3. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/poddar.

[10] Wenjing Fang et al. "SecretFlow-SCQL: A Secure Collaborative Query pLatform". In: *Proc. VLDB Endow.* 17.12 (2024), pp. 3987–4000. URL: https://www.vldb.org/pvldb/vol17/p3987-fang.pdf.

[11] Dan Bogdanov et al. "Rmind: a tool for cryptographically secure statistical analysis". In: *IEEE Transactions on Dependable and Secure Computing* PP.99 (2016), pp. 1–14. DOI: 10.1109/TDSC.2016.2587623.

Glossary

Client
: Member who uses System facilities to perform Computation Tasks on-demand. They are in the role of Input Party, Output Party or both.

Client Portal
: a component of the System where Clients consolidate Computation Task Specifications. Part of the Control Plane Subsystems.

Computation
: the execution of a Computation Task.

Computation Task
: realization of a Computation Task Specification which is deployed across the distributed MPC infrastructure.

Computation Task Agreement
: a legally enforceable agreement between the Input, Output and, where necessary, System Operator to create and manage a Computation Task. Signed version of the Computation Task Specification.

Computation Task Logs
: logs concerning a specific Computation Task, e.g., its state changes and any logging output produced by the data analysis algorithm itself.

Computation Task Specification
: definition of a Computation Task including (among other details) a human-readable description, the corresponding computation algorithm, data-model, identities of all involved Computing Parties, Users and assignment of roles.

Computing Node
: one of the (at least three) servers hosting the MPC technology needed for executing the Computation Tasks.

Computing Party
: an independent Member in the system who owns, operates or otherwise provides a Computing Node in order to execute Computation Tasks.

Contractor
: a third party non-Member who is providing products or services to the System Operator or Members to ensure the proper functioning of the System or Computation Task.

Control Plane
: one of the three conceptual layers of the System, encompassing the planning, deployment, and coordination of Computation Tasks.

Control Plane Subsystems
: System components, services, or tools that facilitate Control Plane activities.

Cybernetica
: a public limited company called Cybernetica AS, which is registered in the Estonian commercial register under registry code 10140133.

Data Plane
: one of the three conceptual layers of the System, encompassing any operation on Restricted Data.

Default set of Computing Parties
three independent organizations pre-selected by the System Operator to act as Computing Parties for the Computing Tasks where Input Parties do not want to change them e.g. Input Party needing a particular Computing Party to participate due to internal policy requirement.

European Commission
is the executive body of the European Union.

European Union
is an economic and political community of 27 European countries.

Eurostat
the European Union statistical authority, a Directorate-General of the European Commission.

Identity and Access Management
framework for managing and controlling access to information systems and resources based on User identities and their associated roles and permissions; it encompasses authentication, authorization, and account provisioning.

Input Data
the pre-existing data (sets) used as input for a Computation Task.

Input Party
Member who provides Input Data for a Computation Task.

Interim Data
auxiliary Protected Data which may be created as a byproduct during the Computation Task execution, to be used by the Computation Task as an intermediate result.

Management Plane
one of the three conceptual layers of the System, encompassing System management and auditing activities.

Management Plane Subsystems
System components, services, or tools that facilitate Management Plane activities.

Member
see *System Member*.

MPC Engine
a program utilising the chosen MPC technology that executes Client-defined computation code in the Data Plane of the System.

MPC infrastructure
a set of Computing Nodes allocated for a Computation Task.

Multi-Party Secure Private Computation
Secure Private Computation (or computation under input privacy) in which inputs are sourced from multiple parties [2].

Operational System
the production ready System that has gone through the development, testing and risk assessment and is deemed ready for analysing confidential microdata by ESS members and all stakeholders.

Output Data
the new data (sets) combined from the Output Secret Shares.

Output Party
Members who receive Output Secret Shares from a specific Computation Task.

Parties
: set of System Members connected to a specific Computation Task, i.e., any Input Party, Output Party or Computing Party referenced in a given Computation Task Specification.

Procurement
: the procurement procedure carried out by Eurostat as the contracting authority under tender reference number ESTAT/2023/OP/0004. For further information, please see the TED eTendering website: https://etendering.ted.europa.eu/cft/cft-display.html?cftId=12503 (09.07.2024).

Project
: the joint project titled "JOCONDE" initiated between Eurostat and Cybernetica as a result of the Procurement. For further information, please see the Project website: https://cros.ec.europa.eu/joconde/ (08.07.2024).

Protected Data
: a representation of data that is made illegible by applying cryptographic techniques specific to the MPC technology. The representation allows the System to perform computations on the data without removing the protection.

Protected Input Data
: a representation of Input Data that is illegible for any unauthorized set of entities.

Protected Output Data
: a representation of Output Data that is illegible to everybody except the authorised Output Party.

Prototype
: a "lighter" version of the System with a reduced set of features, supporting *in vitro* functional testing by test users in order to demonstrate the technical feasibility of the SPC servitisation concept.

Restricted Data
: Confidential Input Data or Output Data that must be kept secret from other Members of the System and third parties due to regulatory (e.g. data protection or confidentiality requirements) or other reasons.

Secret sharing
: cryptographic technique, a message sharing algorithm, to protect the confidentiality of a message by dividing it into a number of pieces called shares (source: ISO/IEC 19592-1[4]).

Secure hardware
: hardware which can protect sensitive data of some remotely attestable, select business logic, from any extraneous co-located business logic, even while the data is being processed. E.g. hardware with TEE support.

Secure Multi-Party Computation
: technique for evaluating a function with multiple peers so that the agreed party learns the output value but not each other's inputs.

Secure Private Computation
: computation technique that provides input privacy [2].

SPC technology
: software product, platform, application, or library that enables the Secure Private Computation paradigm, i.e. it can be used for the intended purpose of private computation.

[4] ISO/IEC 19592-1:2016(en) Information technology — Security techniques — Secret sharing — Part 1: General https://www.iso.org/obp/ui/en/#iso:std:iso-iec:19592:-1:ed-1:v1:en

Side-channel attack
attack based on information gained from the physical implementation of a cryptosystem, rather than on brute force or theoretical weaknesses in the underlying algorithms

Example: Timing information, power consumption, or electromagnetic emissions can provide extra sources of information and can be exploited to attack the system. (Source: ISO/IEC 29192-1[5]).

System
an ICT solution implementing the MPSPCaaS concept, whereby ESS members and their partners could perform on-demand SPC tasks on their respective data without sharing it in intelligible form, neither with each other nor with an external Trusted Third Party (TTP).

System Agreement
a legally enforceable agreement between a Member and the System Operator regarding the membership in and use of the System.

System Auditor
an authorized entity (independent from the System Operator) who provides auditing services to verify the correctness of the operation of the System by, for example, detecting errors, attacks or attempts to deviate from the System workflow.

System Logs
logs that transcend any specific Computation Task.

System Member
a legal person or other entity with an autonomous information system who has been accepted by the System Operator to interface with the System.

System Operator
Eurostat (European Commission) within the capacity to manage membership and coordinate communication between Members in the System.

System Participant
any System Provider or System Member.

System Provider
System Participants acting on Management Plane, i.e., System Operator and System Auditor.

System User
a natural person who is authorised to use the System on behalf of a System Member.

User
see *System User*.

Vendor
a third party who is providing products or services in order to ensure the proper functioning of the System. Examples include software providers and hardware suppliers.

[5] ISO/IEC 29192-1:2012(en) Information technology — Security techniques — Lightweight cryptography — Part 1: General https://www.iso.org/obp/ui/en/#iso:std:iso-iec:29192:-1:ed-1:v1:en

Abbreviations

Common

CP
Computing Party

CT
Computation Task

CTA
Computation Task Agreement

CTS
Computation Task Specification

CSP
Cloud Service Provider

IAM
Identity and Access Management

IP
Input Party

MNO
Mobile Network Operator

MPC
Secure Multi-Party Computation

NSI
National Statistics Institute

OoB
Out-of-Band

OP
Output Party

PET
Privacy Enhancing Technology

SA
System Auditor

SPC
Secure Private Computing (aka input privacy techniques)

SO
System Operator

TEE
Trusted Execution Environment (an input privacy technique based on *specific CPU extensions*)

TTP
Trusted Third Party

Organisations

EC, Commission
: European Commission

ESS
: European Statistical System

EU, Union
: European Union

UNECE
: United Nations Economic Commission for Europe

List of deliverables

Work in Project JOCONDE is divided into six tasks, each producing one or more deliverables. For context, the following is the list of tasks along with a few earlier deliverables that are referenced from this document:

- Task 1 – Usage scenarios and system requirements
  - *D1.1 Usage Scenarios and System Requirements* (this document)
  - *D1.2 Usage Scenarios and System Requirements (final version)*
- Task 2 – Technology analysis
  - *D2.1 Technology Survey and Analysis*
  - …
- Task 3 – Legal aspects
  - *D3.1 Initial Legal Analysis*
  - *D3.2: Draft of reference DPIA and model agreements*
  - …
- Task 4 – System specifications and architecture
  - *D4.1 System Specification and Architecture (first version)*
  - …
- Task 5 – Demonstrator prototype and functional testing
  - …
- Task 6 – Trust building plan
  - *D6.1 Trust Building Plan*

Appendix A Business requirements

Business requirements focus on the goals and objectives that the System is intended to achieve; they are identified by the prefix *BUS*. To provide context, business requirements are arranged into five groups: System roles, governance roles, powers, policies, and supplementary.

ID	Requirement
System roles	
BUS-1.1	Input Parties shall make their Input Data available in protected form to the Computation Task.
BUS-1.2	Output Parties shall extract results of the Computation Task to use in a statistical product.
BUS-1.3	Computing Parties shall contribute resources to ensure availability of the System to carry out secure computations on demand.
BUS-1.4	Clients shall be able to independently initiate new Computation Tasks with minimal manual intervention by others to streamline operations and lessen personnel workload.
BUS-1.5	Clients can invite other Clients to participate in their Computation Task as Input or Output Parties.
BUS-1.6	All Input and Output Parties must understand and approve the details of the planned statistical computation, forming a Computation Task Agreement, before it could be carried out to ensure the acknowledgement of any residual risk.
Governance roles	
BUS-2.1	The System Operator organises the selection, onboarding, and attesting of Computing Parties, checking compliance with all legal and technical requirements before they are admitted to operate in the System.
BUS-2.2	The System Operator appoints three Computing Parties to act as the default for all Computation Tasks for which the System Members have not identified specific Computing Parties.
BUS-2.3	The System Operator appoints System Auditors.

ID	Requirement
BUS-2.4	The System Operator operates the System components responsible for collaboration and coordination mechanisms in a manner which, to a reasonable degree, rules out any possibility of attacks against System Members and their data.
BUS-2.5	A System Auditor shall audit the System to detect misbehaving components or actors including the System Operator, System Members, Computing Parties, and external entities that could put Restricted Data at risk.
BUS-2.6	The System Auditor should be independent from the Computing Parties.
BUS-2.7	System Members shall undergo a manual onboarding procedure with the System Operator for ensuring conformance with all legal and technical requirements set in the System Agreement before being able to use the System.
BUS-2.8	The System Operator shall not be a single point of trust.
Powers	
BUS-3.1	System Members shall have their access rights associated with a given Computation Task, limited in accordance with the principle of least privilege to permit only the minimal set of actions necessary for their role(s) in the Computation Task at hand.
BUS-3.2	By default, all Computation Tasks shall be approved by the System Operator before execution in order to enforce compliance to System Agreement. The System Operator shall be able to disable this option if the context of the deployment allows it.
BUS-3.3	The System Operator shall be able to terminate a Computation Task at any time to mitigate a System violation, abuse, or possible harm when detected.
BUS-3.4	Each Input Party shall be able to terminate an ongoing Computation Task at any time, in line with conditions outlined in the System Agreement, if it detects potential violation of Computation Task Agreement.
BUS-3.5	Input and Output parties shall be able to agree on a custom set of Computing Parties to use for a specific Computation Task.
BUS-3.6	Parties involved in a Computation Task shall be able to reliably verify the identity of each other and the Computing Parties to mitigate attacks involving impersonation.
BUS-3.7	A System Auditor should be able to audit the correctness of a Computation Task after its completion.

ID	Requirement
BUS-3.8	A System Auditor should have access to all Computation Task Specifications, Computation Task Agreements, Computation Task Logs and System Logs.
Policies	
BUS-4.1	Restricted Data access controls for the Computation Task shall be expressed in a non-proprietary formal language.
BUS-4.2	No single entity shall have the possibility to reveal the Protected Data (Input, Interim, or Output Data), unless explicitly stated in the Computation Task Agreement.
BUS-4.3	The Computation Task Agreement shall be legally valid and enforceable.
BUS-4.4	A Computation Task shall be executed only after approval by all Input and Output Parties.
BUS-4.5	Computing Parties shall strictly follow the Computation Task Agreement for all decisions in order to counter any possibility of data being used outside of the pre-agreed context.
BUS-4.6	There shall be at least three Computing Parties for any particular Computation Task.
BUS-4.7	A Computing Party shall be an independent legal entity from other Computing Parties.
BUS-4.8	No single entity shall have control over more than one Computing Party involved in a Computation Task.
BUS-4.9	Protected Input Data and Interim Data connected to a Computation Task shall be securely deleted or rendered permanently illegible once the Computation finishes or the deadline is reached.
BUS-4.10	Output Data connected to a Computation Task shall be securely deleted or rendered permanently illegible once the result is delivered to Output Parties or the deadline is reached.
Supplementary	
BUS-5.1	All operations within the System shall be as simple and lightweight as possible for the Clients with only marginal costs in order to lower the barrier of entry for utilising PETs for statistics.
BUS-5.2	The System shall provide Clients an up-to-date overview of active workflows and status updates for ongoing Computation Tasks.

ID	Requirement
BUS-5.3	Computation Tasks shall expect Input Data in a well-defined tabular format.
BUS-5.4	The Data Plane shall support a portfolio of multiple different MPC and TEE technologies.
BUS-5.5	All Management Plane and Control Plane components shall be open source.
BUS-5.6	The System should incorporate security measures at both the hardware and software levels, complementing one another in order to secure the computation environment and achieve the highest possible degree of protection and trustworthiness.

Appendix B System requirements

System requirements focus on the technical aspects of the software, identified by the prefix *SYS*. To provide context, system requirements are arranged into ten groups: System security, trust, using the System, managing the System, System, task and data lifecycle, computational capabilities, task I/O (input and output), privacy, and auditing.

Requirements are classified as either *hard* or *soft* – a *soft* requirement is qualitative (or aspirational) in nature, meaning it poses an important dimension to maximize, while a *hard* requirement is quantifiable (or prescriptive) and must be met as-is. The requirements classified as *hard* can be further separated into tiers. *Hard* requirements marked as *required* are deemed critical for the success of the System, while *recommended* requirements are desirable, but not mandatory. *Optional* requirements can be seen as desirable in certain scenarios, but are also not mandatory.

ID	Requirement	Class	Tier	Plane
System security				
SYS-1.1	The System shall use techniques that split the Input Data elements and/or the encryption keys across multiple nodes by applying secret sharing, multi-key homomorphic encryption or other MPC schemes in combination with secure hardware.	hard	required	data
SYS-1.2	The technology and implementation of choice for any operation on Protected Data shall be robust to collusion of two out of three Computation Parties.	hard	required	data
SYS-1.3	The System shall be robust to intrusions at up to two out of three Computing Nodes.	hard	required	data
SYS-1.4	The System shall detect attacks on data integrity during the execution of a Computation Task, causing it to halt; an attack shall not cause the deliverance of an incorrect result which cannot be distinguished from a correct result.	hard	required	data
SYS-1.5	Upon detecting an attack on data integrity the System should reveal the party at fault.	hard	optional	data
SYS-1.6	The System should incorporate security measures at both the hardware and software level to secure the computation environment.	soft		data control

ID	Requirement	Class	Tier	Plane
SYS-1.7	The Computing Nodes shall employ secure hardware technologies, e.g. TEE with hardware isolation.	hard	required	data control
SYS-1.8	The System shall be robust against side-channel attacks.	soft		data
SYS-1.9	The System shall be robust against software and hardware attacks.	soft		data
SYS-1.10	The System should use a combination of multiple technologies and security/privacy layers with complementary security guarantees to achieve the highest possible degree of protection and trustworthiness.	soft		data control
Trust				
SYS-2.1	The System shall enable Users to verify identities of other Members contained in the Computation Task Specification locally. The verification shall not rely on trust in the System Operator.	hard	required	control
SYS-2.2	The System shall use Computation Task Specification approval mechanism that provides integrity and non-repudiation.	hard	required	control
SYS-2.3	The System shall allow Input and Output parties to verify the integrity of their Computation Task Agreements deployed in Computing Nodes directly.	hard	required	control
Using the System				
SYS-3.1	Preparation, configuration, and execution of a Computing Task in the System should be as simple and lightweight as possible for the Clients and should involve only minimal marginal costs.	soft		control
SYS-3.2	Technical requirements for Clients of the System in the role of Input Party and/or Output Party should be minimal – without the need to install specialised hardware.	soft		data control
SYS-3.3	The System shall allow for flexible configuration of Computing Tasks serving different Clients, i.e. in the role of Input Party, Output Party or both at the same time.	hard	required	control

ID	Requirement	Class	Tier	Plane
SYS-3.4	The System shall support Input Parties and/or Output Parties taking the role of a Computing Party in Computation Tasks which they are a part of.	hard	required	data control
Managing the System				
SYS-4.1	The System shall provide the System Operator with Identity and Access Management (IAM) for Member management and access.	hard	required	management
System				
SYS-5.1	The distributed secure multi-party computing infrastructure of the System shall consist of at least three distinct Computing Nodes.	hard	required	data management
SYS-5.2	Computing Nodes shall only accept and enforce Computation Task Agreements sourced by trusted means.	hard	required	control
SYS-5.3	The System shall maintain a log of the proceedings of each Computation Task, containing Computation Task lifecycle events, and make it available to all Parties of the Computation Task.	hard	required	control data
Task and data lifecycle				
SYS-6.1	The System shall associate all data with its respective Computation Task to enforce data lifecycle policies.	hard	required	data control
SYS-6.2	The Computation Task shall expect time deadlines for task lifecycle stages, specifically manual I/O operations including Protected Input Data upload and Protected Output Data retention.	hard	required	control
SYS-6.3	The System should provide telemetry about the progress of a Computation Task to Input and Output Parties in order to help coordinate their actions (e.g. if the task is waiting for input from a specific Input Party).	hard	recommended	control

ID	Requirement	Class	Tier	Plane
SYS-6.4	The Computation Task shall support one or more of the following initiation methods: manual; scheduled (time-based); upon receiving all Protected Input Data.	hard	required	control
SYS-6.5	The System shall employ technical measures ensuring that all Protected Input Data and Interim Data is permanently and securely deleted or rendered permanently illegible immediately after completion of the Computation Task or upon reaching the deadline.	hard	required	data
SYS-6.6	The System shall employ technical measures ensuring that all Protected Output Data is permanently and securely deleted or rendered permanently illegible immediately after retrieval by all Output Parties or upon reaching the deadline.	hard	required	data
Computational capabilities				
SYS-7.1	The operations on private Input Data shall include all combinations of elementary private set operations (private set union, private set intersection, private set difference) along with simple computations on the items of the resulting set.	hard	required	data
SYS-7.2	The System shall support exact record-matching based on common keys (identifiers or concatenation of variables values).	hard	required	data
SYS-7.3	The System should support additional operations (e.g. probabilistic record-matching).	soft		data
SYS-7.4	The System should allow Members to choose the MPC technology for each Computation Task and specify the chosen technology in the Computation Task Specification.	hard	recommended	data control
SYS-7.5	Operations shall be able to run on pairs of Protected Input Data sets in tabular format of size up to 100 million records (rows) with up to 40 variables (columns) in less than 72 hours using commercially available hardware.	hard	required	data
Task I/O				
SYS-8.1	The System shall support Computation Tasks with at least two Input Parties and at least one Output Party.	hard	required	control

ID	Requirement	Class	Tier	Plane
SYS-8.2	The System should support Computation Tasks with more than one Output Party.	hard	recommended	control
SYS-8.3	The Computation Task shall support Input Data in tabular format.	hard	required	data control
SYS-8.4	The Computation Task shall expect a detailed data model for each Input Data table as a part of the Computation Task Specification.	hard	required	control
SYS-8.5	The Computation Task should support multiple Input Data tables per input party.	hard	recommended	data control
SYS-8.6	The Computation Task shall support Output Data values in scalar and vector formats.	hard	required	data
SYS-8.7	The Computation Task shall support multiple Output Data values.	hard	required	data
SYS-8.8	The Computation Task should allow configuring individual Output Data values to only be retrieved by specific Output Parties.	hard	recommended	data control
SYS-8.9	The Computation Task should support server-side data validation or data quality checks during Protected Input Data upload based on custom data quality assurance algorithms, part of the Computation Task Specification.	hard	recommended	data control
Privacy				
SYS-9.1	The System shall not disclose Input Data values to anyone other than the Input Party, unless explicitly stated otherwise in the Computation Task Specification.	hard	required	control
SYS-9.2	No single entity shall have the ability to learn any Input Data values of any individual computation task, unless explicitly stated in the Computation Task Specification.	hard	required	data
SYS-9.3	The System shall not execute any other computing functions than the one approved in the Computing Task Agreement.	hard	required	control

ID	Requirement	Class	Tier	Plane
SYS-9.4	Any Interim Data produced in the System during the computation shall not be disclosed to any party.	hard	required	control
SYS-9.5	The System shall not deliver any information to Output Parties other than the final result predefined in the Computation Task Agreement.	hard	required	control
SYS-9.6	The final result shall be disclosed only to the intended stakeholder(s) identified as Output Parties in the Computation Task Agreement.	hard	required	control
Auditing				
SYS-10.1	The System shall provide auditing facilities in order to allow ex-post verification and detection of errors, attacks, and workflow deviation attempts.	hard	required	management
SYS-10.2	The System shall maintain an audit trail that logs significant events, including the signing of Computation Task Agreements, Input Data submission, Computation Task execution, and Output Data retrieval.	hard	required	management