D8.1

Components Requirements

WP8 – C3ISP Data Sharing, Analytics and Crypto Technology Maturation

C3ISP

Collaborative and Confidential Information Sharing and Analysis for Cyber Protection

Due date of deliverable: 30/09/2017
Actual submission date: 30/09/2017

29/09/2017
Version 1.3

Responsible partner: CNR
Editor: Paolo Mori
E-mail address: paolo.mori@iit.cnr.it

Project co-funded by the European Commission within the Horizon 2020 Framework Programme		
Dissemination Level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

The C3ISP Project is supported by funding under the Horizon 2020 Framework Program of the European Commission DS 2015-1, GA #700294

Authors: Paolo Mori, Ilaria Matteucci, Andrea Saracino, Gianpiero Costantino (CNR), Carmela Gambardella, Mirko Manea (HPE), Ali Sajjad (BT), Vincent Herbert (CEA), Francesco Di Cerbo (SAP)

Approved by: Carmela Gambardella, Mirko Manea (HPE), Jovan Stevovic (CHINO)

Revision History

Version	Date	Name	Partner	Sections Affected / Comments
0.1	25/07/2017	Paolo Mori	CNR	Initial ToC
0.2	04/08/2017	Carmela Gambardella	HPE	Section 3.1
0.2	25/08/2017	Ali Sajjad	BT	Section 8
0.3	25/08/2017	Vincent Herbert	CEA	Section 7 and 7.2
0.4	28/08/2017	Ali Sajjad	BT	Section 6
0.5	29/08/2017	Francesco Di Cerbo	SAP	Section 7.1
0.6	29/08/2017	Paolo Mori	CNR	Section 3 and Section 4.1
0.7	01/09/2017	Carmela Gambardella	HPE	Revised Section 3.1.1
0.8	01/09/2017	Andrea Saracino	CNR	Section 5
0.81	06/09/2017	Ilaria Matteucci	CNR	Section 3.2
0.82	08/09/2017	Paolo Mori	CNR	Section 1
0.9	14/09/2017	Paolo Mori, Mirko Manea	CNR/HPE	Section 2
0.91	14/09/2017	Francesco Di Cerbo	SAP	Section 4.2
0.92	15/09/2017	Vincent Herbert	CEA	Reworked Section 7.2
1.0	21/09/2017	Paolo Mori, Andrea Saracino, Gianpiero Costantino	CNR	Introduction and conclusion, reworked figures 3 and 5, improved section 5
1.1	24/09/2017	Jovan Stevovic	CHINO	Review
1.2	25/09/2017	Carmela Gambardella, Mirko Manea	HPE	Review
1.3	29/09/2017	Paolo Mori	CNR	Addressed reviewers' comments

Executive Summary

This deliverable is the first output of WorkPackage 8, “C3ISP Data Sharing, Analytics and Crypto Technology Maturation” due at M12 and its main aim is to report the requirements of the components of the C3ISP Framework reference architecture (defined in D7.2 and, for the convenience of the readers, reported in Section 2) that will be developed in WP8 by leveraging on the tools and technologies that are provided by the C3ISP partners. Hence, this deliverable reports a description of each of the provided technologies and tools, specifying the current Technology Readiness Level (TRL) and whether they have been developed within a previous EU project. Another important contribution of this deliverable is that it also specifies which of the components of the C3ISP Framework reference architecture can exploit the tools provided by the C3ISP partners for its implementation. In addition, this deliverable specifies the requirement of these components. It starts from the general requirements that have been defined in D7.1, and it specifies which of them relates to each of the components provided by the C3ISP partners. Finally, for each of these requirements, this deliverable specifies which is already satisfied by the current version of the tool and which, instead, is partially or not satisfied at all, thus requiring a maturation of the component in order to be adopted to implement the C3ISP Framework.

Table of contents

1. Introduction

1.1. Overview

This deliverable is the first output of WP8, "C3ISP Data Sharing, Analytics and Crypto Technology Maturation" due at M12. The WP8 main goal is the maturation of a set of tools and technologies that are provided by the C3ISP partners and that can be exploited for the implementation of the Information Sharing Infrastructure (ISI) and of the Information Analysis Infrastructure (IAI) of the C3ISP Framework. These tools and technologies cover most of the functionalities required in the C3ISP project. Namely, they: allow the definition of the Data Sharing Agreements (DSAs) to be paired with the information shared by the Prosumers; allow the enforcement of the DSA paired with the information when such information is used; provide the algorithms for extraction of new knowledge through the collaborative analysis of the information shared by the Prosumers (e.g., Cyber Threat Information, CTI); allow the data anonymization and the execution of homomorphic computing operations for collaborative analysis; and include specific technologies for security managed services and for the visualization of data analytics results.

For each of the tools provided by the C3ISP partners, this document provides a description of the main functionalities, specifying: i) whether it has been developed for a previous EU project; ii) the current maturation level through the Technology Readiness Level (TRL); iii) which of the components of the C3ISP Framework architecture (defined in D7.2 and reported in Section 2) will be developed on top of this, listing the related requirement by starting from the general ones that have been defined in D7.1 for the C3ISP Framework Architecture. Finally, for each of these requirements, this deliverable specifies which is already satisfied by the current version of the tool, which is partially satisfied, and which, instead, is not satisfied at all. In the last two cases, the tool requires a further maturation in order to be adopted in the C3ISP Framework.

1.2. Deliverable Structure

This document is structured as follows. Section 2, for the convenience of the readers, reports a high level view of the architecture of the C3ISP Framework. For a detailed description of the architecture, of its components and of the interactions among them please refer to D7.2. Each section from Section 3 to Section 8 covers one of the functionalities required in the C3ISP Framework, and lists the related tools provided by the C3ISP partner by reporting:

- A brief description of the tool and of the functionalities it provides;
- Which of the components of the C3ISP Framework Architecture can leverage on such tool for its implementation;
- Which of the requirements that have been defined in D7.1 can be satisfied through the tool;
- Which of the previous requirements are already satisfied by the current version of the tool and which requirements require a further maturation of the tool.

Finally, Section 9 draws the conclusions.

1.3. Requirements Naming Convention

In according with D7.1, this document reports the requirements of the components of the C3ISP Framework Architecture indexed with an identifier to make it easy to trace its fulfilment in the next phases of the project, as well as with a priority that follows the MoSCoW scale. In fact, the architecture will be designed to address the requirements by considering the different priorities assigned.

The requirements naming convention follows this format:

C3ISP-[ReqClass]-[Id]

Where [ReqClass] = **Fun** (Functional, further split in DS=Data Sharing and DA=Data Analytics, due to their importance), **Sec** (Security), **Ope** (Operational), **Per** (Performance), **Usa** (Usability), **Dev** (Development environment), **Tst** (Test Bed environment), **Com** (Component, further split in the subclasses shown in Table 1).

Table 1 – Acronyms used to identify the subclasses of the Com (Component) class

Subclass	Meaning
DE	DSA Editor
DM	DSA Mapper
CAE	Continuous Authorization Engine
OE	Obligation Engine
CDA	Collaborative Data Analysis
VSA	Visualization of Security Analysis
AA	Anonymization Algorithms
HE	Homomorphic Encryption
MSS	Managed Security Services

Some examples of requirement names are the following:

- C3ISP-Fun-DS-001 for C3ISP Functional requirement no. 001 for Data Sharing;
- C3ISP-Sec-001 for C3ISP Non-Functional requirement no. 001 for Security;
- C3ISP-Com-CAE-001 for C3ISP requirement no. 001 of the component: Continuous Authorization Engine.

In case subsections are present for each class of requirement, we adjust the numbering. E.g. Information Security Requirements→C3ISP-Sec-001; Regulatory Requirements→C3ISP-Sec-101, because both are part of the Security Requirement class and so we can accommodate new requirements during the on-going work of requirement elicitation and refinement.

The requirements naming convention also allows us to trace the requirements across different deliverables should this be necessary.

1.4. Definitions and Abbreviations

Term	Meaning
AES	Advanced Encryption Standard
C&C	Command and Control
C3ISP	Collaborative and Confidential Information Sharing and Analysis for Cyber Protection
CybOX	Cyber Observable eXpression
CI	Continuous Integration
CPE	Common Platform Enumeration
CSP	Cloud Service Provider
CTI	Cyber Threat Information
CVE	Common Vulnerability and Exposure
CWE	Common Weakness Enumeration
DAST	Dynamic Application Security Testing
DDoS	Distributed Denial of Service
DMO	Data Manipulation Operations
DSA	Data Sharing Agreement
FHE	Full Homomorphic Encryption
GDPR	General Data Protection Regulation (EU 2016/679), http://eur-lex.europa.eu/eli/reg/2016/679/oj
IAI	Information Analytics Infrastructure
IDE	Integrated Development Environment
IDS	Intrusion Detection System
IP	Internet Protocol
ISI	Information Sharing Infrastructure
LTS	Long-Term Support
LOWMC	Low Multiplicative Complexity (a family of block ciphers)
MITRE	The MITRE Corporation, https://www.mitre.org/
NFR	Non Functional Requirement
NVD	National Vulnerability Database
OASIS	Organization for the Advancement of Structured Information Standards
OWASP	Open Web Application Security Project
OpenC2	Open Command and Control
MoSCoW	Must have, Should have, Could have, and Won't have but would

	like
Multiplicative depth	Multiplicative depth is the maximum number of multiplicative gates between an input and an output of the circuit
PRINCE	64-bit block cipher with a 128-bit key optimized for low latency in hardware
Prosumer	An entity which is both a Producer and a Consumer of information, in particular of Cyber Threat Information
REST	Representational state transfer, a type of web services
RFI	Remote File Inclusion attack
SaaS	Software as a Service
SQLi	SQL injection attack
STIX	Structured Threat Information eXpression
TAXII	Trusted Automated eXchange of Indicator Information
TTP	Techniques, Tactics and Procedures
VCG	VisualCodeGrepper
VM	Virtual Machine
WAVSEP	Web Application Vulnerability Scanner Evaluation Project
XACML	eXtensible Access Control Markup Language
XSS	Cross-Site Scripting attack

2. High-Level Architecture

This section briefly recalls the high level C3ISP Framework reference architecture (shown in Figure 1) that has been defined at month 12. The main aim of this section is to give a quick overview of the main components of the architecture and a brief description of their main functionalities. A very detailed description of the components of the architecture, of their functionalities, of their interactions, and of the workflow of the main operations of the C3ISP Framework can be found in D7.2.

The main aim of recalling the C3ISP Framework reference architecture here is that in the following, we describe each of the tools that are being provided by the C3ISP partners, and for each of them this document specifies which of the components shown in Figure 1 can benefit of such tool.

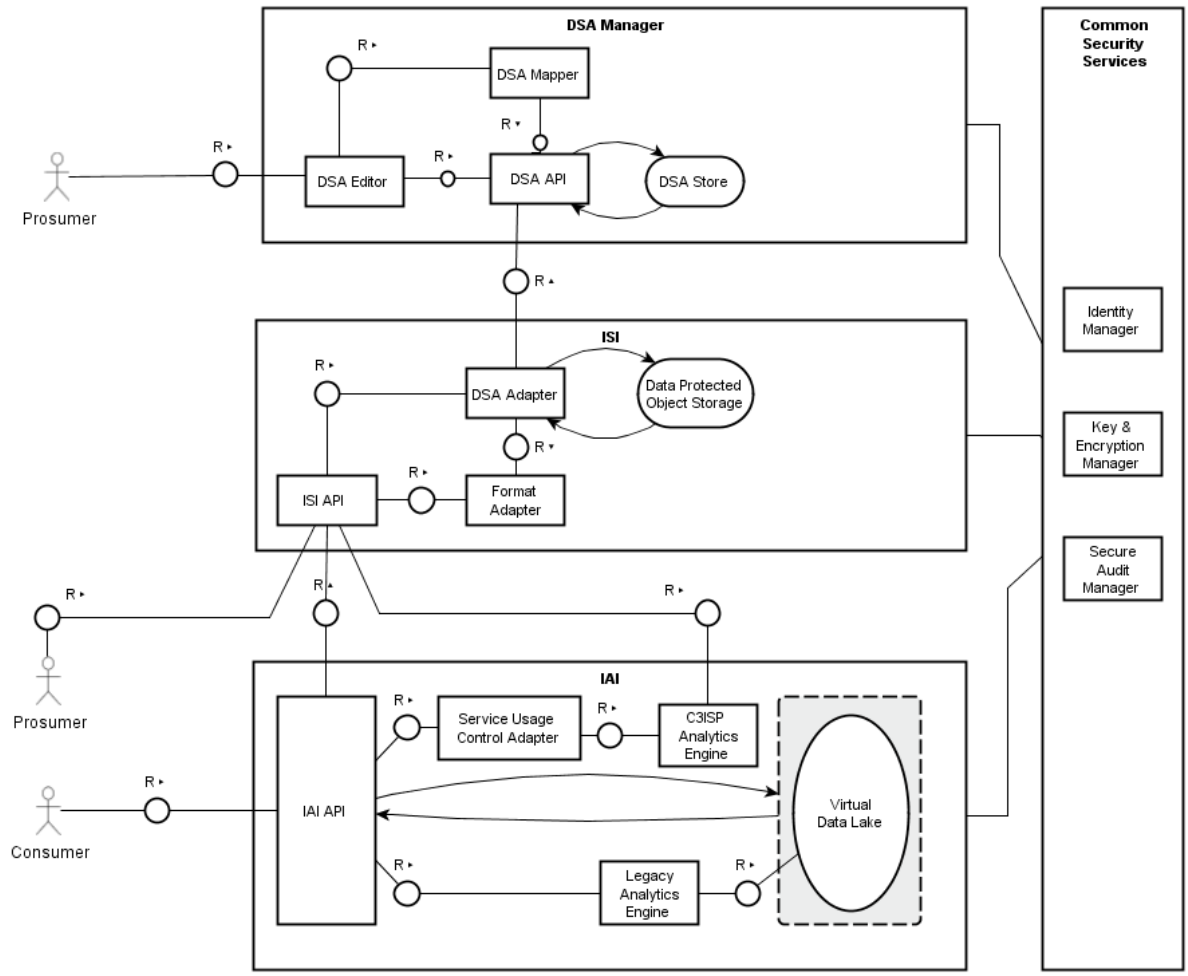

Figure 1: C3ISP high-level architecture – version 1 (Month 12), taken from D7.2

As shown in Figure 1, the C3ISP Framework architecture is composed of the following main **subsystems:**

The **Data Sharing Agreement (DSA) Manager** is the subsystem in charge of handling the DSA lifecycle, from the editing phase to its usage till its termination. Each Prosumer will define its own sharing and analytics rules to be enforced by the C3ISP Framework to handle his data. This subsystem is composed of three main components: **DSA Editor**: allows the Prosumers to create and edit their DSAs through a user friendly graphical interface; **DSA Mapper**: translates the DSAs written by the DSA Editor in their executable format, which is paired to the shared information to allow their enforcement at runtime; **DSA API**: defines the API for operating on DSA.

The **Information Sharing Infrastructure (ISI)** is the subsystem used by a Prosumer to provide data to the other Prosumers under the governance of an appropriate DSA. Several deployment models are defined for the ISI subsystems, depending on the requirement of the specific use case (refer to Sect. 3 of D7.2 for further details on the deployment models). The core feature of the subsystem is provided by the **DSA Adapter**, a component that is able to enforce the constraints defined in the DSA, in particular those related to access and usage control, and the manipulation of i) the data itself (through the DMOs, Data Manipulation Operations) before the execution of the collaborative analytics; ii) the results obtained from the analytics.

The **Information Analytics Infrastructure (IAI)** subsystem provides the interface to invoke analytics services executed by the **C3ISP Analytics Engine** on the shared data and stored through the ISI. The analytics execution result is computed considering the DSA paired with the data exploited in the analytics, which also apply to results that will have its own derived DSA (i.e., the DSA rules to enforce on the analytics result). The result is submitted again to the ISI to be both shared among the Prosumers and possibly used as an input for a new analytics service. In addition, to be able to manage legacy analytics engines, upon such requests, the subsystem instantiates a **Virtual Data Lake**, prepared to be used by the required legacy analytics service: this lake contains data that is prepared according to the DSA rules and usage constraints (e.g. part of the data could be anonymised, etc.). The Consumer actor is the person in charge of requesting the analytics services to be executed. Finally, the IAI subsystem also includes a **Service Usage Control Adapter**, which is a security component in charge of protecting the C3ISP services through the enforcement of proper access and usage control policies as well.

Finally, a bunch of integrated **Common Security Services (CSS)** are necessary to support the functions of the Framework. For instance, access and usage control need identities and profile information from an **Identity Manager** to evaluate their logic; a **Secure Audit Manager** service is necessary to trace the operations performed within the C3ISP Framework, and in particular those related to access and usage decisions and in general to guarantee the system accountability to show it operates as planned and as specified in the DSA rules; a **Key and Encryption Manager** is necessary to provide the confidentiality of the computations (in the case of homomorphic encryption) and the secrecy for the shared data.

3. Data Sharing Agreements

A DSA (Data Sharing Agreement) is a contract regulating the sharing of data among entities (organizations, individuals, etc.). A DSA can regulate the data sharing between two or more entities (bilateral/multilateral DSA). In the C3ISP scenario, DSAs are defined by the Prosumers when the C3ISP Framework is set up to regulate the usage of the information that they share in the data analytics processes. These DSAs concern both the information shared by the Prosumers as input of the analytics processes, and the data derived from the analytics computation. The DSA also regulates the storage of information. In particular, DSAs express constraints on the shared information in terms of:

- manipulation operations: operations required to pre-process the shared information before its usage, e.g. to anonymize the input data before processing them or the analysis results before sharing them with the other Prosumers (or even with a specific member); in particular, we consider anonymization and homomorphic computing operations;
- analytics operations: to permit or forbid the execution of an analytic service of the shared data depending on certain conditions.

Summarizing, DSAs define whether the shared information can be exploited to compute a given analytic operation, which manipulation operations must be performed on the shared information before the elaboration of the analytics engine, and which manipulation operations must be performed on the results before they can be shared with the other Prosumers.

3.1. DSA Editor

3.1.1. Tool/technology description and current state

The DSA Editor is a standalone Web application, accessible via an Internet browser, also available *as-a-service* to simplify its integration into other infrastructures. The tool allows the definition of Data Sharing Agreements (DSAs)in a collaborative way, proposing an interactive and guided approach: it takes advantage of the ontology [2] technology to support the user in the definition of the DSA policies (also called rules). When the user is creating a policy, the application suggests terms and actions on these terms, which are compliant with a predefined *vocabulary*, defining the semantics of the rules. The vocabulary is an ontology that describes the context in which the DSA should be used and it can be tailored to meet specific pilots' background, including terms and actions with some peculiarities, own of that context. Moreover, in order to simplify the definition of the rule for a non-technical user, the definition of the rules uses a very close natural language, based on a formal language, named *CNL – Controlled Natural Language* [1]. The user can define authorisations, prohibitions and obligations that are specific types of rules to express concepts respectively about what an entity **can**, **cannot**, and **must/must not** do; moreover, the tool provides support for defining temporal and space constructs and also for specifying the value for parametric fields involved in the policies (for example, specify the id of a user or an attribute of the data).

The tool and the language have been developed during two previous EU projects (FP7); in particular in the Consequence[1]project a first prototyped version was designed and developed

[1] http://www.consequence-project.eu/

and in the next Coco Cloud[2] project, it was refined and enhanced in order to support more complex scenarios.

In C3ISP Framework, the tool is part of the DSA Manager component and a prototype of it, released in Coco Cloud EU FP7 project, will be used as the foundation for building the DSA Editor by extending its core functionalities.

The current version of the DSA Editor defines an editing workflow process where a DSA may assume different states as the following state diagram (Figure 2) summarises:

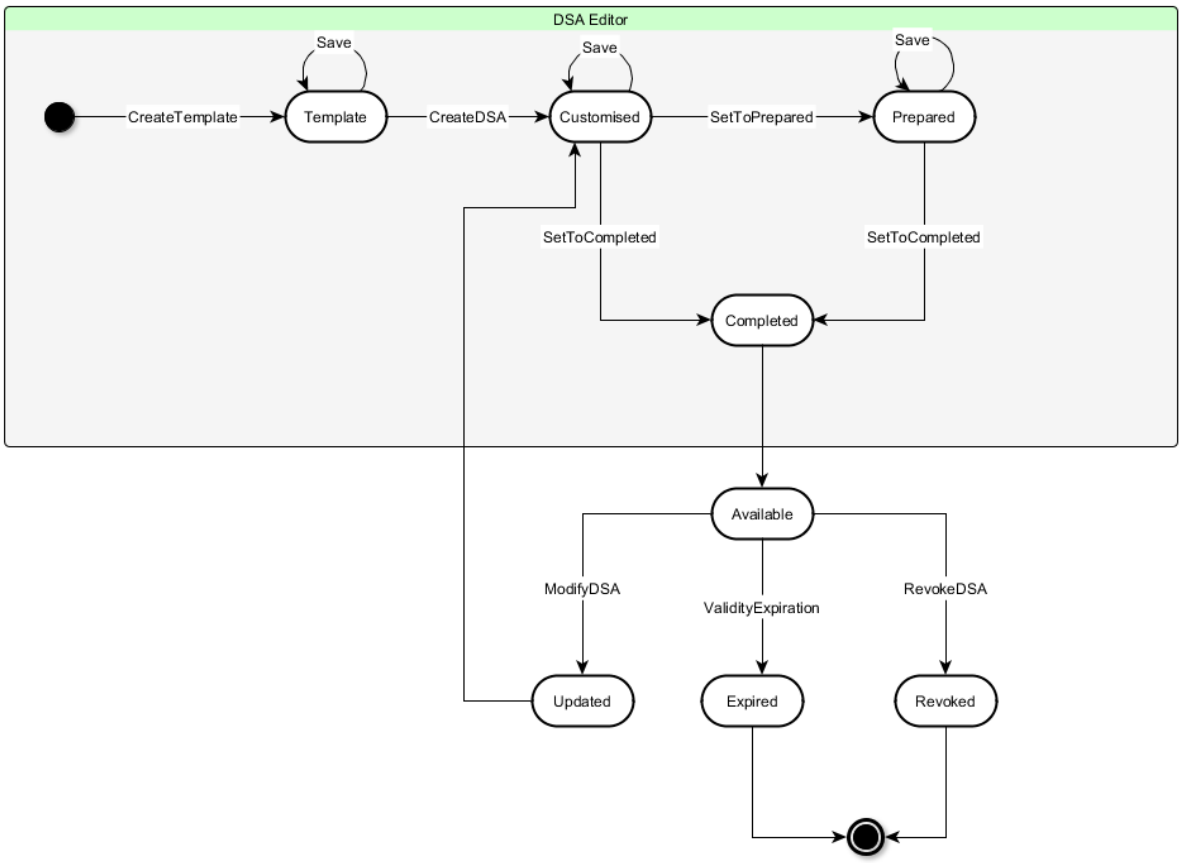

Figure 2: DSA State Diagram

At the beginning of the editing process, a DSA Template is created: it is the starting point for creating a DSA with rules for a specific application context, starting from a predefined vocabulary described in a standard semantic Web language (OWL – Ontology Web Language[3]); the status of the DSA is "Template" at the end of this phase. Then rules can be added to the Template and some fields can be instantiated for the specific data sharing: this moves the DSA into the "Customised" state. The DSA, in fact, has some metadata in addition to the rules: it is possible to specify the purpose of the data sharing, the type of data involved (data classification) and the parties participating to the agreement. Usually the Template is more generic and some of these metadata might be specified at the Template creation time because are common for a certain use case, while others might be defined during the DSA creation phase (i.e. the instantiation of a DSA from a Template). Additionally, the tool supports the definition of parametric fields which can be defined by the end user. In this case,

[2] http://www.coco-cloud.eu/
[3] https://www.w3.org/OWL/

the DSA is not completed but marked as "Prepared", in order to be finalized later. The final state for the DSA Editor is "Completed": it means that no parametric fields need to be filled in and the DSA can be marked as "Available", ready to be used for protecting data (i.e. to be paired with the data). This latter status is managed externally to the DSA Editor, as highlighted in Figure 2. Finally, the DSA can be "Revoked" (by the user or some other entity) or can be "Expired", according to a validity period, specified when created, beyond which it is no more applicable to the data. An Available DSA can be also further updated by coming back to the DSA Editor to be changed. The user can also specify policies to express what happens when the DSA is revoked or expires.

The diagram shows the transition between the supported states as in the Coco Cloud EU FP7 project prototype: the states will be reviewed and validated, in order to support the specific needs of the C3ISP Framework.

The access and the use of the tool follows a role-based access control (RBAC [3]) approach and three different roles are supported:

- A "*legal expert*", which is in title to create DSA Template, assuming s/he has a legal background and is able to create a generic DSA to be used as starting point for creating a DSA for a specific data sharing;
- A "*policy expert*", responsible for defining business policies and DSA metadata, assuming s/he is familiar with the context in which the DSA will be applied;
- The "*end user*" that can be – optionally– involved in the DSA for specifying user preferences or for accepting the conditions defined in the DSA, if it is defined on his or her personal data.

According to a specific role-capabilities matrix (described in the deliverable 4.2 of the project[4]) and the DSA state, the tool is adapted to the user; it means that some functionalities and some data are visible and changeable only having a certain role in a certain status of the DSA. The supported roles and the RBAC matrix should be reviewed according to the C3ISP Framework needs and the pilot specific scenarios.

Finally, the internationalization (I18N) is supported: currently the tool is available in English (the default), Spanish, Italian and French, but additional languages can be supported with little effort.

The current Technology Readiness Level (TRL) of the DSA Editor is 4; we plan to reach a TRL level 6 adding the following improvements:

- Define new vocabularies for the pilots' context, with particular attention to the legal aspects related to the security of the information sharing;
- Support the definition of Data Manipulation Operations (pre/post-processing rules) [it copes the requirementsC3ISP-Fun-DS-011 and C3ISP-Fun-DA-002 described in D7.1];
- Support for data analytics operations [it meets C3ISP-Fun-DA-001 requirement];
- Support notifications for results [it meets C3ISP-Fun-DS-009 requirement];
- Improvements on the DSA Editor usability [it meets C3ISP-Usa-002 requirement].

[4] http://www.coco-cloud.eu/content/d42-first-dsa-management-infrastructure

3.1.2. Tool/technology requirements

Table 2 – DSA Editor Requirements

ID	Priority	Requirement	In order to fulfil D7.1 Requirement(s)
C3ISP-Com-DE-001	MUST	Define Data Sharing Agreements between parties that want to exchange CTI data and analytics results	C3ISP-Fun-DS-001 C3ISP-Fun-DA-001
C3ISP-Com-DE-002	MUST	Specify two or more parties involved in the data sharing	C3ISP-Fun-DS-003 C3ISP-Fun-DS-014
C3ISP-Com-DE-003	MUST	Specify policies that regulate the data (CTI) and analytics results sharing	C3ISP-Fun-DS-005
C3ISP-Com-DE-004	MUST	Define access and usage control policies on data	C3ISP-Fun-DS-006 C3ISP-Fun-DS-007 C3ISP-Sec-104
C3ISP-Com-DE-005	SHOULD	Define parametric policies	C3ISP-Fun-DS-008 C3ISP-Fun-DS-012
C3ISP-Com-DE-006	SHOULD	Define policies on data manipulation operation	C3ISP-Fun-DS-011 C3ISP-Fun-DA-002 C3ISP-Ope-004
C3ISP-Com-DE-007	COULD	The definition of policies is based on ontology (OWL standard) and uses a natural controlled language translated in a XACML [4] standard expression	C3ISP-Fun-DS-013 C3ISP-Ope-102 C3ISP-Ope-103
C3ISP-Com-DE-008	MUST	Specify a validity period for the DSA after which the DSA expires	C3ISP-Sec-102

C3ISP-Com-DE-009	MUST	Allow to revoke or update a DSA	C3ISP-Sec-102
C3ISP-Com-DE-010	MUST	The tool is available as a standalone Web application or as a service	C3ISP-Ope-001 C3ISP-Ope-002 C3ISP-Ope-003
C3ISP-Com-DE-011	MUST	The tool is interactive and user friendly: it supports the user in the definition of the policies leveraging on a predefined vocabulary (ontology)	C3ISP-Usa-002

3.1.3. Requirements analysis

Table 3 – DSA Editor Requirements Status

ID	MET	Description
C3ISP-Com-DE-001	PARTIALLY	The tool supports the definition of Data Sharing Agreements between parties. For defining policies on CTI data and analytics results we need to define a vocabulary (based on an ontology) for expressing these constructs and operations on them.
C3ISP-Com-DE-002	YES	The tool supports the definition of multi-lateral Data Sharing Agreements that is two or more parties can be specified in the agreement.
C3ISP-Com-DE-003	NO	The definition of a specific vocabulary is needed for expressing policies and legal constraints on CTI and analytics results.
C3ISP-Com-DE-004	YES	It allows defining access and usage control policies.
C3ISP-Com-DE-005	YES	It allows defining policy that requires a parametric value for some fields.

C3ISP-Com-DE-006	NO	The definition of a specific vocabulary is needed for expressing data manipulation operation.
C3ISP-Com-DE-007	YES	The tool already uses Web ontologies for defining its vocabularies.
C3ISP-Com-DE-008	YES	It already allows for a DSA validity period.
C3ISP-Com-DE-009	PARTIALLY	The tool only allows updating DSA, but has to be extended to support DSA revocation.
CEISP-Comp-DS-010	YES	The tool is a Web application available also *as a service*.
C3ISP-Com-DE-011	YES	The tool allows interactive and guided policy authoring.

3.2. DSA Mapper

3.2.1. Tool/technology description and current state

The DSA Mapper aims to transform the set of rules of a DSA, defined through the DSA Editor and specified in Controlled Natural Language (CNL), into policies that can be automatically enforced. As a matter of fact, once a DSA has been edited and finalised, a translation from CNL to an executable format is needed to allow the enforcement of the policy by the usage control engine, implemented by the DSA Adapter in the C3ISP Framework. These enforceable policies are expressed exploiting an extension of the XACML language which supports with Usage Control features. XACML is a well known standard for writing attribute based access control policies, and also define a reference architecture for the enforcement of such policies. For a detailed description of the XACML standard please refer to [4]. In particular, we plan to maturate the UPOL language, a XACML-based security policy language developed within the Coco Cloud EU FP7 project.

Due to the nature of DSAs and the possible heterogeneity of the parties among whom the agreement is going to be established, the set of policies to be enforced can be extremely rich including aspects derived from legislation, from organisational policies and from security requirements, all of which have been given without making specific assumptions about the

enforcement model. The problem is therefore very difficult, if not intractable in the general case.

This component has been designed and developed within the Coco Cloud EU FP7 project in order to provide a mapper function that is suitable to cope with all vocabularies and DSA provided by the use cases. In the C3ISP Framework, the DSA Mapper is part of the DSA Manager component, and the DSA Mapper prototype released in Coco Cloud EU FP7project will be extended to cover the C3ISP requirements. The current Technology Readiness Level (TRL) of the DSA Mapper is 4, and within the C3ISP Framework we plan to maturate it to reach TRL 6.

The DSA Mapper exposes two main functionalities:

- An automatic refinement of each term of pilot vocabularies in such a way that they result understandable for the enforcement component: the mapper implements a translation function able to recognize entities of each Pilot vocabulary used to edit DSA rules. A vocabulary is composed by Terms and Properties on terms. Each term of the vocabulary corresponds to an OWL object in the vocabulary. However, to be correctly interpreted at enforcement level, each term needs some *additional information*, such as if the term has to be evaluated at access request time only (Access Control) or it has to be continuously evaluated during the access time (Usage Control).
- A mapping of both CNL rules syntax and semantics: this function maps the CNL constructs into UPOL constructs (which are derived from XACML ones), thus it is possible to identify in each CNL statement the main UPOL elements:
  - A *subject* element is the entity requesting the access. A subject has one or more attributes.
  - The *resource* element is a data, service or system component. A resource has one or more attributes.
  - An *action* element defines the type of access requested on the resource. Actions have one or more attributes.
  - An *environment* element can optionally provide additional information.

3.2.2. Tool/technology requirements

Table 4 – DSA Mapper Requirements

ID	Priority	Requirement	In order to fulfil D7.1 Requirement(s)
C3ISP-Com-DM-001	MUST	The Mapper functionality grants Prosumers that the translation of data sharing constraints is compliant and consistent from the high level to the low level specification.	C3ISP-Fun-DS-003

C3ISP-Com-DM-002	MUST	The Mapper output consists of access control policy directly enforceable	C3ISP-Fun-DS-006
C3ISP-Com-DM-003	MUST	The Mapper output consists of usage control policy directly enforceable	C3ISP-Fun-DS-007
C3ISP-Com-DM-004	SHOULD	The Mapper should be able to properly parse the vocabulary of each pilot in such a way that all the contextual condition and attributes will be correctly interpreted at execution time	C3ISP-Fun-DS-008
C3ISP-Com-DM-005	MUST	The Mapper must translate as post-obligations policies related to notifications that are triggered once the analytics service result is available	C3ISP-Fun-DS-009
C3ISP-Com-DM-006	MUST	The Mapper must translate as pre-obligations policies related to data manipulation operations that must be performed before the execution of the analytic operations	C3ISP-Fun-DS-010
C3ISP-Com-DM-007	COULD	The Mapper could be able to interpret rules about the risk of data sharing in a proper way	C3ISP-Fun-DS-012
C3ISP-Com-DM-008	COULD	The Mapper could be able to translate each data sharing policy into a standard enforceable language, such as, XACML	C3ISP-Fun-DS-013

3.2.3. Requirements analysis

Table 5 – DSA Mapper Requirements Status

ID	MET	Description
C3ISP-Com-DM-001	YES	The current version of the Mapper already grants Prosumers that the translation of data sharing constraints is compliant and consistent from the high level to the low-level specification.
C3ISP-Com-DM-002	YES	The current version of the Mapper provides as output directly enforceable access control policies.
C3ISP-Com-DM-003	YES	The current version of the Mapper provides as output directly enforceable usage control policies.
C3ISP-Com-DM-004	PARTIALLY	The Mapper needs to learn every new vocabulary in order to correctly interpret it.
C3ISP-Com-DM-005	PARTIALLY	The Mapper needs to be upgraded to be able to translate as post-obligations policies related to notification that are triggered once the analytics service result is available.
C3ISP-Com-DM-006	PARTIALLY	The Mapper needs to be upgraded to be able to translate as pre-obligations policies related to data manipulation operation.
C3ISP-Com-DM-007	NO	The Mapper should be maturated to interpret rules about the risk of data sharing in a proper way
C3ISP-Com-DM-008	YES	The current version of the Mapper is already able to translate policies from CNL into a XACML-based language.

4. Data collection and Usage Enforcement

The evaluation and the enforcement of the Access and Usage Control policies expressed through the DSAs paired with the pieces of data require a proper component of the architecture able to collect the data required for the evaluation and to perform the DSA evaluation itself in order to decide whether an access to that piece of data should be granted or denied. Since the DSA embeds Usage Control policies, beyond traditional access control tasks, this component must be also able to perform the continuous evaluation of the policy, in order to interrupt the usage of the data as soon as the policy is violated. This also implies being able to monitor all the attributes which are evaluated in the policy in order to promptly detect any change. Moreover, the component must be able also to execute obligations. Some of these obligations will be the Data Manipulation Operations (DMO), which transforms the data for privacy reasons.

As an example, the anonymization of some fields of the data in order to hide the subject to whom the data are related to, is one of the DMOs taken into account by C3ISP. Other obligations are, instead, operations that are performed on the data piece such as deletion. This component is reported in the architecture defined in D7.2 as "DSA Adapter". The DSA adapters are placed in a way such that the information shared by the Prosumers must transit through them before being elaborated by the C3ISP Analytics Service and results must transit through them before being sent back to the Prosumers. The DSA Adapter is actually a Usage Control service that receives the information and the DSA, evaluates the usage policy of the DSA, and enforces this policy by either executing the manipulation operations expressed by the obligations and forwarding the manipulated information to the next component of the architecture in case the requested analytics operation is permitted, or not releasing the information to the next component of the architecture when the policy evaluation denies the execution of the operation. Besides denying the execution of an analytics operation, the DSA Adapter could also ask to interrupt the execution of an ongoing operation when the corresponding policy is not satisfied any more.

4.1. Continuous Authorization Engine Requirements

4.1.1. Tool/technology description and current state

The **Continuous Authorization Engine** is an authorization engine which supports both traditional access control (i.e., the authorization process performed at request time) and continuous access control (an enhanced feature introduced by the Usage Control model, UCON, defined in [5]). The traditional access control phase (called *preAuthorization* in UCON) enforces the security policy when the access request is received, in order to check whether the subject who requests the access actually holds the right to perform the action on the object. The continuous authorization phase (called *onAuthorization* in UCON), instead, checks that the right to perform the action continuously holds during the execution of the action itself, in order to take a countermeasure (such as interrupting or suspending the execution of the action) as soon as this right expires.

This component has been developed within the Coco Cloud EU FP7 project (GA #610853), and its current Technology Readiness Level is 4, and within the C3ISP project the component will be maturated to reach TRL 7. With reference to the high level C3ISP Framework reference architecture, this component will be exploited as basis for the implementation of the DSA Adapter and of the Service Usage Control Adapter.

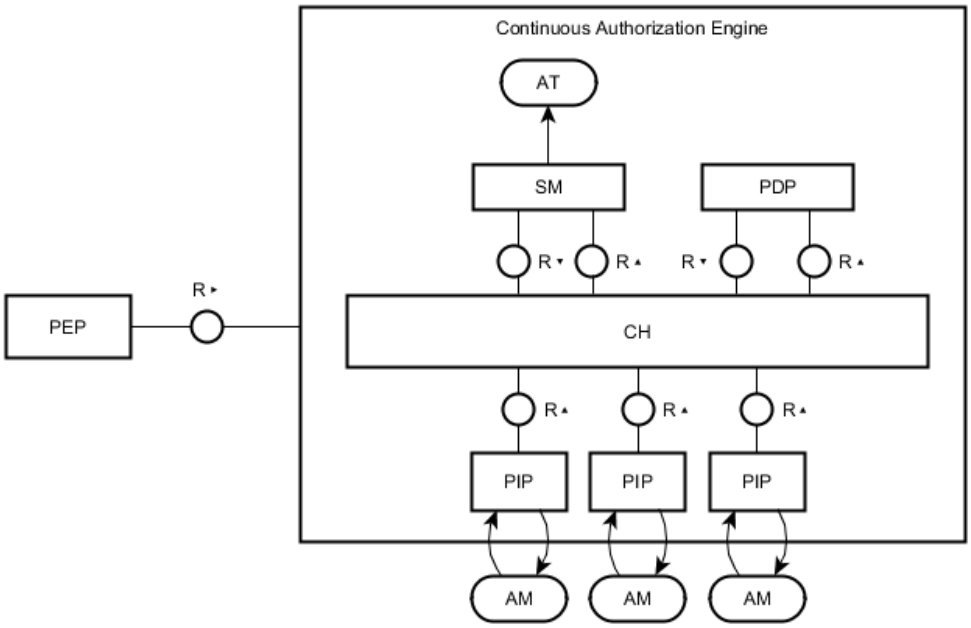

Figure 3: Components of the Continuous Authorization Engine

The architecture of the Continuous Authorization Engine, shown in Figure3, is an extension of the XACML reference architecture, described in [4]. In particular, the components of the Continuous Authorization Engine are the following:

- Policy Enforcement Point (PEP) is the component that intercepts the requests to perform security relevant actions, triggers the decision process and enforces the results. In C3ISP the role of the PEP is played by the Event Handler (see Sect. 4.1 of D7.2), which is in charge of coordinating the components involved in the decision process;
- Context Handler (CH) is the entry point of the Continuous Authorization Engine and it manages the protocol for communicating with the PEP (i.e., the Event Handler). This protocol is defined by a subset of the usage control actions: *tryaccess, permitaccess, denyaccess, revokeaccess*, and *endaccess* (see [4] for further details). The CH also coordinates the internal components of the Continuous Authorization Engine for the execution of the policy evaluation process, as described in the following of this section;
- Session Manager (SM) is the components which is responsible for keeping track of the ongoing usage sessions, i.e., of the access that are currently in progress, and it exploits an Access Table (AT) to store the meta-data regarding these sessions. It is the key component of the continuous authorization phase, and it represents an extension with respect to the XACML reference architecture;
- Policy Decision Point (PDP) is the component which evaluates security policies and produces the access decision. In our framework, the PDP evaluates standard XACML policies because the usage control specific features are managed by the CH and by the SM;
- Attribute Managers (AMs) are components which manage attributes, allowing to retrieve and to update their current values for running the policy evaluation process.

Each specific scenario in which the Continuous Authorization Engine is exploited has its own attributes, which describe the security relevant features of subject, resource and environment for that specific scenario. AMs could be local, i.e., they run on the same machine as the Continuous Authorization component or remote, i.e., they could run on external servers. In some scenarios, these servers could be even located in other domains run by third-parties. Each AM has its own protocol, and the Continuous Authorization Engine exploits the Policy Information Pints as plug-ins for interacting with the required AMs;

- Policy Information Points (PIPs) are the interfaces exploited by the Continuous Authorization Engine for interacting with the Attribute Managers which manage the attributes required to evaluate the Usage Control Policy. For instance, some attributes are retrieved from other C3ISP components, other attributes are directly managed by the Continuous Authorization Engine component, while other attributes are managed by third organizations. PIPs are required because distinct Attribute Managers typically support different protocols for interacting with them, and PIPs mimic a plug-in architecture to let the Continuous Authorization Engine be as flexible as possible enabling the integration with any kind of Attribute Managers. In particular, the proposed architecture includes a set (chain) of PIPs which provide the same interface to the CH (*retrieve, subscribe/unsubscribe* and *update*), while each PIP implements the specific protocol to interact with a given Attribute Manager and the specific algorithm to perform the requested operation and to provide the required information. Hence, to integrate the Continuous Authorization Engine within the C3ISP Framework is necessary to implement the set of PIPs which allows the Continuous Authorization Engine to retrieve, subscribe and update the set of attributes defined in the DSAs of the Prosumers.

4.1.2. Tool/technology requirements

Table 6 – Continuous Authorization Engine Tool Requirements

ID	Priority	Requirement	In order to fulfil D7.1 Requirement(s)
Com-CAE-001	MUST	The tool must be able to grant to the Prosumers the control over the sharing of data enforcing the Usage Control policies specified in the DSA	C3ISP-Fun-DS-003
C3ISP-Com-CAE-002	MUST	The tool must be able to evaluate the DSA paired with the data when an access is requested in order to decide whether to grant the access or not	C3ISP-Fun-DS-006 C3ISP-Fun-DA-001

C3ISP-Com-CAE-003	MUST	The tool must be able to continuously evaluate the DSA paired with the data when an access is in progress in order to decide whether to revoke the access	C3ISP-Fun-DS-007
C3ISP-Com-CAE-004	SHOULD	The tool should be able to retrieve and use the contextual information for the evaluation of the DSA paired with the data	C3ISP-Fun-DS-008
C3ISP-Com-CAE-005	MUST	The tool must be able to evaluate the DSA to support the Obligation Engine (see Section 4.2) to decide when notifications must be sent	C3ISP-Fun-DS-009 C3ISP-Fun-DA-012
C3ISP-Com-CAE-006	SHOULD	The tool should be able to evaluate the DSA to support the Obligation Engine (see Section 4.2) to decide which data manipulation operation must be performed on the data before being shared	C3ISP-Fun-DS-011
C3ISP-Com-CAE-007	COULD	The tool could be able to take into account also the risk of data sharing to make the decision	C3ISP-Fun-DS-012
C3ISP-Com-CAE-008	COULD	The tool could support standard policy description languages, such as XACML, or their extensions	C3ISP-Fun-DS-013
C3ISP-Com-CAE-009	MUST	The tool must be able to evaluate the DSA to decide which data manipulation operation must be performed on the results before being returned to the Prosumers	C3ISP-Fun-DA-002
C3ISP-Com-CAE-010	MUST	The tool must be able to exploit the user ID provided by as result of the authentication process	C3ISP-Sec-001
C3ISP-Com-CAE-011	MUST	The tool must support multi-tenancy	C3IP-Ope-002
C3ISP-Com-CAE-012	MUST	The tool must not introduce significant delay in the analytics operations	C3ISP-Per-001

4.1.3. Requirements analysis

Table 7 – Continuous Authorization Engine Tool Requirements Status

ID	MET	Description
C3ISP-Com-CAE-001	YES	The current version of the tool is able to enforce both the and access and usage control policies embedded in the DSA paired with the data shared with the Prosumers which define the controls to be performed to regulate the data sharing
C3ISP-Com-CAE-002	YES	The current version of the tool is able to evaluate the DSA paired with the data, to perform the decision process and determine whether to grant the access or not.
C3ISP-Com-CAE-003	YES	The current version of the tool is able to allow usage control on the shared data by continuously evaluating the DSA paired with the data, and by performing the decision process when the access context is changed in order to determine whether to interrupt the access in progress or not.
C3ISP-Com-CAE-004	PARTIALLY	The tool can be easily configured and extended in order to retrieve the contextual information and to process them in order to be exploited for the evaluation of the DSA paired with the data.
C3ISP-Com-CAE-005	YES	The tool is able to evaluate the DSA to support the Obligation Engine to decide whether an obligation expressed in the DSA (e.g., sending a notification) must be performed before, during, or after the end of the operation into.
C3ISP-Com-CAE-006	YES	The tool is able to evaluate the DSA to support the Obligation Engine to decide whether an obligation expressed in the DSA (which can be used to express data manipulation operations) must be performed before, during, or after the end of the operation.
C3ISP-Com-CAE-007	PARTIALLY	A further component which compute the risk of data sharing must be developed. This component should act as AM. The tool must be configured to consider the risk of data sharing as a new attribute.
C3ISP-Com-CAE-008	YES	The language currently supported by the tool is UPOL, an extension of XACML developed within the Coco Cloud EU FP7 project.
C3ISP-Com-CAE-009	YES	The tool allows to enforce the obligations expressed in the DSA, which can be used to express data manipulation operations on the result of the operations.

C3ISP-Com-CAE-010	PARTIALLY	The tool is able to exploit the user ID in the access and usage decision process by representing it as an attribute of the user.
C3ISP-Com-CAE-011	PARTIALLY	The tool must be instrumented to support multi-tenancy.
C3ISP-Com-CAE-012	PARTIALLY	The impact of the policy enforcement on the performance of the analytics operations depends on several factors. The performance will be analysed for each use case and proper optimization will be proposed where necessary.

4.2. Obligation Engine Requirements

4.2.1. Tool/technology description and current state

The Obligation Engine is a component that allows the execution of pre-determined operations when specific conditions occur. Such operations are defined as *Usage Control Obligations* that are included in the enforceable policy representing the DSA associated to a specific data. The enforceable policies are expressed in UPOL language, a security policy language developed in the Coco Cloud EU FP7 project (GA #610853).

The structure of obligations can be described as follows:

- Usage Control Obligation = do *Action* when *Trigger*

Where *Trigger* is defined by:

- *Trigger* = *Event* AND *Condition*

Therefore, an obligation results in the execution of a particular action, when a specific event occurs but only if a condition is verified.

Usage Control Obligations, therefore, may be defined by specifying the desired combination of actions and triggers. An initial set of triggers and actions is currently supported by the actual implementation of the Obligation Engine, and through interactions with the pilots' owners, this set may be extended and adapted as needed.

A number of triggers and actions are currently supported, from time-based to continuous obligations that are coupled with the Continuous Authorization Engine.

Internally, the Obligation Engine is constituted by a number of modules, as depicted in Figure 4:

- The **Obligation Handler**, in charge of processing (and persisting) the obligation definitions coming from the policies;
- The **Trigger Engine**, that permits to support multiple types of triggers by implementing their specific business logic;
- The **Action Engine**, that, similarly to the Trigger Engine, is responsible for materialising the actions in obligations;
- The **Event Handler,** that interacts with the Event Handler of the DSA Adapter to filter and process the events relevant for the Obligation Engine.
- For performance reasons, the Obligation Engine caches the obligations when new resources enter its control domain. This cache is represented in the picture with the name **Obligations for managed resources.**

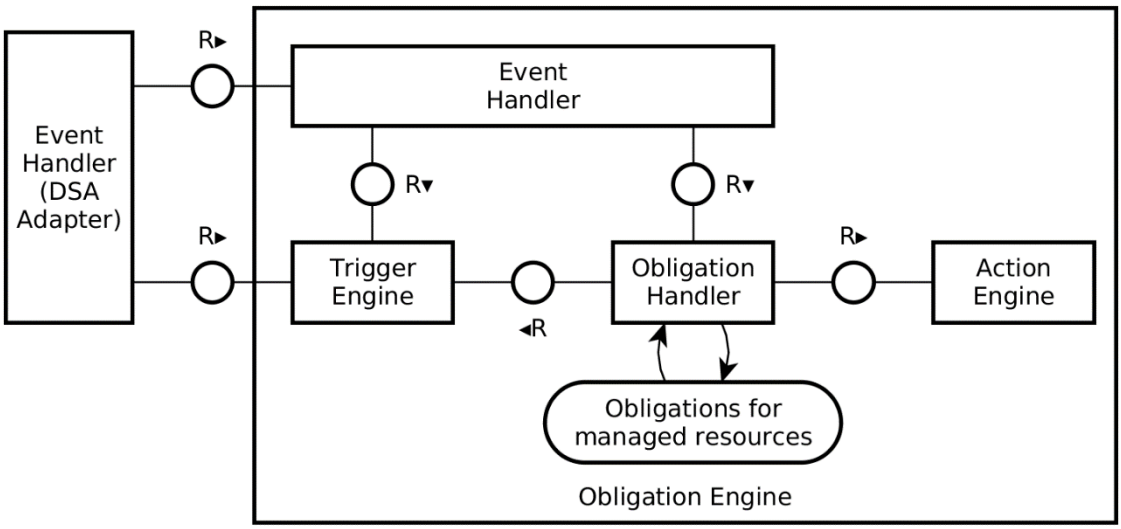

Figure 4: Obligation Engine Architecture

This component has been developed within the Coco Cloud EU FP7 project (GA #610853) and the current Technology Readiness Level is 4, and it is planned to reach TRL 7 within the C3ISP project.

4.2.2. Tool/technology requirements

Table 8 – Obligation Engine Requirements

ID	Priority	Requirement	In order to fulfil D7.1 Requirement(s)
C3ISP-Com-OBE-001	MUST	The component must be able to grant the enforcement of the obligations specified in the DSA during the data sharing process.	C3ISP-Fun-DS-003, C3ISP-Fun-DS-007, C3ISP-Sec-101, C3ISP-Sec-103
C3ISP-Com-OBE-002	MUST	The component must be able to grant the enforcement of the obligations specified in the DSA associated to the data analysis process.	C3ISP-Fun-DS-003, C3ISP-Fun-DS-007
C3ISP-Com-OBE-003	MUST	The component must be able to execute a number of actions of interest for the C3ISP pilots, as part of its supported obligation actions.	C3ISP-Fun-DS-009, C3ISP-Fun-DA-012

C3ISP-Com-OBE-004	SHOULD	The component should collaborate to the creation of evidences (e.g. audit logs, emails, etc.) as part of its obligation actions when specific events are detected.	C3ISP-Fun-DS-010
C3ISP-Com-OBE-005	MUST	The component must interact with DMO Engine in order to execute pre-processing rules dealing with data manipulation operations.	C3ISP-Fun-DS-011, C3ISP-Fun-DA-003
C3ISP-Com-OBE-006	COULD	The component can contribute to the risk computation in risk-aware decision making processes.	C3ISP-Fun-DS-012
C3ISP-Com-OBE-007	MUST	The component must interact with DMO Engine in order to execute post-processing rules dealing with data manipulation operations.	C3ISP-Fun-DA-002, C3ISP-Fun-DA-003
C3ISP-Com-OBE-008	SHOULD	The component should not introduce significant delays in the C3ISP operating process.	C3ISP-Per-001

4.2.3. Requirements analysis

Table 9 – Obligation Engine Requirements Status

ID	MET	Description
C3ISP-Com-OBE-001	PARTIALLY	Additional effort is needed in order to support the newly defined events connected to the C3ISP analytics service invocation workflow. However, a number of events connected with sharing operations are already supported.
C3ISP-Com-OBE-002	NO	Additional effort is needed in order to support the newly defined events connected to the C3ISP analytics service invocation workflow.
C3ISP-Com-OBE-003	PARTIALLY	Additional effort is needed in order to support the newly defined actions connected to the C3ISP analytics service invocation workflow. However, a number of actions connected with sharing operations are already supported.

C3ISP-Com-OBE-004	YES	Certain audit obligations (generation of audit trails, email sending etc.) are already supported by the current implementation. Adaptations or further extensions may be needed, though.
C3ISP-Com-OBE-005	PARTIALLY	The actual connection with the DMO Engine does not exist in the present prototype. However, the Action Engine is designed to support interactions with external services.
C3ISP-Com-OBE-006	PARTIALLY	The Obligation Engine can trigger any necessary operation needed to estimate the risk associated to an operation, when this process is modelled as a supported obligation. Concrete measures have to be identified and thus implemented, though.
C3ISP-Com-OBE-007	PARTIALLY	The actual connection with the DMO Engine does not exist in the present prototype. However, the action engine is designed to support interactions with external services.
C3ISP-Com-OBE-008	PARTIALLY	The current Obligation Engine implementation needs to be evaluated for its performances, small adaptations may be necessary in order to achieve scalability for performance enhancements.

5. Collaborative Data Analytics

5.1. Tool/technology description and current state

The Collaborative Data Analytic is a set of methods and tools offered by the C3ISP Framework to extract additional knowledge from information shared by Prosumers. These tools include computational intelligence functions, in particular clustering and classification algorithms, data aggregation and correlation functions, statistical analysis tools, and data visualization primitives. They are either implemented through open source libraries for machine learning, in particular WEKA[5]and Scikit-Learn[6], libraries and tools for Big Data analysis mainly derived from the Apache Hadoop software suite, such as Spark, Flink and Mahout, or internally implemented by project partners, either as output of research activities in data analysis, or as a commercial product.

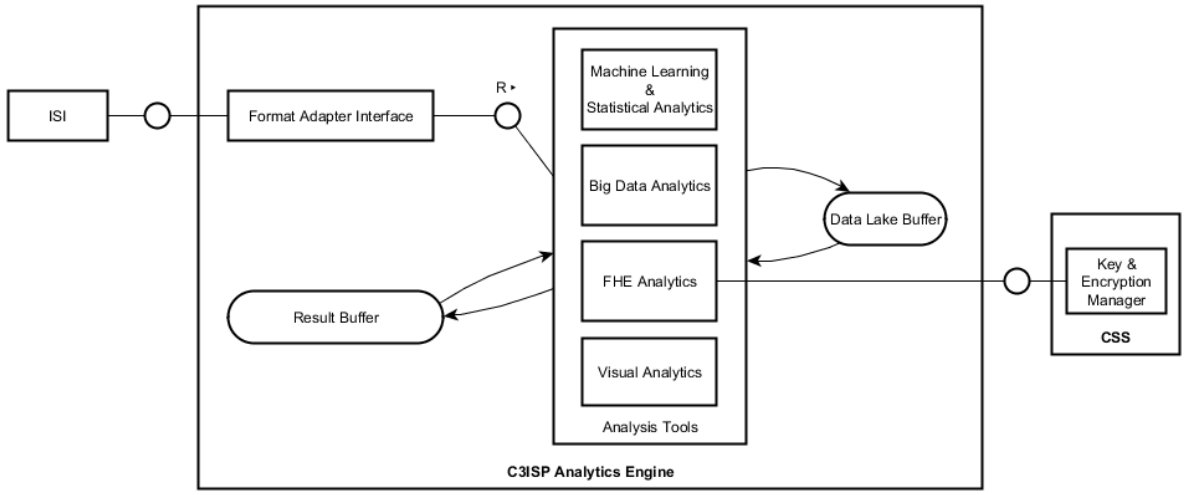

Figure 5: Data Analysis Infrastructure

The infrastructure depicted in Figure 5 is based on an Analysis Tools set and three functional components, which will handle the data flow in the engine. The Format Adapter Interface, is an interface to the Format Adapter component already described in the previous section, which will prepare the format of information, from the structured CTI format, to the one needed by the required analysis algorithm. The Data Lake buffer is a temporary storage in which pieces of information used for analysis are stored. The specific structure for storage in the buffer will depend from the desired analysis. The result buffer is a temporary container for the final results, acting thus both as a complimentary component to the data lake buffer, and to store the final results before they are sent to the Format Adapter.

The analysis tools are divided in the aforementioned sets. All the analytics functions considered are compatible with the Data Manipulation Operation described in the ISI and their analysis is completely under the control of the C3ISP Framework. For what concerns maturation, the Analysis Tools component is currently at TRL 4 and we expect to maturate it at TRL 6, by improving the usability, flexibility and robustness of the component, in particular for what concerns the input management and data handling.

[5] http://www.cs.waikato.ac.nz/ml/weka/
[6] http://scikit-learn.org/stable/

5.2. Tool/technology requirements

Table 10 – Tool Requirements

ID	Priority	Requirement	In order to fulfil D7.1 Requirement(s)
C3ISP-Comp-CDA-001	MUST	The Collaborative Data Analytics component provides a programming interface for supporting the analysis on the data shared by the Prosumers in compliance with the associated DSA policies.	C3ISP-Fun-DA-005
C3ISP-Comp-CDA-002	SHOULD	The Collaborative Data Analytics component supports different categories for analytics operations results, i.e. threat types, threat risks, threat origins, threat costs, regulatory and compliance concerns.	C3ISP-Fun-DA-014
C3ISP-Comp-CDA-003	COULD	The Collaborative Data Analytics component supports the provisioning of analytics operation results in form of actionable items (e.g. security patches, recommended configurations, fixes, etc.).	C3ISP-Fun-DA-015
C3ISP-Comp-CDA-004	SHOULD	The Collaborative Data Analytics component is able to process anonymised or homomorphically encrypted CTI shared with it by the Prosumers.	C3ISP-Sec-006
C3ISP-Comp-CDA-005	MUST	The representation of analytics results is effective and efficient for the end user.	C3ISP-Usa-004
C3ISP-Comp-CDA-006	SHOULD	Minimum security level is at least 80 bits (security strength). See discussion on Full Homomorphic Encryption (FHE) in section 2.2.1.1 of D7.1.	C3ISP-Sec-007

C3ISP-Comp-CDA-07	MUST	Maximum security level is at most 128 bits (computational efficiency), for real world scenarios. See discussion on FHE in section 2.2.1.1 of D7.1.	C3ISP-Sec-008
C3ISP-Comp-CDA-08	MUST	The homomorphic encryption uses randomization methods (see section 2.1.3 of D7.1). It is required to have semantic security. That is, it should be hard to distinguish between the encryption of any two messages, even if the public key is known to the attacker and even if the two messages are chosen by the attacker (chosen plaintext attacks). (In return, cipher-text size is greater than plaintext size). See discussion on FHE in section 2.2.1.1 of D7.1.	C3ISP-Sec-009

5.3. Requirements analysis

Table 11 – Tool Requirements Status

ID	MET	Description
C3ISP-Comp-CDA-001	PARTIALLY	The Collaborative Data Analytics component supports several kinds of analysis on the data shared by the Prosumers, in compliance with the associated DSA policies. These analysis are invoked though the IAI API.
C3ISP-Comp-CDA-002	NO	At this level of project maturation, this requirement has not been elaborated yet.
C3ISP-Comp-CDA-003	PARTIALLY	Pilots are designed to execute collaborative functions to provision C3ISP on security issues, such as vulnerabilities, malicious IPs and so on.
C3ISP-Comp-CDA-004	PARTIALLY	The Collaborative Data Analytics component will be able to work on anonymised or homomorphically encrypted CTI, adopting the technique described in Section 7
C3ISP-Comp-CDA-005	PARTIALLY	The Collaborative Data Analytics component will be developed to be provide an effective and efficient representation of the results of the C3ISP analytics for the end-user. For the legacy analytics, if required, a result format adapter could be defined.

C3ISP-Comp-CDA-006	YES	The tool provided for FHE (Cingulata, see Section 7.2) supports the required minimum security level (at least 80 bits).
C3ISP-Comp-CDA-07	YES	The tool provided for FHE (Cingulata, see Section 7.2) supports the required maximum security level (128 bits).
C3ISP-Comp-CDA-08	YES	The tool provided for FHE (Cingulata, see Section 7.2) supports the required randomization methods

6. Visualization of Security Analytics

6.1. Tool/technology description and current state

The SATURN Visual Analytics software suite will be used for visualizing the security data and analytics results that are produced and shared by different stakeholders in various C3ISP pilot scenarios. SATURN is a stand-alone web application that has been developed by BT Research & Innovation and is currently used within BT's Security Operation Centres (SOCs) to identify data anomalies and attack patterns in network and system logs. With SATURN, visualization is the start of the analysis process not the end as is the case with most business intelligence and analytics systems. This enables humans to be an integral part of the analysis process to spot those patterns that automated systems would otherwise miss. SATURN allows users to explore visual summaries of data from stores in excess of billions of events with low-latency querying. Data may be read from existing tables of data in an RDBMS, in tables in HDFS accessed via Cloudera Impala, or from uploaded Microsoft Excel or CSV files. Figure 6 shows a typical flow of SATURN use involving its different components.

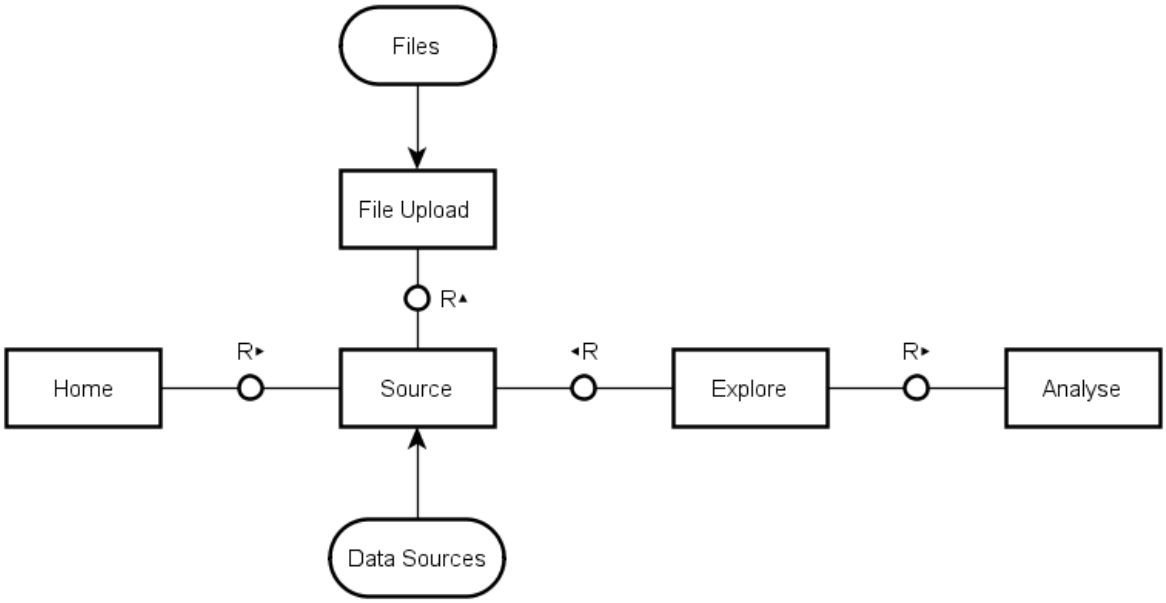

Figure 6: Typical flow of SATURN use

The *Home* component provides an overview of the datasets available for visualization. This information is requested from the *Source* component, which manages the list of data sources as well as the settings and configurations for retrieving the data, e.g. connection URL, database name, access credentials, attributes mapping, etc. The *File Upload* component ensures that data can be read from files. The *Explore* component is a visualization layer for interacting directly with the data held in a database, or uploaded from a file. Users can construct complex queries intuitively using graphical representations without requiring programming or database skills. Once created, the queries can be passed on to the *Analyse* component for fine-grained visualization and in-depth analysis. Users can apply flexible and responsive filtering on the data based on workbench of ten different types of interactive visual gadgets (e.g. bubble chart, link-analysis, trends, clustering, geographical) which are linked together, as depicted in Figure 7.

SATURN already has TRL 9 status. It is considered as a *Legacy Analytics Engine* within the IAI subsystem of the C3ISP reference architecture. It is anticipated that SATURN service can be invoked by C3ISP Prosumer via the IAI API and the required data retrieved from the corresponding *Virtual Data Lake* instance. The data may first need to be pre-processed or sanitized by appropriate C3ISP components, such as the encryption and anonymization components (part of the DSA Adapter), in order to comply with the associated DSA policies.

Figure 7: SATURN Visual Analytics applied to Cyber Security Data

6.2. Tool/technology requirements

Table 12 – Visualization of Security Analytics Requirements

ID	Priority	Requirement	In order to fulfil D7.1 Requirement(s)
C3ISP-Com-Usa-001	MUST	Provide an intuitive graphical interface for user interactions.	C3ISP-Usa-006
C3ISP-Com-Usa-002	MUST	Provide a capability for reading structured data (e.g. raw security data, analytics results, etc.) from stores, or uploading data files.	C3ISP-Usa-006

C3ISP-Com-Usa-003	MUST	Provide a capability for querying, filtering and exploring (security) data that has been ingested to the system.	C3ISP-Usa-006
C3ISP-Com-Usa-004	MUST	Provide a capability for visualizing and analysing security data to spot anomalies, trends, patterns, etc.	C3ISP-Usa-006

6.3. Requirements analysis

Table 13 – Visualization of Security Analytics Requirements Status

ID	MET	Description
C3ISP-Com-Usa-001	YES	The tool offers a rich and intuitive graphical user interface to allow users carry out different tasks, starting from configuring data sources, event attributes mapping, exploring and filtering the datasets, up to visualising the data using a variety of visual gadgets.
C3ISP-Com-Usa-002	YES	The tool supports two ways for data ingestion. Data can be read from databases, i.e. RDBMS (Oracle, PostgreSQL, MySQL, SQL server), columnar database (Vertica), and Hadoop cluster (via Cloudera Impala), or uploaded from files in Excel and CSV formats.
C3ISP-Com-Usa-003	YES	The tool provides an “Explore” component which is capable of handling billions of events at a time and may be used to construct more specific, complex database queries for further analysis. In total there are 10 different graph views, such as bar or line charts, mini graph view, parallel coordinates, etc., which can be used to help users visually filter and explore the data.
C3ISP-Com-Usa-004	YES	The tool provides an “Analyse” component which allows for fine-grained visualization and analysis of the data read into the system. It consists of one or more ‘Gadget’ windows which can themselves be configured to show one of several different visualizations such as bubble chart, trends, network, radial, pie chart, geolocation map, etc. It also supports unsupervised clustering algorithm and visualization to elicit patterns of interest in the data.

7. Anonymization and Homomorphic Encryption Algorithms

Anonymization and homomorphic encryption algorithms permit privacy-preserving data treatments. Both encryption algorithms and anonymization algorithms aim at protecting privacy. Privacy is one major goal in information security [7], it targets **message content confidentiality**. We consider anonymization algorithms in Section 7.1. In the literature, anonymity can refer to message source (or destination) confidentiality [4],[11], while here we consider data anonymization [9]. It conceals **personally identifiable information** (PII for short) from database while preserving its format and data type. It serves to minimize the risk to associate an identity with a record. Certain techniques belongs to cryptography, for instance, differential privacy [9]. In a second time, we consider homomorphic encryption algorithms such as FV scheme proposed by Fan and Vercauteren in [10]. Homomorphic encryption answers a long-time [8] open problem in cryptography. It permits to evaluate computations over encrypted data (also called **ciphertexts**) without being able to decrypt them. The result of a homomorphic computation is a new ciphertext, which encrypts the awaited result without revealing input data. This serves, for instance, to delegate computations to an untrusted party.

7.1. Anonymization Algorithms Requirements

7.1.1. Tool/technology description and current state

The Anonymization Toolbox is a research prototype and its main feature is a randomization method that anonymizes individual values (and thereby one's membership in the data set) via Differential Privacy [9], e.g. the Laplace mechanism and Geo-indestinguishability for location data [14]. As mentioned in [13]: “Differential privacy addresses the paradox of learning nothing about an individual while learning useful information about a population.”

Furthermore, it supports the removal/suppression of columns or certain information based on delimiters, e.g. removal of lower parts of a MAC address (delimited by “:”) or subnets from IPv4 addresses (delimited by “.”).

The tool runs on a server and listens on port 8080 for XML requests. A XML request contains as parameters:

- input and output table names,
- columns to anonymize,
- and the anonymization method and its parameters for each column.

A minimal example XML request looks as follows:

```
<?xmlversion="1.0"?>
<request>
      <storage>
            <type>MySQL</type>
            <source>
                  <name>inputTableName</name>
                  <query>
                  <!-- columns to process -->
                        <columns>
```

```
                        <column>
                              <name>column_name</name>
                              <type>real</type>
                        </column>
                  </columns>
            </query>
      </source>
      <target><name>outputTableName</name></target>
</storage>
<!-- pre-processing of data -->
<preprocessors>
</preprocessors>
<!-- processing of columns via anonymizationmechanisms -->
<mechanisms>
      <mechanism>
            <name>laplace</name>
            <privacy>0.1</privacy>
            <sensitivity>
                  <name>globalSensitivity</name>
                  <parameters>
                        <parameter>
                              <name>globalSensitivity</name>
                              <value>1</value>
                        </parameter>
                  </parameters>
            </sensitivity>
            <columns>
                  <column>
                        <name>column_name</name>
                  </column>
            </columns>
      </mechanism>
</mechanisms>
<!-- final processing ifneeded -->
<resultprocessors>
</resultprocessors>
</request>
```

Data is read from the input table and the anonymized output is stored in the output table (or returned with the XML response if desired). Access to a MySQL or SAP HANA[7] database containing the data is required.

[7] https://www.sap.com/products/hana/features/in-memory-database.html

The tool is currently at TRL 4 for data processing and anonymization, and it will be maturated to TRL 6 in the C3ISP project. It currently supports anonymization via suppression (removal of entire columns and some delimited substrings), for real numbers (Laplace mechanism) and geo-location (Geo-indistinguishability). Support for string, i.e. categorical attribute, anonymization with Differential Privacy based on the Exponential mechanism [14] is being investigated.

The tool will be integrated in the DMO Engine within the DSA Adapter which in turn is a part of ISI subsystem.

7.1.2. Tool/technology requirements

Table 14 – Anonymization Algorithms Requirements

ID	Priority	Requirement	In order to fulfil D7.1 Requirement(s)
C3ISP-Com-AA-001	MUST	Define post-processing parameters on data / analytical results for sanitisation. (For details see C3ISP-Com-AA-002, C3ISP-Com-AA-003)	C3ISP-Fun-DA-002
C3ISP-Com-AA-002	MUST	Define input data type before anonymization.	C3ISP-Fun-DA-018 C3ISP-Fun-DA-019
C3ISP-Com-AA-003	MUST	Define anonymization method and parameters based on analytical treatment.	C3ISP-Fun-DA-021
C3ISP-Com-AA-004	MUST	Provide an interface for privacy-preserving operations (e.g. data sanitisation).	C3ISP-Fun-DA-003

7.1.3. Requirements analysis

Table 15 – Anonymization Algorithms Requirements Status

ID	MET	Description
C3ISP-Com-AA-001	PARTIALLY	Anonymization techniques and their parameters can be easily defined via XML request. The tool currently supports: • Suppression of columns; • Laplace mechanism for numbers; • Geo-indistinguishability for single locations; • Removal of delimited substrings. Further techniques and their parameterization, e.g.

		Exponential mechanism for certain differentially private anonymization of strings, could be added. This requires additional XML parsing and implementation work.
C3ISP-Com-AA-002	PARTIALLY	See C3ISP-Com-AA-001.
C3ISP-Com-AA-003	PARTIALLY	See C3ISP-Com-AA-001.
C3ISP-Com-AA-004	PARTIALLY	An interface is provided via XML requests. The ccomponent should be implemented following the "as a service" paradigm to be integrated as a plug-in in the DMO Engine.

7.2. Homomorphic Encryption Algorithms Requirements

7.2.1. Tool/technology description and current state

The Full Homomorphic Encryption (FHE) Analytics is a part of C3ISP Analytics Engine (see Section 2).It is based on Cingulata (previously Armadillo) which is a source to source compiler developed by CEA, used for applying homomorphic cryptographic techniques which on top of allowing the scrambling of data in order to protect its confidentiality also provides the necessary mathematical building blocks for performing privacy-preserving calculations, by the execution of general algorithms directly on encrypted data. It is described in Figure 8.

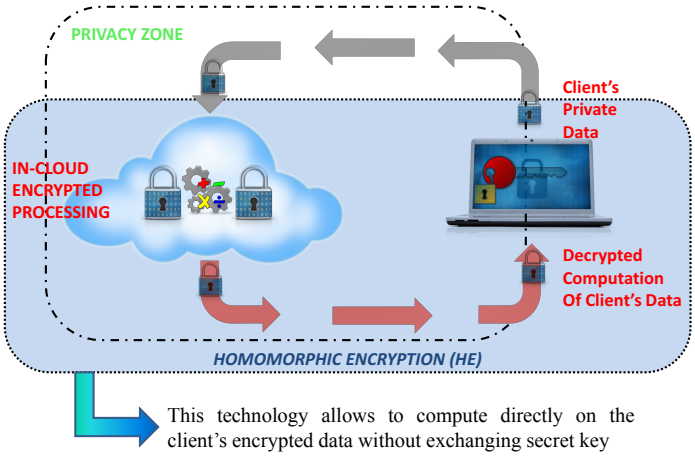

Figure 8: High level view of Cingulata

Its architecture is pictured in Figure 9. It consists mainly of the following components:

- A compiler infrastructure for high-level cryptocomputing-ready programming, taking C++ code as input.
- Boolean circuit optimization, parallel code generation and «cryptoexecution» runtime environment.

- Optimized prototypes of the most efficient homomorphic encryption systems known so far.

Due to the low-level formalism of homomorphic encryption, a compiler chain has to manipulate Boolean circuits. In homomorphic encryption context, the main characteristics of a Boolean circuit is the Multiplicative Depth, i.e. the largest number of chained AND operators required to compute an output. The higher the multiplicative depth, the larger the parameter of the homomorphic encryption systems (with direct impact on the performances).

In Figure 9, Cingulata first compiles an application written in C++ language into a Boolean circuit (blue area). Whereupon, Cingulata optimizes the circuit to improve performance (red area). To save bandwidth, transcryption can be performed with a homomorphic-friendly standard cryptosystem (green area). In the end, the circuit is evaluated over ciphertexts using a homomorphic scheme (red area).

With respect to Boolean circuit optimization we are presently using ABC[8] as the Boolean circuit optimization engine within our compiler chain. ABC is a well-known open source System for Sequential Synthesis and Verification developed by the Berkeley Logic Synthesis and Verification Group. The ABC-optimized Boolean circuit is turned into OpenMP[9]-compliant parallel code.

More details on our homomorphic encryption compiler is provided in [15].

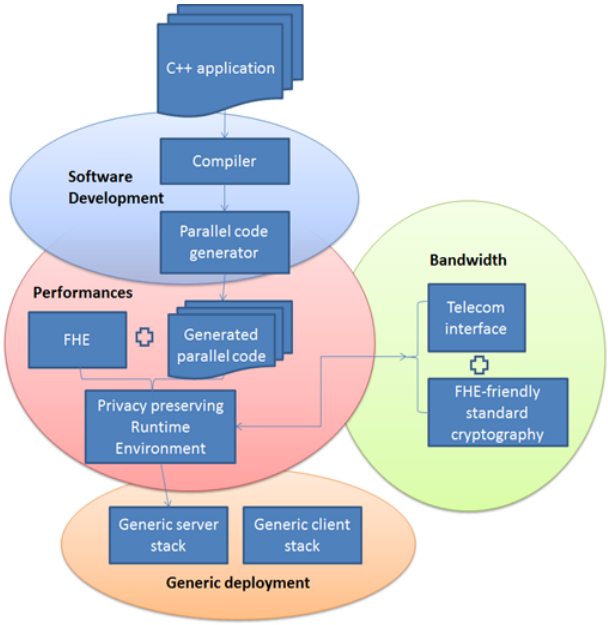

Figure 9: Cingulata workflow

Currently, Cingulata is in TRL 4 for data analytics in cybersecurity, and the C3ISP project will bring it to TRL 6. Cingulata has been developed in 2016 for the EIT Digital[10] project HC@WORKS.

[8] https://people.eecs.berkeley.edu/~alanmi/abc/
[9] http://www.openmp.org/
[10] https://www.eitdigital.eu/

7.2.2. Tool/technology requirements

Table 16 – Homomorphic Encryption Algorithms Requirements

ID	Priority	Requirement	In order to fulfil D7.1 Requirement(s)
C3ISP-Com-HE-001	MUST	The component must support several input data type..	C3ISP-Fun-DA-018 C3ISP-Fun-DA-019 C3ISP-Fun-DA-020
C3ISP-Com-HE-002	MUST	The component must support the Pilots' scenarios and the related analytics treatments on data.	C3ISP-Fun-DA-021 C3ISP-Fun-DA-022 C3ISP-Fun-DA-023
C3ISP-Com-HE-003	MUST	The component must support the order of magnitude of number of treated data.	C3ISP-Fun-DA-022 C3ISP-Fun-DA-023
C3ISP-Com-HE-004	MUST	The component must support the encoding of the clear data to be manipulated adopted in the pilots.	C3ISP-Fun-DA-019

7.2.3. Requirements analysis

Table 17 – Homomorphic Encryption Algorithms Requirements Status

ID	MET	Description
C3ISP-Com-HE-001	PARTIALLY	We represent IPs addresses in Cingulata. Other data type could be defined to address the C3ISP Pilots' needs.
C3ISP-Com-HE-002	PARTIALLY	We can realise the following treatments over ciphertexts with Cingulata. 1) test (in)equality of IPs. 2) test membership of an IP to a list of IPs 3) compute the intersection of two lists of IPs 4) count the number of occurrences of an IP in a list of IPs.
C3ISP-Com-HE-003	NO	We ignore IP list size in practice. List size impacts time and memory requirements. Managing big lists can require additional (pre/post) computations on the client side.
C3ISP-Com-HE-004	PARTIALLY	We use precomputation to represent IPs addresses as integers of fixed size in Cingulata. Other data types would require other precomputations before data encoding such as hashing for variable-length data. Encoded values can be integers of size 8, 16, 32 or 64 bits.

8. Managed Security Services

8.1. Tool/technology description and current state

In context of the C3ISP project, the Managed Security Services (MSS) component provided by the BT is part of BT's Managed Cloud Security Products & Solutions portfolio, and is also known internally as the "Intelligent Protection Service" (IPS). As currently it is only required for use as an MSS in the SME Pilot, it is also referred to as "SME MSS". However, other C3ISP Pilots can also make use of different commercial or open source MSS as per their requirements, including IPS. IPS is a TRL 9 solution and is commercially sold to BT Managed Cloud Security customers as a value-added service.

IPS enables the protection of systems, applications and data processing on a mix of public and private cloud environments through a collection of security functions that can be offered as managed security services. The IPS can also be integrated in the BT's Cloud Service Store as a horizontal/common capability at the cloud management automation layer, thereby enabling the application and data protection functions to become selectable properties of any application stack that the user chooses to assemble on any cloud platform. Due to this flexibility, the IPS offers a new customer experience to seamlessly manage security in the cloud.

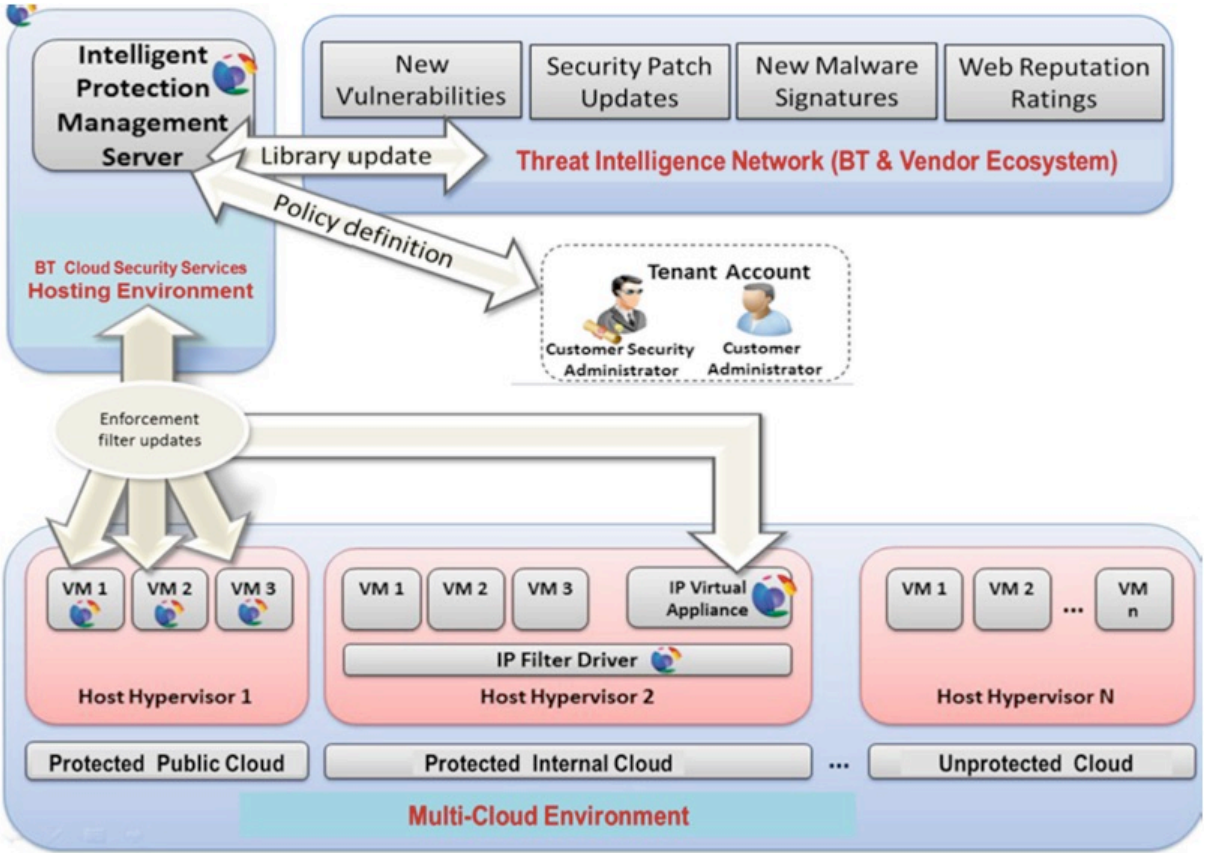

Figure 10: Overview of the IPS high level architecture

An architectural overview of the cloud-based Intelligent Protection solution offered to each tenant can be divided in three dimensions as depicted in Figure 10:

- Policy enforcement: this is the mechanism used to manage the protection of a system; in this case it is an agent installed on a virtual machine or a physical server.
- BT Cloud Security Service: this is the management mechanism used by the IPS for defining security policies based on a library of rules that include virtual patches for a very large number of systems and applications, firewall and protocol rules, etc., and for updating the configuration and enforcement rules of the agents.
- Threat intelligence: this is the mechanism for enhancing the data-base of heuristic rules, attacks or virus signatures, vulnerabilities, etc., via a network that includes a large number of security and application vendors, as well as contributions from BT's security ecosystem.

Using an agent-based on-boarding model, VMs or physical servers can be connected to the IPS and enable their administrators to remotely monitor and manage the protection of their environment. In addition to the BT Cloud Service Store [17] integration, there are three further ways of making a VM manageable by Intelligent Protection, depending on the level of integration of the corresponding Cloud environment with the Intelligent Protection service:

- The user can download the agent installer from the IPS, according the operating system and machine architecture of their VMs or physical servers.
- The users can be provided an installation and configuration script compatible with the operating system and machine architecture of their VMs or physical servers. On running the script, it automatically contacts the IPS and download, register and activate the appropriate agent software on the VM or the physical server.
- The Cloud service providers offer a VM template with a pre-installed agent that is then activated by the users by providing their IPS credentials for verification.

8.2. Tool/technology requirements

Table 18 – Managed Security Services Requirements

ID	Priority	Requirement	In order to fulfil D7.1 Requirement(s)
C3ISP-COM-REQ-MSS-1	MUST	Hardware Requirements for IPS - 8 GB RAM - 100+ GB Disk Space	
C3ISP-COM-REQ-MSS-2	MUST	Software Requirements for IPS - Windows Server 2012 or 2012 R2 (64-bit) - Microsoft SQL Server 2014	
C3ISP-COM-REQ-MSS-3	MUST	Network Requirements for IPS - Open incoming ports80, 443, 1433-1434, 4118-4128	

8.3. Requirements analysis

Table 19 – Managed Security Services Requirements Status

ID	MET	Description
C3ISP-COM-REQ-MSS-1	YES	This requirement is currently fulfilled by an instance of the IPS deployed on the BT's Cloud research platform (https://ipserver.zion.bt.co.uk:4119/).
C3ISP-COM-REQ-MSS-2	YES	This requirement is currently fulfilled by an instance of the IPS deployed on the BT's Cloud research platform (https://ipserver.zion.bt.co.uk:4119/).
C3ISP-COM-REQ-MSS-3	YES	This requirement is currently fulfilled by an instance of the IPS deployed on the BT's Cloud research platform (https://ipserver.zion.bt.co.uk:4119/).

9. Conclusions

This deliverable is the first output of WP8, "C3ISP Data Sharing, Analytics and Crypto Technology Maturation", due at M12. The main goal of this deliverable is to collect the features and functionalities offered by the set of tools and technologies that are provided by the C3ISP partners for the implementation of the C3ISP Framework, i.e., of the Information Sharing Infrastructure (ISI) and of the Information Analytics Infrastructure (IAI). These tools and technologies cover most of the functionalities required in the C3ISP project. This deliverable reports a detailed description of each of them, specifying the current maturation level through the Technology Readiness Level (TRL) and the components of the C3ISP architecture (defined in D7.2) that can benefit of each of the tools provided by the C3ISP partners. Another important contribution provided by this deliverable is the specification of the requirement of these components starting from the general requirements that have been defined in D7.1. Finally, for each of these requirements, this deliverable reports which is already satisfied by the current version of the tool, which is partially satisfied, and which, instead, is not satisfied at all, thus identifying the tools which require a further maturation in order to be adopted to satisfy the requirements of the C3ISP Framework.

10. References

This section lists the references used throughout the document:

[1] I. Matteucci, M. Petrocchi, and M. L. Sbodio. *CNL4DSA: a Controlled Natural Languagefor Data Sharing Agreements*. In SAC: Privacy on the Web Track. ACM, 2010. 21, 23, 54

[2] A. Grigoris and F. Van Harmelen. *Web Ontology Language: OWL*. In Handbook on Ontologies in Information Systems, pages 67–92. Springer, 2003. 18

[3] http://csrc.nist.gov/groups/SNS/rbac/

[4] OASIS Consortium. *eXtensible Access Control Markup Language (XACML) Version 3.0,*OASIS Standard, 22 January 2013

[5] J. Park and R. Sandhu. 2004. The UCON ABC usage control model. ACM Trans. Inf. Syst. Secur. 7, 1 (February 2004), 128-174.

[6] R. J. Anderson. *Security engineering: a guide to building dependable distributed systems*. John Wiley & Sons, 2010.

[7] A.J. Menezes, P.C. Van Oorschot, and S.A. Vanstone. *Handbook of applied cryptography*. CRC press, 1996.

[8] C. Gentry. *Fully homomorphic encryption using ideal lattices.STOC*. Vol. 9. No. 2009. 2009.

[9] C. Dwork. "Differential privacy: A survey of results.*International Conference on Theory and Applications of Models of Computation*. Springer, Berlin, Heidelberg, 2008.

[10] J. Fan and F. Vercauteren. *Somewhat Practical Fully Homomorphic Encryption.* IACR Cryptology ePrint Archive 2012 (2012): 144.

[11] C. Paar, and J. Pelzl. *Understanding cryptography: a textbook for students and practitioners*. Springer Science & Business Media, 2009.

[12] B. Raghunathan. *The complete book of data anonymization: from planning to implementation*. CRC Press, 2013

[13] C. Dwork and A. Roth. *The algorithmic foundations of differential privacy*. Foundations and Trends in Theoretical Computer Science, 2014.

[14] M.E Andrés, N.E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. *Geo-indistinguishability: Differential privacy for location-based systems.* In Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, 2013

[15] F. McSherry, and K. Talwar. *Mechanism design via differential privacy*. Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on. IEEE, 2007

[16] S. Carpov, P. Dubrulle, R. Sirdey. "*Armadillo: a compilation chain for privacy preserving applications*", Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security (3rd International Workshop on Security in Cloud Computing), pp. 13-19, 2015

[17] G. Ducatel, J. Daniel, T. Dimitrakos T, F. El-Moussa, R. Rowlingson, A. Sajjad. *Managed security service distribution model*. In Cloud Computing and Intelligence Systems (CCIS), 2016 4th International Conference on 2016 Aug 17 (pp. 404-408). IEEE.