Establish Pan-European Information Space to Enhance seCurity of Citizens

D5.2. - Information Interoperability Specification

Grant agreement number:	**607078**	**Date of deliverable:**	**2016-07-31**
Date of project start:	**2014-06-01**	**Date of submission:**	**2016-09-30**
Duration of project:	**36 months**	**Deliverable approved by:**	**S. Bounouri (ADS)** **W. Vorraber (TUG)**
Lead Beneficiary:	**FRQ**		
Contributing Beneficiaries:	**ADS, AIT, HITEC, HWC, IES, KUL, UNIST**		

Executive Summary

This document *D5.2 – Information Interoperability Specification* describes the technical core of the Common Information Space (CIS) introduced in EPISECC. It summarises and combines the results of the previous work packages and specifies the architecture and concepts for secure automated information sharing. Different types of information are exchanged between tools used by organisations related to crisis and disaster management (responders, infrastructure providers, authorities, scientific institutes etc.).

The need for fast and reliable exchange of information between autonomous stakeholders which are not collaborating in their day-to-day business but have to co-operate during disaster response can be derived from the results of WP2 (analysis of the crisis management approaches) and WP3 (Pan European Inventory of events/disasters). The taxonomy model developed in WP4 together with the terminologies used by CIS participants builds the basis for semantic transformations and annotations that help overcoming language barriers and incompatible encodings in different organisations and tools. This specification D5.2 builds the basis for the development of a CIS prototype to be evaluated in WP6 proof of concept.

The basic idea is to connect tools via adaptors. Proprietary data formats and interfaces of one tool are transformed to messages using established standards and protocols that can be received by the adaptors of any other CIS participant (syntactical interoperability). Peer-to-peer message distribution and synchronisation keeps all participants up-to-date even in the case of a temporary or partial network breakdown, without the need of a heavy central message broker/server.

Information security is key in the CIS architecture. All message contents are encrypted. The participants need trusted certificates, generated during the registration process. Cooperation group online rooms (CGOR) are defined for partners collaborating in one event/incident and have to be accepted by the members. Information assigned to a CGOR is readable only for CGOR members.

Making messages of a foreign sender understandable for the receiving tools and human users is very important. Based on the EPISECC taxonomy, key terms of the messages are interpreted. The information consumer gets the original message together with semantic transformations and annotations reflecting his own terminology (semantic interoperability).

Table of Content

List of Tables

List of Figures

List of Acronyms

Abbreviation	Description
ACL	Access Control List
C² / C&C	Command and Control (system)
CAP	Common Alerting Protocol (OASIS standard)
CEN	European Committee for Standardization (Comité Européen de Normalisation)
CGOR	Cooperation Group Online Room
CIS	Common Information Space
CIS DA	CIS Directory Agent
CM	Crisis Management
EDXL	Emergency Data Exchange Language (family of OASIS standards)
EDXL DE	EDXL Distribution Element (message envelope)
EmOrg	Emergency Organisation
EMSI	Emergency Management Shared Information (ISO/TR 22351:2015)
GIS	Geographic Information System
ICT	Information and Communication Technology
MLP	Mobile Location Protocol
MOM	Message Oriented Middleware
OASIS	Organization for the Advancement of Structured Information Standards
OGC	Open Geospatial Consortium
PKI	Public Key Infrastructure
PPDR	Public Protection and Disaster Relief
REST	Representational State Transfer (Web service interface)
SA	Situational Awareness
SensorML	Sensor Model Language
SOAP	Simple Object Access Protocol (Web service interface)

SOS	Sensor Observation Service
W.I.	Wrapped Information (see 5.4)
XML	Extensible Mark-up Language

Information Interoperability Specification

1. Introduction

1.1. Context of D5.2 in EPISECC

EPISECC WP5 *Architecture of Common Information Space* designs an information sharing platform for crisis and disaster management on all relevant layers. Task 5.1 *Protocol & network interoperability* highlights the physical layer (radio and mobile communication) and the connection of radio network with the common information space. Task 5.3 *Operational Interoperability* handles the procedures required for establishing interoperability between involved organisations. Task 5.2 *Information Interoperability* and this document D5.2 specify the technical platform of the Common Information Space (CIS). It summarises and combines the results of the previous work packages and describes the architecture and concepts for secure automated information sharing. Different type of information is exchanged between tools used by organisations related to crisis and disaster management (responders, infrastructure providers, authorities, scientific institutes etc.).

The need for fast and reliable exchange of information during disaster response has been derived from the results of WP2 (analysis of the crisis management approaches) and WP3 (Pan European Inventory of events/disasters). The taxonomy model developed in WP4 together with the terminologies used by CIS participants builds the basis for semantic transformations and annotations. This specification D5.2 builds the basis for the development of a CIS prototype to be evaluated in WP6 proof of concept. Following an agile and iterative approach, the first versions of the prototype (Task 6.2, D6.2) and this specification were developed step by step in parallel.

1.2. Structure of D5.2

This document *D5.2. – Information Interoperability Specification* describes the technical core of the Common Information Space (CIS) and specifies the concepts for secure sharing of understandable information.

Starting from the WP3 results, Chapter 2 summarises the goals for information interoperability and related use cases in crisis and disaster management, and the idea of a common operating picture.

The concept of a common information space as an information sharing platform is based on standardised messages exchanged between the collaborating parties. The selected standards are described in Chapter 3, while the concepts and the architecture of CIS are content of Chapter 4. Chapter 5 goes deeper into the design and provides details of the components and features of the CIS as well as data security concepts.

Chapter 6 considers ethical and legal aspects related to information ownership and sharing of sensitive data. Chapter 7 finally describes the processes and procedures foreseen for secure registration of CIS participants and administration of cooperation groups.

1.3. Methods

The need for new solutions for information interoperability in public protection and disaster relief (PPDR) is confirmed by the results of WP2 Analysis of the Crisis Management Approaches [1] and WP3 Pan European Inventory of Disasters [2]. While an increasing number of PPDR organisations have proprietary IT based command and control tools and information management systems in place, there is a lack of standardised interoperability between systems of organisations which don't collaborate in their day-to-day business. Information is still exchanged between organisations by human interactions like phone and radio calls, e-mail and fax.

Based on the insights from the interviews conducted in the Inventory, the Task 5.2 partners developed a first idea of an information sharing platform (the Common Information Space CIS) in a workshop. This early concept was presented at the first Advisory Board meeting and discussed with PPDR practitioners. Based on this consolidated idea, the architecture of the Common Information Space was developed in detail in an iterative process with many online meetings and additional four physical workshop meetings among the technical partners. The concepts were regularly aligned with the practical experience of consortium members and reviewed in the second Advisory Board meeting.

The development of the CIS prototype (Task D6.2) started in parallel to the elaboration of the CIS architecture. The agile development process allowed a continuous feedback loop between SW prototyping and architecture definition and the exploration of novel technologies and open source solutions available in the public domain.

A first proof of concept of parts of the CIS concept (adaptors providing syntactical interoperability) was the successful use of a Common Information Space instance for experiments 41 and 42 in the FP7 project DRIVER [5] in April 2016.

For the collaborative elaboration of the CIS concepts, drafts of documents and diagrams were circulated and discussed between the partners. The final diagrams and texts are part of this document. The present deliverable document is the result of collaborative editing, where every partner took responsibility for defined sections, and the deliverable leader for the final compilation. All partners contributed to the whole document with their comments and suggestions.

UML use case diagrams illustrate the processes for secure information interoperability within the CIS. More detailed UML diagrams documenting the design of the CIS prototype will be provided within D6.2.

2. Goals for Information Interoperability

2.1. What information interoperability means

Management concepts based on hierarchical organisations have dominated crisis & disaster management in the past. Today, a paradigm change is on the way. While professional organisations are of course still organising themselves in a hierarchical way, they are pushed into collaborating with an increasing number of organisations and even the citizens (i.e. crowds) operating in a different manner. Crisis & disaster management will in the future be a highly networked and collaborative activity while still aiming for maximum efficiency and best performance. Today, the term "networked security" is used for a new way of electronically facilitated collaboration between stakeholders involved in a civil disaster relief mission within one territory or even cross-national. Process data are shared via the information cloud and enhance the effectiveness of well proven processes (Figure 1).

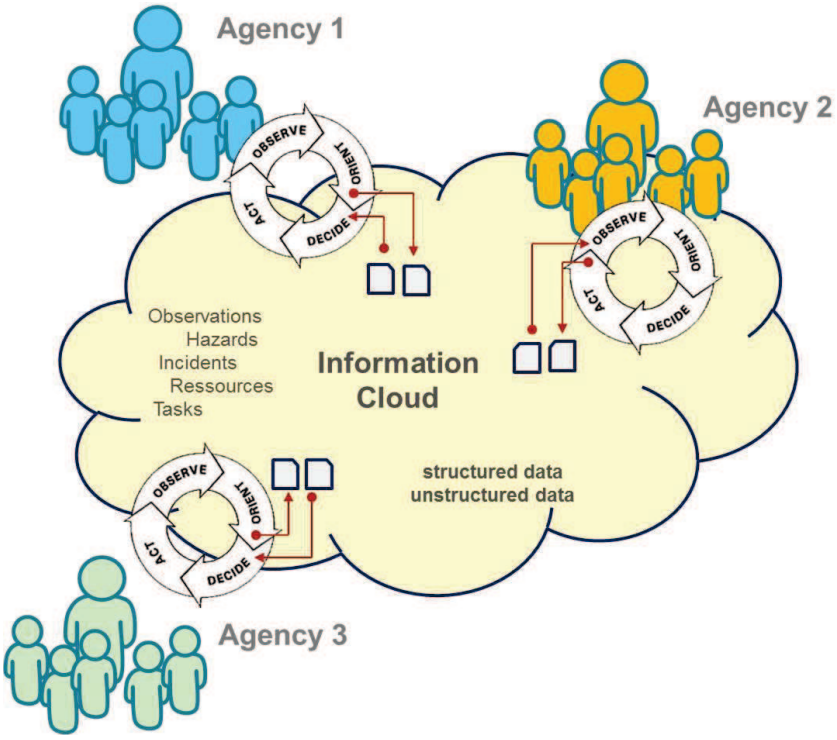


Figure 1: networked security – information interoperability

The established networks generally are mixed-sector networks in the sense that the strategic orientations of the contributing organisations differ. The stakeholders are on the one hand organizations from the public safety domain where crisis management is part of their core business. On the other hand, also organizations whose core-business has nothing to do with crisis management play an important role; they are required to contribute to the crisis management effort in addition to their own business continuity management.

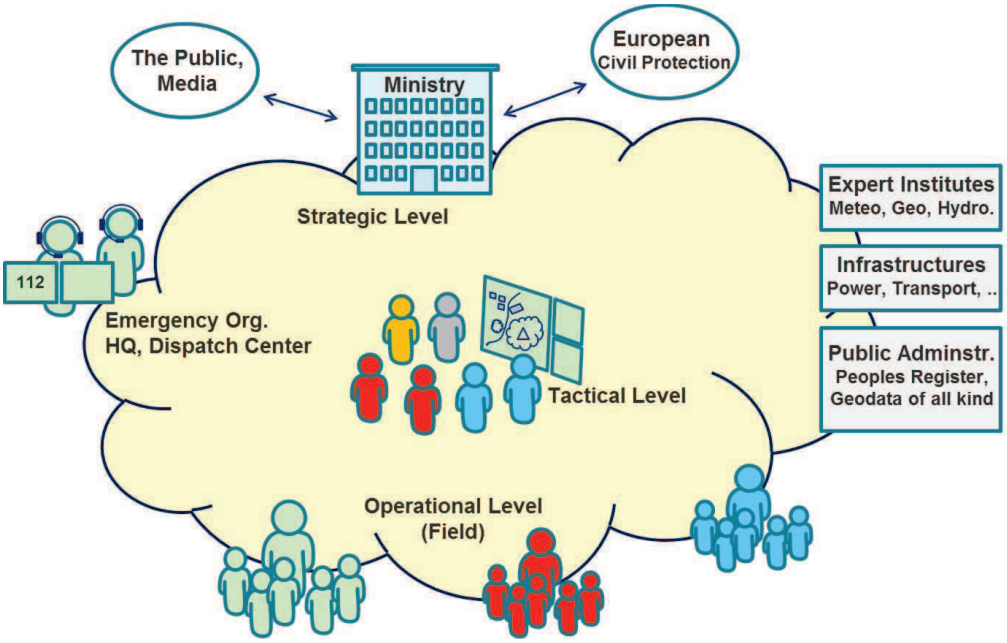


Figure 2: Information interoperability stakeholders

- Professional responders (governmental and non-governmental organisations)
- Public authorities, directly responsible for civil protection
- Departments of the public administration who can contribute all kind of valuable data
- Operators of vital infrastructures (in many cases privately owned)
- Industries working with dangerous goods (Seveso II)
- Experts and organisations who operate sensor networks, evaluate data or calculate forecasts
- Public media and citizens

2.2. Objectives of information interoperability in crisis management

The main focus of the Common Information Space is to facilitate and support multi-organizational collaboration during the response phase of a crisis and disaster management effort.

During the time-critical response phase within a crisis or disaster management action, cross-organizational collaboration and the related information management today is still mostly based on face-to-face meetings, telephone calls, fax transmissions, email messages, paper charts, whiteboards, and proprietary electronic systems. The project team gathered this insight from the inventory of disasters [2] and from consortium members' experience as supplier of control centre solutions for the public safety domain in various European member states. As a consequence, situation awareness is hampered by a fragmentation of relevant information into pieces held by different stakeholders. Within the highly collaborative scenarios of the civil crisis management operations such as floodings, forest fires, or earth quakes this fragmentation causes uncertainty whether the information base for critical decisions is up-to-date, comprehensive and valid. Figure 3 shows an example for fragmentation in the context of typical stakeholders within a federal system like Germany.

Figure 3 Fragmentation of Information between Stakeholders

Decision making based on a comprehensive picture of the situation requires exchange, verification and integration of all the different pieces of information provided by the stakeholders with their organizational and cultural background. At the same time a common understanding of the situation is also a basic prerequisite for successful collaboration.

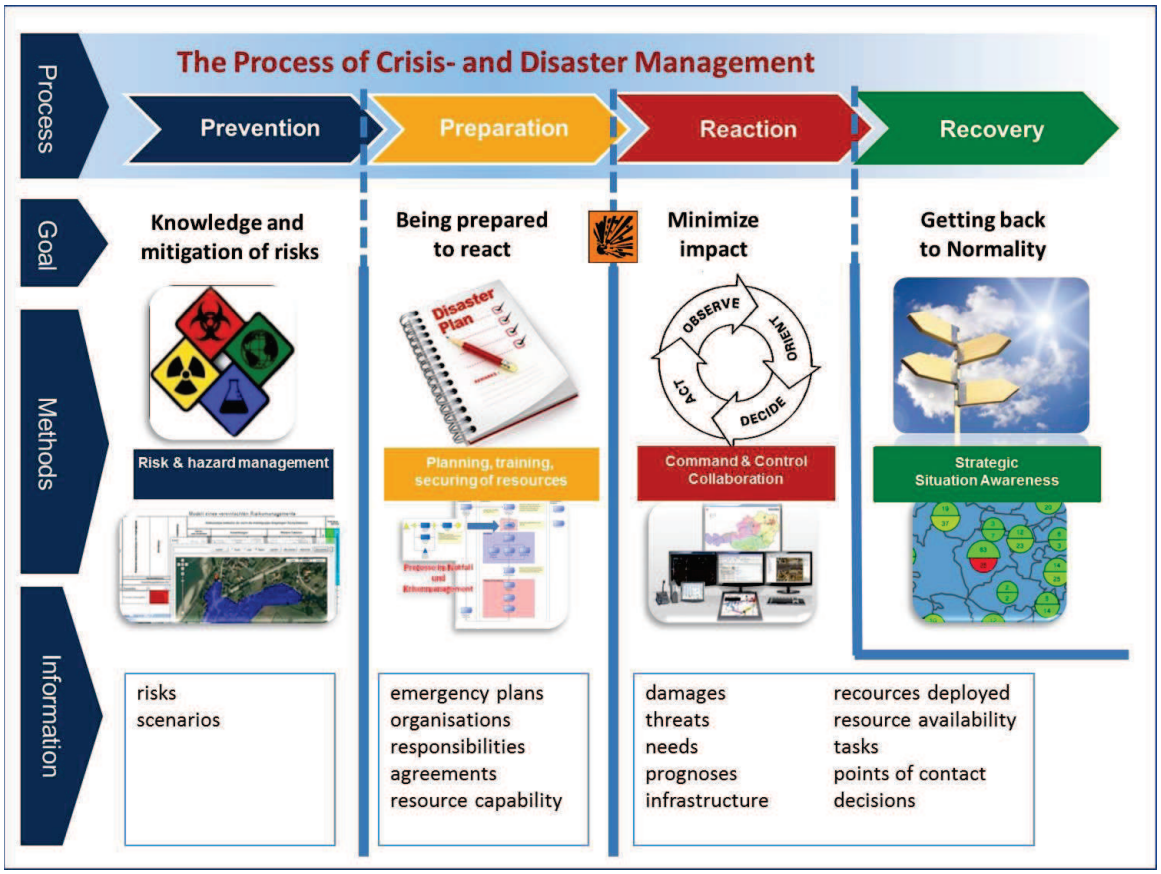


Figure 4: Linear Presentation of the Crisis Management Lifecycle

The information involved is generated within the four phases of the crisis management life-cycle. The linear representation shown in Figure 4 allows visualizing the specific goal, the applied processes, and the involved information for each phase.

2.3. Situation Awareness and Common Operating Picture

As already depicted in section 2.2, one of the most essential challenges in emergency and disaster response operations is to immediately obtain and continuously maintain situational awareness (SA). The investigation regarding the status quo in information and communication technology (ICT) in the domain of PPDR (commercial, research and governmental projects and applications) [1] has shown that the majority of ICT applications and projects address SA aspects. To set the scene for the interoperability use-cases some base line definitions concerning major aspects of an organization-specific situational awareness are given:

General situation in the affected area

- Time of the day / year,
- Weather conditions,
- Special local circumstances,
- Restrictions in infrastructure,
- Restriction in basic supply,
- Public behavior,
- Other particularities with general effects.

Each of these aspects has to cover not only the status at a given moment, but also possible future developments.

Threats and Damages

- Kind and cause of threats / damages,
- Extent and impact of threats / damages,
- Who or what is affected by threats / damages,
- Subsequent threats and damages.

Own Resources

The statements about the organisation's own resources include:

- Forces and means already deployed,
- Forces and means at further disposal,
- Additional means for disaster relief.

Organisations involved in disaster relief operations perform information gathering activities to reach situational awareness that includes as many of the above mentioned aspects as possible. The extent of information gathering is dependent from the organisation's role within the disaster relief network, its responsibilities and capabilities and from the point in time the organisation is activated/deployed to disaster response. Basically two main characteristics in information gathering can be observed:

Assessing: information collection activities in the operational area to support the own organization's responsibilities and/or operations

Receiving: information collection from other responders/stakeholders to support the own organization's responsibilities and/or operations

Simultaneous self-assessing and receiving of information with regard to the same aspects is very common. As confirmed by the analysis performed in WP3 using the inventory of past disasters and as shown in Figure 3 when exchanging information related to the disaster, their operations etc. the actors in disaster response still mainly rely on means of voice communication (meetings, phone, radio) or on FAX and email.

The major groups of actors in disaster relief operations and their sources/channels of information are show in Table 1.

Organisation specific situational awareness	Aspects	Spatial reference Yes / No	Responders - Assessing the situation vs. Receiving information			
			local responders	national responders	international responders	local emergency management authority (LEMA)
General situation in the affected area	Time of the day / year	N	A /	A /	A /	A /
	Weather conditions	Y	A / R	A / R	/ R	A / R
	Special local circumstances	Y	A / R	A / R	/ R	A / R
	Restrictions in infrastructure	Y	A / R	A / R	/ R	A / R
	Restriction in basic supply	Y	A / R	A / R	/ R	A / R
	Public behaviour	Y	A / R	A / R	/ R	A / R
	Other particularities with general effects	Y	A / R	A / R	/ R	A / R

Threats and Damages	Kind and cause of threats / damages	N	A / R	/ R	/ R	A / R
	Extent and impact of threats / damages	Y	A / R	A / R	/ R	A / R
	Who or what is affected by threats / damages	Y	A / R	A / R	/ R	A / R
	Subsequent threats and damages	Y	A / R	A / R	/ R	A / R
Own resources	Forces and means already deployed	Y	A /	A /	A /	/ R
	Forces and means at further disposal	(Y)	A /	A /	-	/ R
	Additional means for disaster relief	N	A /	A /	-	A / R

Table 1: Information gathering of responders for organisation specific situational awareness

Given the fact that the different actors usually start their disaster related activities and operations time-displaced with a minimum of 24 hours (up to several days and even weeks) and taking into account the uniform need of information dedicated to creating situational awareness the need for the Common Information Space (CIS) is obvious. Connecting the various actors in disaster relief the CIS enables the timely distribution and exchange of situational awareness (SA) information. Especially the local emergency management authority (LEMA) as the responsible body in disaster management can benefit from preceding and ongoing situational assessments from various sources. On the other hand the LEMA, when having compiled a common operating (operational) picture (COP), is able to provide a homogeneous, continuously updated overview of an event. The COP is compiled throughout the whole lifecycle of the operation based on data shared between integrated systems for communication, information management and intelligence. Almost any information that is related to situational awareness has a spatial reference. This leads to the advantageous situation that this information may be send/exchanged/received automatically via the CIS between the organisations command and control (C^2) and information systems. So the current information can immediately be displayed on the organisation's digital situation maps.

Situational Awareness

Definitions:

"The perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future" [18]

"The ability to identify, process, and comprehend the critical information about an incident – knowing what is going on around you – which requires continuous monitoring of relevant sources of information regarding actual incident and developing hazards" [19]

"A result of comprehensive information collection, analysis, and dissemination in a context relevant to the authorities and responsibilities of a particular organization level" [20]

Key elements of situational awareness:

- **Perception:** Gathering or collecting information
- **Comprehension:** Interpreting information
- **Projection:** Anticipating future status

Common operating picture

Definition:

"A shared situational awareness that offers a standard overview of an incident and provides information in a manner that enables incident leadership and any supporting agencies to make effective, consistent, coordinated, and timely decisions." [20]

Key concepts of a common operating picture:

- A single set of relevant, usable information that is shared across response organizations at all levels.
- A continuously updated overview of an incident compiled throughout the incident's lifecycle.
- A graphical representation displayed on a map that visualizes various aspects of a situation in a geographical context.

SA and COP are not the same but rather dependent on each other (Figure 5). Sharing of one's own organisation specific situational awareness via the common operating picture helps others to achieve/refine their organisation specific situational awareness (Figure 6).

Figure 5 Connection of SA and COP (1/2)

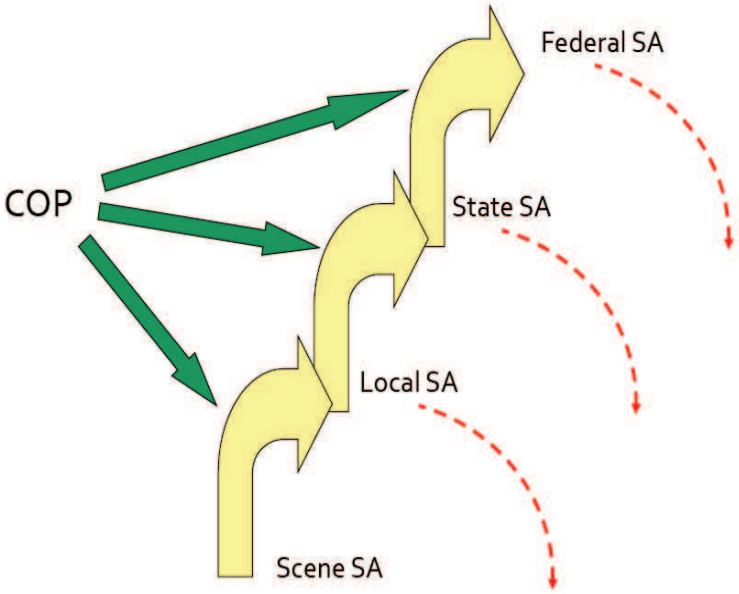


Figure 6 Connection of SA and COP (2/2) [FEMA L948 Course]

2.4. Information interoperability use-cases

Based on the previously described situation the following generic use-cases were derived:

Generic use-case I: Data/information flow from RESPONDERS to LEMA

Since LEMA is to a great extent dependent on data/information that is received from responding organisations (their organisation specific situational awareness) the establishment of information interoperability for the core data and information elements from Table 1 in the CIS will be of value.

Generic use-case II: Data/information flow from LEMA to RESPONDERS

LEMA is the entity in need/in charge of the COP. LEMA is responsible for taking the overall tactical decisions for the disaster relief operations. In short these decisions comprise WHO has to perform WHAT activities WHERE and WHEN (eventually together with WHOM). Such decisions together with accompanying information are usually communicated during briefings of the LEMA with all responding organisations. Covering this in the CIS information elements and mechanism will be a great contribution to the efficiency and the quality of a collaborative mission execution.

Generic use-case III: Data/information flow from RESPONDERS to RESPONDERS

Everything stated with generic use-case I is true for this as well. A standardised information exchange between responders is currently neither state of play nor established frequently. Nevertheless a systematic exchange of data/information regarding the *General situation in the affected area* and the *Threats and Damages* (see Table 1) would be of a great value for each and every responding organisation.

These generic use-cases have been discussed with practitioners from first responders and with selected members from the advisory board. Based on these consultations six use-cases were defined that may be implemented and tested within the proof-of-concept of the CIS concept:

2.4.1. LEMA receiving situational awareness information

Action

LEMA (Local Emergency Management Authority) collects organisation specific situational awareness (SA) information relevant for the current event/missions/operations

Steps	Remark/Constraints
LEMA establishes a session of the CIS	• several sessions are possible
LEMA invites other organisations to join this session	• denial may be possible
LEMA receives data/information provided by the invited organisations	• merging of several sessions may be possible to create a common operating picture at a higher level • the information flow is towards the LEMA

Comments

To show the ability of integrating data/information from diverse and independently operated legacy systems the setup for technical and procedural testing requires a certain number of first-responder-tools connected to at least one higher level LEMA-application.

2.4.2. Sharing of common operating picture information by the LEMA

Action

LEMA provides COP information to others

Steps	Remark/Constraints
Use case *LEMA receiving situational awareness information* executed	• several participants
SA data/information is sent by responders' systems to CIS, received by LEMA system and compiled to a COP	• requires the capability of LEMA-application to do so
SA information reported by different communication (e.g. voice) is added to the COP by LEMA	• requires the capability of LEMA-application to do so
LEMA send COP information to the CIS	• receivers may be pre-selected by the LEMA • COP info may be map layers (WMS) or dedicated items (CAP/EMSI)

Comments

COP information can be shared between tools by CIS, or e.g. superior authorities can get direct access to the LEMAs COP

2.4.3. Sharing of strategic/tactical decisions taken by the LEMA

Action

LEMA provides information about mission related decisions that where taken

Steps	Remark/Constraints
Use case *LEMA receiving situational awareness information* executed	• several participants
Mission related decisions are taken and documented	• tool used by LEMA
LEMA send decision information to the CIS (EMSI MISSION)	• receivers may be pre-selected by the LEMA

Comments

It is not intended to substitute the original way of disseminating LEMA decisions (briefings, one-to-one communication etc.) but to make mission related LEMA decisions accessible in a timely, location and tool-independent way.

2.4.4. Sharing of SA data/information between responders

Action

Organisations share situational awareness data/information amongst each other

Steps	Remark
One organisation starts a CIS-session	Or uses the CIS session started by LEMA
The organisation invites others to join this session	• denial may be possible
Session participants share their organisation specific SA amongst each other	• information filtering (send / receive) may be applied by each participant

Comments

At the first glance this seems to make very much sense for organisations from the same domain (medical serv. -> medical serv., police -> police), but also for sharing general assessment information on the operational area or the incident status (to avoid multi-assessments of the same issues).

The same info sent to LEMA (2.4.1) can also be received by responders sharing a CIS session.

2.4.5. Accessing/sharing of general and public information

Action

Sharing of information that does not specifically belong to a single CIS session

Steps	Remark
One organisation publishes information. A specific service reading this info pushes it to the CIS (weather, traffic, …)	• A global CIS session can be used • pure sender of information (service URL), info has to be retrieved by the tools
Information receiver subscribe/unsubscribe the information service	• filtering may be applied by each participant

2.4.6. Reporting from national entities to the ERCC

Action

ERCC collects information from national contact points on a dedicated topic

Steps	Remark
ERCC starts a CIS-session	• such a session might stay up and running without a defined ending
ERCC invites others to join this session	• agreements have to be set up • denial is not foreseen • technical capabilities must be available
National contact point participants distribute the intended share of information to the ERCC	• ERCC will be able to process the received information according to its predefined structure and semantically annotated

Comments

This use-case has been derived from the process in emergency response coordination center (ERCC) of the EU that has been established in order to monitor the national activities during the forest fire season during the summer month. Here European countries that are regularly exposed to forest fires during the summer send a daily report of their country specific situation (structured according to a fixed schema) to the ERCC. The interpretation of the reports by ERCC duty officers is heavily complicated by the nonconformity of semantics and language barriers.

3. Information Interoperability Standards

3.1. Interoperability standardisation in CM

The use of standards is a key driver for information interoperability in Crisis Management systems. The main factors preventing information interoperability between tools currently used by organisations in the emergency management domain, are:

- these tools are often closed, with missing software interfaces for communicating with external systems and tools
- they use, internally, proprietary data formats, therefore preventing an easy extraction and interpretation of the generated information, by other systems

The implementation of software API leveraging the use of widely adopted technologies such as Web Services and REST, is a must to address the need of interconnecting different, legacy or new systems and, as such, will be the basis of the Service Oriented Architecture of the EPISECC CIS.

In addition, the EPISECC CIS concept for information interoperability will leverage the use of existing standards, mainly XML data formats, specifically focused on emergency related data exchange. Amongst the existing activities in this direction, particularly relevant is the work of the OASIS emergency management technical committee [7] , which has developed and is continuously updating the EDXL (Emergency Data eXchange Language) family of standards. These OASIS standards are aimed at enabling information exchange to advance incident preparedness and response to emergency situation. The EDXL specifications comprise several different XML formats, each of them elaborated for sharing specific types of information, i.e.:

- EDXL Common Alerting Protocol (EDXL-CAP)
  - Alerts, Warnings, and Incidents information in PSA-to-PSAP communication
- EDXL Hospital AVailability Exchange (EDXL-HAVE)
  - Hospitals status and capabilities / resources (e.g. beds capacity)
- EDXL Resource Messaging (EDXL-RM)
  - Resources (humans, equipment, vehicles) status, deployment, availability, quantity
- EDXL Situation Reporting (EDXL-SitRep)
  - Information about the evolving situation, the management status, observations from the field
- EDXL Tracking Emergency Patients (EDXL-TEP)
  - Emergency patients status and tracking (e.g. hospital admission, release, …)
- EDXL Distribution Element (EDXL-DE)
  - Envelope for the distribution of different payloads (including messages in other EDXL standards)

Apart of the OASIS standards, another relevant standardisation effort in the Crisis Management domain that will be implemented in the EPISECC PoC is represented by the EMSI (Emergency

Management Shared Information) standard. Conversely to the OASIS approach, EMSI specifies a single XML data structure for sharing different type of emergency related information (events, resources, missions). This approach is aimed at providing general situation information, therefore improving the situation awareness of all involved emergency organisations (see also following section 3.4).

Situation awareness in information interoperability scenarios may be strongly improved by the availability of geographic maps where relevant geo-referenced information can be visualised, and organised as geographic layers. In this context, the project will leverage the ongoing work of the OGC (Open Geospatial Consortium), and therefore the availability of existing standards for geographical information / layers sharing, such as WMS and WFS.

Finally, location information sharing and resources location tracking is another crucial aspect for creating a Common Operational Picture, including the position of resources and related ongoing activities during emergency response. A well-known, widely adopted standard is in this case is represented by the MLP (Mobile Location Protocol), which will be adopted and demonstrated during the EPISECC PoC, too.

The EPISECC CIS prototype implementation will mainly focus on the abovementioned standards EDXL-DE, CAP, EMSI and MLP, as it will be described in more details in the following of the present chapter, although the developed CIS Adaptor concept will allow the integration of potentially any other XML standard for emergency data exchange.

3.2. Information envelope EDXL DE

EDXL DE V 2.0 is defined as a standard draft [8] issued by the OASIS Emergency Management TC [7] . It provides a standard message distribution format for data sharing among emergency information systems, and it serves two important purposes:

(1) The DE 2.0 allows an organization to wrap separate but related pieces of emergency information, including any of the EDXL message types, into a single “package” for easier and more useful distribution;

(2) The DE 2.0 allows an organization to “address” the package to organizations or individuals with specified roles, located in specified locations or those interested in specified keywords.

Every message exchanged in the Common Information Space shall be encapsulated in an EDXL DE envelope in order to identify and provide information to enable the routing of encapsulated payloads, called Content Objects. One EDXL DE may contain several different Content Objects if they belong to the same sender, time stamp and descriptive information given in the EDXL DE.

The authentication and authorization of information in the CIS should be handled by the data provided in the DE.

3.3. Common Alerting Protocol CAP

The Common Alerting Protocol CAP [10] is a standard provided by OASIS [6]. CAP is a simple but general format for exchanging all-hazard emergency alerts and public warnings over all kinds of networks.

The CAP protocol is used in its current version V1.2 in the Common Information Space (CIS) in order to communicate alerts, warnings and notifications from any application that detects a critical situation (e.g. call center, sensor system, mobile device) to all interested systems (e.g. common operational picture, public alerting device).

The CAP message is sent embedded in the EDXL DE envelope (EDXL CAP message). The consistency of redundant data in the envelope and the payload (CAP message) has to be guaranteed by the sending adaptor. It is possible that the sender information differs between EDXL DE and CAP, e.g. in case of forwarded messages. For authentication and authorization purpose, always the information in the envelope counts and the sender is responsible to maintain confidentiality of forwarded messages.

The sender of CAP messages is further responsible to be in line with the standard and to avoid sending corrupted messages. The receiver of CAP messages shall accept all features defined in the standard. If the receiver can't process a CAP message, it should reply with a CAP error message to the sender (status=system, msgType=error).

In addition to the data elements defined within the standard, additional information might be provided in <parameters>. These parameters can be specified in CAP profiles to be agreed upon between specific applications, and might be ignored by applications not concerned with the profile. The Commission may recommend a European CAP profile that defines specific use of optional attributes and value lists in the context of European CDM (example: see Australian CAP profile, http://docs.oasis-open.org/emergency/edxl-cap1.2-au/v1.0/cs01/edxl-cap1.2-au-v1.0-cs01.doc)

3.4. Emergency Management Shared Information EMSI

The TSO (Tactical Situation Object) was developed under the EU-FP6- OASIS project (2004-2008) and approved as a CEN Workshop Agreement (CWA) in October 2008 [11].

ISO/ TR 22351 Technical Report: Societal security - Emergency management - Message structure for exchange of information [12] adopted the TSO as the message structure and data dictionary for the exchange of situational awareness information in emergency management scenarios in 2012 and renamed it as EMSI (Emergency Management Shared Information).

The EMSI is used to transfer the view of an emergency situation as seen by a particular observer at a particular time to another observer, thus contributing to the situational awareness of the various parties regarding a given disaster or crisis event. The message can be used peer-to-peer for observers (either from the same or different organisations) at the same level of the command hierarchy, or used to send information up and down the hierarchy.

The EMSI message follows an XML structure (that is embedded into an EDXL DE envelope for its transfer) based on a concrete object model whose main entities are:

- the events, understood as something that takes place which an agency should respond to (e.g. a natural or man-made disaster),
- the resources available to support or help in the response to the events, and
- the missions aimed at handling the events and thus reducing their impact.

The objective of the EMSI specification is to ensure that the semantics of an individual message are unambiguous; however, it does not prescribe how to merge messages or how to transfer them.

3.5. GIS standards

Geospatial information to be handled as a map layer can be embedded in the EDXL DE as Content Object "OtherContent" (non-XML). The geospatial information has in this case to follow the selected standards.

For the implementation of the CIS the following standards were selected:

3.5.1. WMS (Web Map Service) and WFS (Web Feature Protocol)

WMS and WFS are standards defined by OGC [13]. The data (map information) is provided as web service. The information transmitted in the CIS is just the URL where the service can be consumed. The service itself will not be routed over the CIS. All necessary meta-data and service descriptions needed for the consumption of the service have to be exposed and can be queried at the service location. The scope of services is assumed as provided by GeoServer [13].

3.5.2. GeoTiff

GeoTiff is a public domain raster image format which provides geographical metadata. As GeoTiff files tend to be very large, only small and limited images shall be transmitted in this format. For large images (e.g. satellite or aerial images of a wider area, the image provider shall render the images and transform them into WMS.

3.5.3. ShapeFile

ShapeFile is the de facto standard format for vector data. One "shapefile" consists of more than one physical file (main file containing geometric objects like points or polygons, the data-file which stores additional data for each geometric object, the index-file holding an index to each record in the data-file. Depending on the used tools other accompanying files might exist, e.g. holding spatial projection details.) So shapefiles are handled as archives (ZIP) containing all files belonging to one "shapefile".

3.6. Other Content

The EDXL DE Content Object "OtherContent" (non-XML) may contain any resource or media (base64 binary file or link to them) which type can be identified by the used MIME type (e.g. pictures, text, tables …). The ContentDescriptor of the EDXL DE ContentObject Element [9] should give additional indication on the content and meaning of the message.

3.7. Mobile Location Protocol MLP

The Mobile Location Protocol (MLP) is an application-level protocol for receiving the position of Mobile Stations (MS: mobile phones, wireless devices, etc.) independent of underlying network technology. Basic MLP Services are based on location services defined by 3GPP.

In context of EPISECC CIS, MLP is used for reporting the location and the status of mobile devices of the responders in the field, and so for tracking of resources.

More details of the protocol MLP and its usage for public safety networks like TETRA are outlined in deliverable D5.1 Protocol & Network Interoperability [4].

3.8. Outlook to other standards

Some more standards will be useful for different use cases that are not worked out in detail in EPISECC. The CIS architecture easily allows extension to any standard that is based on XML. If there are tools able to handle the information provided by the stand, just the corresponding Connectors have to be created.

Two examples of standards potentially useful in CM are PFIF and SOS.

3.8.1. People Finder Interchange Format PFIF

The initial data model on which the first version of PFIF was based is due to the CiviCRM team, David Geilhufe, and Kieran Lal. It was developed further during several disasters (hurricane Katrina 2005, Haiti earthquake 2010, Tohoku earthquake 2011). The current version PFIF 1.4 [15] originates from 2012.

The PFIF standard consists of a data model and an XML-based exchange format for sharing data about people who are missed or have been displaced by natural or human-made disasters.

It is designed in a way promoting convergence: convergence of people who seek the same person, convergence of information about a person obtained from various sources, convergence of duplicated data, and ultimately convergence of missing people with their loved ones.

Another very important design principle is that all data is traceable. Since data comes from sources of unknown reliability and accountability, information on the origins of data should be maintained, to help users ascertain its trustworthiness.

PFIF relies on the notion of multiple information repositories. Each record belongs to an original repository. The record may be copied to other places, but the original repository remains the authority on the record. Only the original repository should ever change the contents of a record.

Data is gathered and presented by independent aggregators. Each aggregator has its own perspective on the world and is responsible for choosing which data sources to trust. Because multiple records might refer to the same person, PFIF allows such records to be associated with each other but each aggregator must make its own decisions about which records to associate.

PFIF guarantees that it is possible to resolve multiple copies of the same record despite the different path it may have been imported.

Finally, it makes sure that all data is stored in a non-localized, universal format (UTC time zone).

3.8.2. SensorML, Observations and Measurements specification O&M

The OGC Sensor Web Enablement (SWE) initiative specifies a suite/family of standards that can be used as building blocks for the interoperable integration of sensor data. The OGC Sensor Observation Service (SOS) provides access to sensor descriptions and sensor observations. The SOS specification leverages the Observations and Measurements (O&M) specification to encode observations and the Sensor Model Language (SensorML) specification to encode sensor descriptions. Both of these formats are based on XML.

Sensor Observation Service (SOS): This standard defines a Web-based interface (Web service) that allows querying observations, sensor metadata or representations of observed features. Furthermore, this standard provides means to register new sensors or remove existing ones and can be used without a-priori knowledge of domain-specific application schemas. It also defines operations to insert new sensor observations. Observations are returned encoded as O&M Observations, while information about sensors is returned encoded in SensorML.

Sensor Model Language (SensorML): The OpenGIS® Sensor Model Language Encoding Standard (SensorML) specifies models and XML encoding that compose a framework through which the geometric, dynamic, and observational characteristics of sensors and sensor systems can be defined. There are many different sensor types, ranging from simple visual thermometers to complex electron microscopes and earth observing satellites. These can all be supported through the definition of atomic process models and process chains. Within SensorML, all processes and components are encoded as an application schema of the Feature model in the Geographic Markup Language (GML). The main purposes of SensorML are to:

- provide general sensor information in support of data discovery
- support the processing and analysis of the sensor measurements
- support the geo-location of observed values (measured data)
- provide performance characteristics (e.g., accuracy, thresholds, etc.)

The characteristics of SensorML play an important role to Sensor Integration Service. Intelligent sensors are utilized and integrated seamlessly to sensor data fusion. The accuracy of data that arrive in the control centre, the analytical description of sensor types and the support of geo-location features makes SensorML an invaluable solution for Sensor Integration Services.

4. The Common Information Space (CIS) Architecture

4.1. CIS concepts – overview

In the SOA context, every application can offer data (information provider) and/or receive data (information consumer), and has no effect on the other partners in the Information space. That means the applications have to expose services that send/receive data in standardized formats and via defined communication protocols without the need for particular interfaces between dedicated partners. If the application uses data communication protocols that are not supported by CIS, the protocols have to be converted by adaptors.

These services have to use standard data formats (e.g. CAP, EMSI, WFS, KML …) and – if necessary – transform proprietary data formats of the application into the standard formats of CIS (adaptor function).

The middleware and the Integration Framework have to assure that every consumer can subscribe on the type of information he is interested in, and that all submitted information is transported properly from the provider to the consumers.

When an application joins the CIS, it has to register its services in order to enable other applications to address the offered services. A metadata model enables the applications to find out services that fit with their own purpose.

The registration process might be subject of authorisation and role concepts (*to be elaborated*) in order to establish a protection hierarchy and to prohibit unauthorised access to sensitive data. This mechanism might be combined with encryption techniques.

Beyond the data connection (network interoperability) and data formats (syntactical interoperability), the interpretation of the content (semantic interoperability) is crucial for automated information exchange. Therefore key terms and taxonomies (e.g. resource type, task mission, event category) have to be translated from the proprietary form of the provider to a standardised form in the CIS and back to the proprietary form of the receiver. This double-translation inevitably leads to loss of information. Therefore it shall be possible to consume the original data in addition to the standardised if both applications speak the same “language”.

The Common Information Space is a data sharing platform but not a data repository. The ownership of the data stays with the applications. The Common Information Space itself doesn’t have any business logic. The validation, interpretation and processing of the transported data is part of the applications. Nevertheless, value added services can be attached to the CIS and made available for authorised users (e.g. logging and legal recording, reporting, monitoring …).

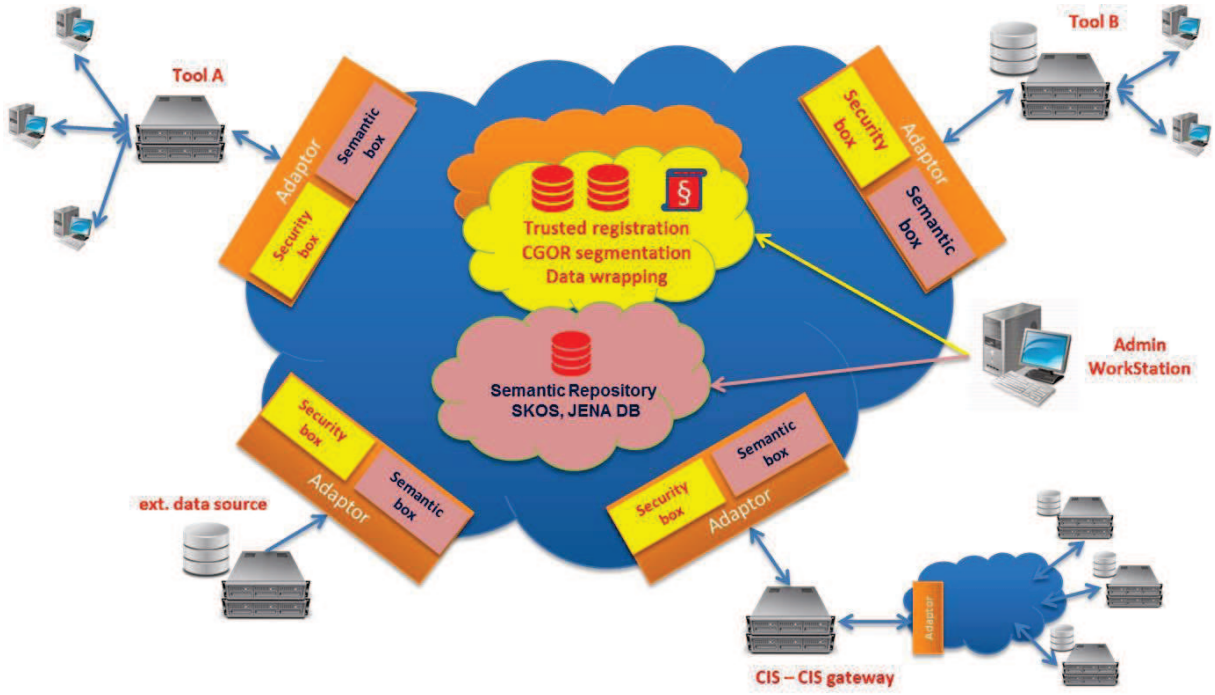


Figure 7 : System integration via Common Information Space

4.2. CIS requirements

The following list of requirements is derived from statements of disaster management practitioners and the analysis preformed in WP3 (inventory of events/disasters) and from the experience of WP5 partners with similar information systems.

#	Requirement	Prio[1]	Category
1.	A common information space (CIS) must handle the sharing of data among the tools, providing publish-subscribe and request-response mechanisms	1	Functional
2.	Synchronization of shared data between applications should be supported after network interruption	2	Functional
3.	The CIS must ensure that the exchanged data are syntactically correct	1	Functional
4.	Interoperability on the semantic layer should be ensured by using common taxonomy and taxonomy based mapping tables	1	Functional
5.	It shall be possible to add value added services (VAS) within the CIS. e.g. for providing aggregated data for reporting, system monitoring …	3	Functional
6.	The information space shall not store data for operational purposes or provide business logic; it shall only connect systems and transport data	1	Functional

[1] Prio: 1 high – 2 medium – 3 low

#	Requirement	Prio[1]	Category
7.	Authentication will be required for a tool to connect to the CIS	1	Security
8.	CIS should provide a certification authorization mechanism so that only tools with the required security level would be granted access to classified data	2	Security
9.	CIS should provide audit and logging tools	2	Security
10.	Scalability should be supported	2	Performance
11.	A "situation manager" shall be able to open up and deploy a specific information space for a specific crisis situation and to invite participants	1	System Mgmt.
12.	CIS should offer the possibility to be administrated so that topology and configuration could be updated	2	System Mgmt.
13.	Interfaces shall be technology-agnostic (shall not depend on the technology used by the respective tools)	1	Technical Constraints
14.	Common standard formats for the exchange of information in the CIS must be agreed; it will be used by all tools involved by means of adaptors whenever needed	1	Technical Constraints
15.	In addition to standardized messages, the consumption of the original format should be enabled to avoid data loss by double conversion	3	Technical Constraints

Table 2: CIS requirements

A more detailed disquisition on legal and ethical requirements derived from WP7 results is given in section 5.6.2.

4.3. CIS adaptors architecture

The CIS adaptors link the participating tools to the Common Information Space. For every tool and every used data protocol, a specific adaptor has to be implemented. Adaptor templates will be provided by the project team in order to enable the tool providers to write their adaptors in an easy and fast way.

The adapters stay in the responsibility and run within the secured network environment of the tool owner. Every access to the data hosted by the adapter is monitored by the authorisation concept implemented in the adaptors and is recorded for audit and tracing purposes.

Every adapter consists of three parts:

A. **CIS Connector**: manages the communication with the tool and translates proprietary protocols to standards, and back. The Connector is written by the tool provider based on the EPISECC Connector template.

B. **CIS Core**: manages central functions in a uniform way. The application of security policy and the semantic matching is controlled by the Core (Security and Semantic boxes)
Value added services can be integrated in the Core (available for the whole system).

C. **CIS Distributor**: manages the connections inside the CIS and the data exchange with the other Adaptors in the CIS.

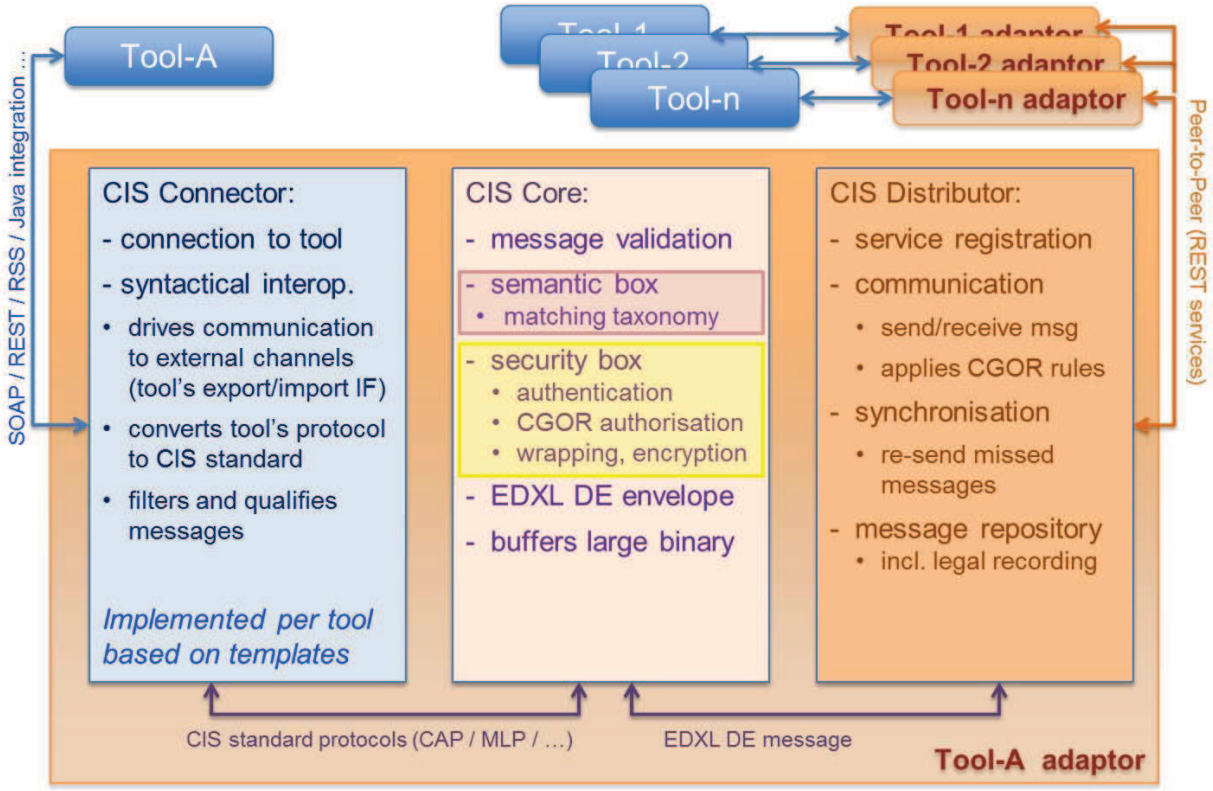


Figure 8 : CIS Adaptor architecture

4.3.1. CIS Connector

The CIS Connector handles the communication on the side of the tool – that means it covers all code specific to the protocols the tool uses. Therefore it has to be assembled and configured by the tool owner or manufacturer based on the adaptor template.

The template consists on components providing the following functions:

- Network connectivity module receives/sends messages from/to the tool according the used network protocol.

  Templates for REST, SOAP and RSS connections will be prepared in the first step. The tool owner has to maintain network configuration tables with the addresses of the tool services to be connected.

- Data format converter transfers proprietary data formats of the message to/from the standard messages exchanged in CIS (this step may be bypassed if the tool already uses the appropriate standard).

- Standard translator replaces proprietary key values and enumerations by standardized ones and vice versa, based on translation tables to be provided by the tool owner. This applies only for code value sets mandatorily listed in the standard definition (e.g. CAP status, msgType, category …) in order to form a correct standard message.

(Do not confuse with the semantic annotations provided by the Semantic Box which are based on mapped taxonomies.)

- EDXL DE generator assembles the parameters for the EDXL Distribution Element that envelops all messages distributed in CIS. The template will provide a minimum set of default values that might be extended by the developer of the CIS connector. I.e. security related parameters can be added dependent on the message content.
- Filtering of received messages based on EDXL DE parameters.
  Filters might be extended by the developer of the CIS connector also based on the message content.
- Logging, for debugging purposes only.

4.3.2. CIS Core

The CIS Core can't be modified by the adaptor provider. It manages central CIS features, partly based on the EDXL DE parameters transferred:

- Authentication assures that incoming messages originate from a trusted partner application, according to the service registration.
- Authorization services control the flow of information and protect sensitive data from unauthorized access. Appropriate encryption mechanisms are defined and implemented in the CGOR concept (see section 5.3).
- Validation of the transferred messages assures the formal correctness and application of standards. Improperly formatted messages will be rejected.
- EDXL DE Wrapper packs the information into an EDXL Distribution Element (envelope) that adds meta-information to the payload message.
- Object Buffer function stores large binary objects (message attachments) in an accessible store (e.g. FTP server) and replaces them by the URI in the message.
- Message Buffer stores all outgoing messages in order to enable the partner applications to query previous messages, e.g. in the case of sync after network interruption.
- Value Added Services (optional plug-ins) may make use of the transferred information e.g. for message logging, auditing, reporting or statistics.

4.3.3. CIS Distributor

The CIS Distributor manages and synchronizes the message exchange between the partner applications in the CIS. Two different architectural approaches were investigated:

A. Enterprise Service Bus based solution, e.g. Apache KAFKA. The middleware manages message transmission, buffering, service registration and consumer groups.

B. Peer-to-Peer communication architecture, using a service registry server (e.g. Netflix's Eureka), message database (e.g. Mongo DB), and specific REST interfaces for registering and message distribution.

In EPISECC we decided on the distributed peer-to-peer solution in order to avoid the central ESB server as a potential single point of failure, and the need for providing and administrating a powerful central server, requiring a centralised administration instance.

The Distributor may support two different distribution mechanisms:

1. Push services (publish – subscribe): The information provider posts a message addressed to a CGOR, or to all CIS participants (global CGOR). The distributor sends the message to all authorised information consumers (CGOR members' Adaptors) that have been registered on this service (information type, topic). Push services don't guarantee if the message actually is received by all subscribers. A synchronisation mechanism can assure that missed messages are delivered at a later stage.
2. Pull services (request – response): The information consumer asks a dedicated information provider for defined pieces of information. The CIS will support criteria based on the EDXL DE structure, e.g. time sent, target area, content key word. The information provider (CIS Core) decides if the requestor is an authorized information consumer, and answers the request with appropriate messages that have been stored in the Message Buffer. The request-response method can be combined with a notification mechanism that publishes the existence of a new message but not the content.

5. CIS design concepts

The following sections describe in detail solutions designed for specific challenges in the scope of secure and understandable information interoperability enabled by the common information space.

5.1. Semantic interoperability

5.1.1. Challenge: understandable information

In general terms, *Semantic interoperability* is defined as the ability of computer systems to exchange data with unambiguous, shared meaning. Differently from *Syntactical interoperability*, *Semantic interoperability* is not concerned with the formatting or packaging of the data, but with the simultaneous transmission of the data together with its meaning (semantic), by linking proprietary data elements (key terms and concepts) to a common vocabulary of terms and concepts. The need of using a shared vocabulary for ensuring a common understanding of the exchanged information, is highlighted in the simple scenario depicted in Figure 9 below. Different end users' tools for disaster management, from Italian Fire Brigades, Greek Fire Brigades, Greek Police, and German Red Cross in the considered example, might be able to seamlessly exchange information, provided each of their systems are connected through suitable software interfaces / API. These tools, might also be able to extract and read the information shared by each of the other tools, provided the involved organisations have agreed on common formats / protocols for packaging of the shared data (*Syntactical interoperability*), therefore using e.g. standard XML data formats. Still, the proper, mutual understanding of the information may be compromised by the fact that each organisation represents relevant terms and concepts (e.g. incident codes, resource codes) using a proprietary encoding. Tools for *Semantic interoperability* are therefore needed in order to map, at configuration time, proprietary terms and concepts in the corresponding common ones (EPISECC Taxonomy), and to provide, at runtime, the matching / conversion between relevant terms and concepts.

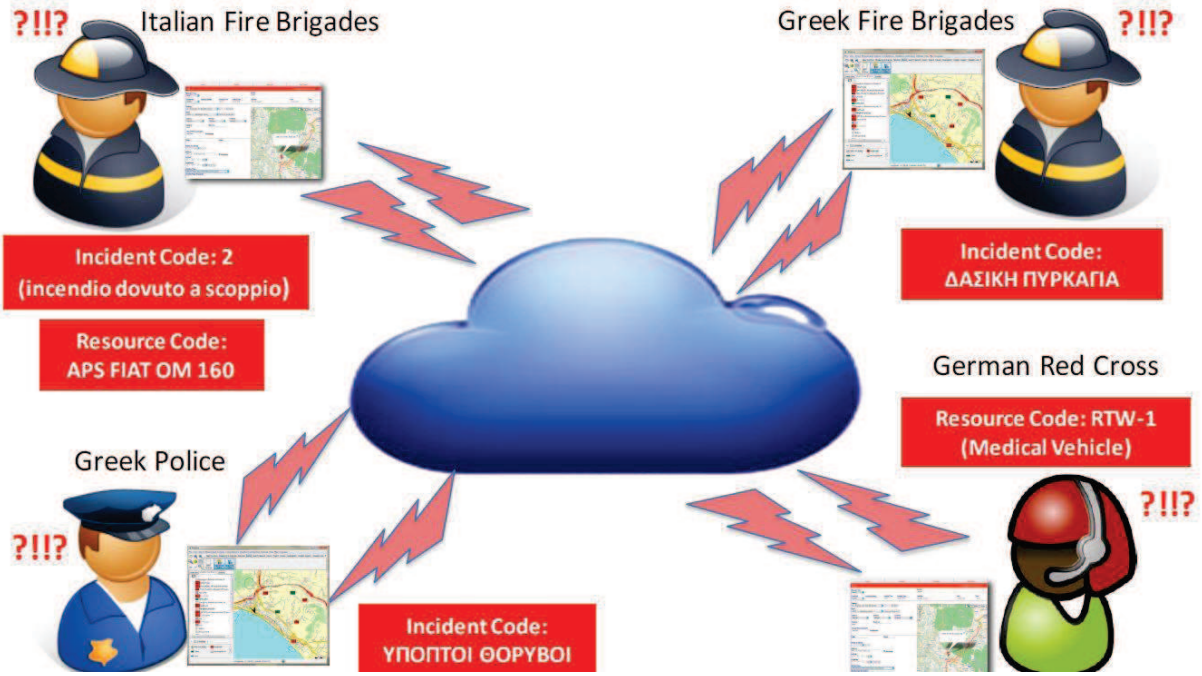


Figure 9: Challenge of sharing commonly understandable information

5.1.2. Semantic Box

It has been described in WP4 (see deliverable D4.3 – Data Model) how the Semantic Interoperability in EPISECC will involve the following steps:

1. EPISECC Taxonomy building: population of the EPISECC Taxonomy, including all relevant terms, concepts and codes used in disaster management, and insertion of the common EPISECC taxonomy into the Semantic Repository.
2. Further population of the Semantic Repository: Semantic Repository needs to be populated with other existing structures containing concepts, like e.g. EMSI, and other common dictionaries, terminologies or taxonomies and cooperating organisations' concepts.
3. Semantic mapping: at configuration time, there is the need to perform the mapping between terms and cooperating organisations' concepts from existing structures and the EPISECC taxonomy, and store this mapping into the Semantic Repository.
4. Semantic matching: at runtime, i.e. during information sharing through the CIS and in order to be sure that the most relevant information can be understood by everyone involved in the communication, there is the need to match key concepts, previously stored in the Semantic Repository and selected from the shared information, with the EPISECC taxonomy, and further with the concepts of other organisations' involved in the CIS. Semantic matching uses relationships between concepts established in the previous step.

The first 3 steps are in first place realised by the EPISECC technical team through the use of Protégé Desktop tool [16], used for taxonomy schemas creation and population, and for mapping of concepts from different taxonomies. The content in the Protégé tool is then transferred to the Semantic

Repository, which is a Triple Store (TDB) realised using the Apache Jena Framework [17]https://jena.apache.org/).

In the actual EPISECC and CIS concept, the EPISECC taxonomy will be a live, dynamic structure that will be constantly updated with new terminologies and with new end users' concepts, together with their mapping. For ease of maintenance, end users will be able to visualise, query and modify the existing Semantic Repository, by the mean of a graphical user interface provided with back-end functionalities for accessing the Semantic Repository itself.

The Semantic Repository is part of the EPISECC Semantic Box which also includes:

- The Semantic Matching Web Service
- The Semantic Repository Wrapper Service

The Semantic Matching Web Service provides a SOAP Web Service API to:

- validate standard messages
- request semantic matching of end users' concepts with the EPISECC taxonomy, and matching between end users' concepts (via the EPISECC taxonomy, or directly in case a direct mapping is provided)

The Semantic Repository Wrapper provides a REST API to convert matching requests from the Semantic Matching Web Service into SPARQL queries to the Semantic Repository. It is based on the Apache Jena Fuseki service.

5.2. CIS Distributor: Distribution and synchronisation of information

In the EPISECC architecture, the distributor represents the central part of the CIS communication chain. Its purpose consists in distributing messages to clients based on their CGOR membership. Like a postman, it just sends a message without worrying about its content. Yet, it considers communication policies like recipient's accreditation level, information disclosure etc. which can be defined by the sender when joining a CGOR.

5.2.1. Design consideration

Three important aspects were considered when designing the CIS Distributor: how it interfaces with the Core, how coupled it is with its associated CIS Core, and its ability to build communication contexts called data partitions. Data partition allows to segment data only according to the CGOR it is addressed to.

Concerning the interface, REST interfaces were chosen because of their simplicity of the API design and integration.

About the coupling, it represents the level of interaction between two pieces of software. The less these pieces are connected, the more independent they are. In order to take the coupling part of the challenge, the design team opted for a micro-service architecture as it enables to:

- **Ease the integration process between CIS Core and Distributor**. It offers great flexibility and allows better technical maintenance since Core's and Distributor's implementations may be replaced entirely without having to care about other services provided their common interfaces remain the same. This is important especially in this current calibration full of technical adjustments and requirements evolution.
- **Ensure software resilience**. Indeed, resilience is built upon error recovery and corrections due to software heavy solicitations. The micro-service architecture allows to run several Distributors; each unit of distributor being called an instance. Then, Distributor instances can either relay themselves in case one instance falls following an error or balance the load in case they are heavily requested.

While focusing on data partition expectations, we looked for a suitable communication medium based on current development trends mixed with our own software. Two main possibilities arose:

- **Message Oriented Middlewares (MOMs)** which is used as root for frameworks like Kafka, RabbitMQ or Spring AMQP. Communication MOMs are software infrastructure design to support the sending and the receiving of messages between distributed services through Queues. Queues are storage zones that welcome messages until they are processed by consumers. Basically, when a sender sends a message, it can continue to do other work until the receiver retrieves it from a queue and processes its content. This defines an asynchronous communication.

- **Web-Services with the usual REST and SOAP**. In the Web service case, the sender posts a message (as request parameter) to its recipient using a specific URL. The recipient processes the message and returns his response when he is done. This illustrates a default synchronous communication. Moreover, these web services mostly rely on HTTP protocol for transmission.

Finally, we chose to rely entirely on the **REST web services** for the moment as the message oriented middleware may be efficient but requires deeper specifications and more complex implementation, which were not thorough when we mocked up the distributor.

Once we decided to go with REST interfaces with micro-services, it was easier to incorporate the CGOR-based communication context in these REST interfaces. Indeed, when the core wishes to address a particular CGOR, it just has to call the distributor interfaces with the CGOR as a path parameter.

For example, generic patterns like

[HTTP Method] /cgors/{CGORIdentifier}/{CGORAttributes}*/{CGORAllowedOperation}* help to generate requests like **GET** /cgors/**episecc**/participants/ to list all participants of CGOR episecc.

The **{CGORIdentifier}** parameter represents the identifier of a CGOR. It can be either its id number or its name. The **{CGORAttributes}** goes for specific CGOR attributes like the participants, messages, resources etc. involved in the CGOR. It should be noted that attributes may also have attributes. For

example, when we need information about all **Hitec**'s committed resources in **episecc** CGOR, we can request its distributor to **GET** /cgors/**episecc**/participants/**hitec**/resources/.

5.2.2. Implementation

As the distributor endorses his role to establish communication between CIS partners, we also preferred to split its features into small independent micro-services that rely on REST communication. Each split has its type of inputs, type of outputs and more importantly its own data storage:

The CIS Directory Structure micro-service (CIS-Struct or Partition Service) provides information about the EPISECC partners and the communication context (CGORs) details they are involved in. It welcomes request parameters and returns JSON responses containing network information. Since such information show the existing interactions between CIS partners, we chose to host these data inside graph data store **Neo4J** (refer to the graph theory), where vertexes (nodes) picture organizations, CGORs user or resources and edges represents the links between them. Using graphs permits to limit the cost of the researching contextual information. For example, when you look for a list of participant, a graph data store returns you the list of organizations that have a participation link with a CGOR. In usual relational database, such connected search would have required several costly join operations to deliver the same result that is delivered instantaneously with Neo4J.

The partition service also includes a synchronization feature that replicates data among other partition services in CIS. The replication operates as shown in the following sequence as any client tool event shared in CIS is expected to be declared to a **partition authority** (CIS partition). Once stored by partition authority, the partition authority notifies the CIS partners about the updates to be made. This allows synchronization between partition services.

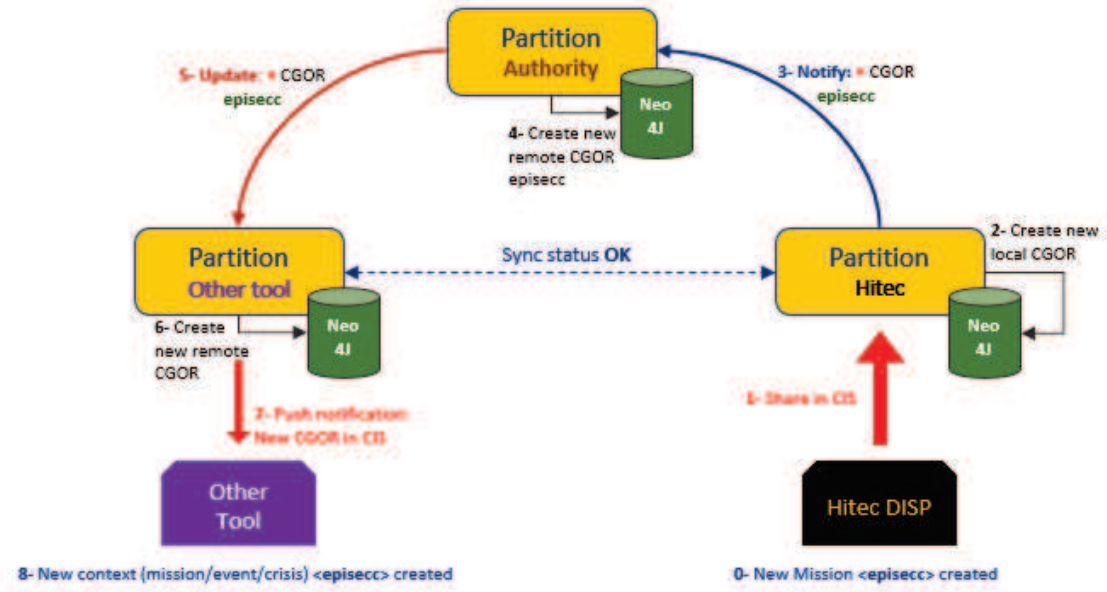


Figure 10: Partition micro-service synchronization in CIS

The CIS Directory Agent micro-service (CIS-DA or Discovery Service), based on Netflix's **Eureka** service, is another central function of the distributor. In our micro-services architecture, **Eureka server registers micro-services within the adaptor only** and helps to access these services just by using their names. Tasks like load-balancing, fail over or service maintenance remain invisible for the user. Eureka can be perceived like a DNS server, but unlike a DNS server, requests are not routed to an unhealthy or unavailable micro-services. Plus, these micro-services addresses are not exposed to the external world.

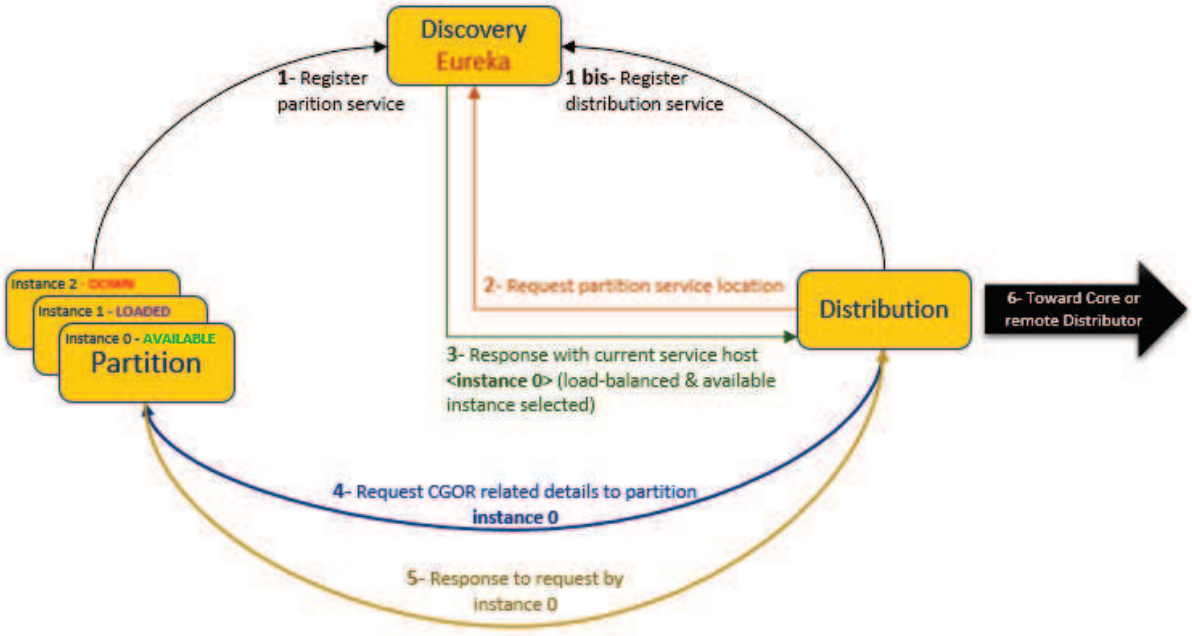


Figure 11: Example of partition service discovery via Eureka

The Distribution service is the core functionality of the distribution mechanism. It forwards any incoming EDXL DE messages to their designated recipients and saves them in a document database (MongoDB) regardless of their delivery status. Once a message is received, there are 2 levels of distribution:

- *A distribution to remote distributor*, required when a client sends a message that is addressed to a recipient that does not share the same distributor as current organization. The current distributor just forwards the same message by changing the host to the appropriate distributor.

- *A distribution to core*, that renders the message to client's Core. At this moment, the message is pushed as body parameter on a callback interface defined in the core configuration properties file.

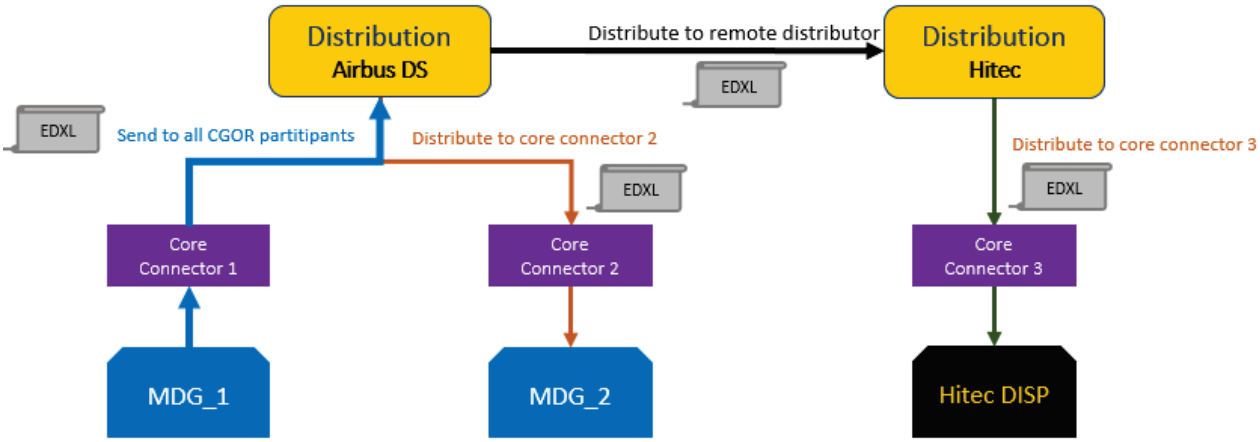


Figure 12: Distribution to client or to remote distributor

Finally, after regrouping all the micro-services and the interfaces together, we obtained the following distribution mechanism:

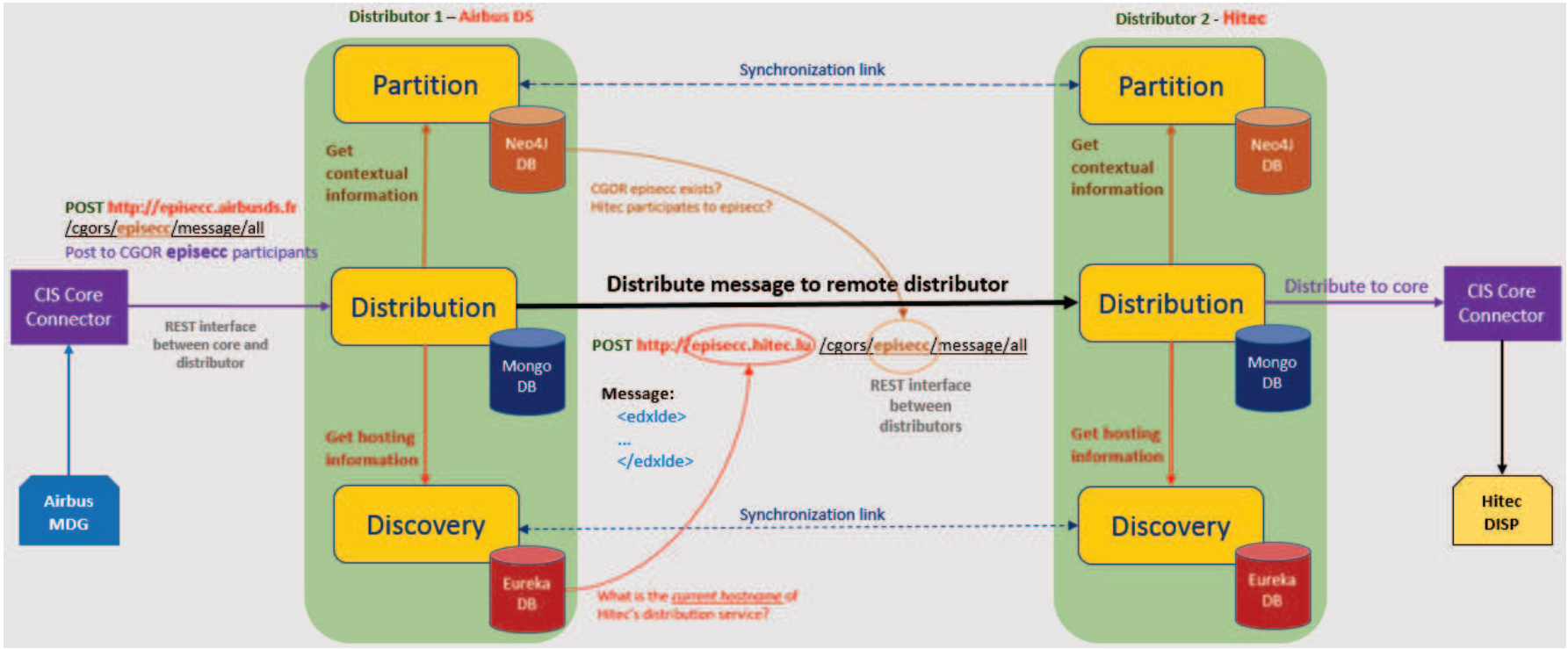


Figure 13: Complete Distribution mechanism: functional perspective

5.3. Security and authorisation

The CIS-Security architecture MUST provide confidentiality and integrity of the data exchanged on the CIS. It should also provide accountability and non-repudiation features. Availability of the services, whilst it is very important, will not be addressed in this section. However, we shall provide recommendations on how to mitigate threats on the availability.

In this section we propose solutions that can address the security requirements.

Main design requirements are scalability and flexibility of the design to work in different scenarios. Specifically our design should address two use cases

- Use Case1: Presence of CIS-Administrating Entity

  This Use Case assumes having an entity administrating the CIS and acting as a certificate authority (CIS Accreditor) that is responsible for registering new emergency organisations, issuing public key certificates to members, and acting as a trust anchor. For this Use case, a Public key infrastructure can be used with the CIS-Administrating entity providing the role of the certificate authority.

- Use Case2: Absence of CIS-Administrating Entity

  This Use Case assumes the absence of an entity administrating the CIS and the absence of certificate authority. Emergency Organisations bilaterally authenticate each other and exchange information using CIS technology. PKI is not possible in this case. Thus, alternative methods for authentication and key establishment-such as Identity Based Encryption, Short Authentication String exchanged over secure parallel channel, pre-shared key, Deffie-Hellman Key generation protocol, etc.-are used. These methods will be explained later. However, the scope will be on UseCase1.

The design should accommodate for two types of external tools

- Legacy External Tools:

  These are existing tools providing and receiving services and information that are exchanged by the CIS. These Tools cannot be modified to connect to the CIS. The CIS is transparent to these tools. Many of the functionalities the CIS can provide are limited by the tool capabilities.

- External Tools specially built for the CIS:

  These are tools specially built to use the full functionalities that can be offered by the CIS.

5.3.1. Trust Model

Trust should be established between all components and users of the CIS architecture. These can be divided into three layers of interoperability

- Information Layer, which includes

  - Individual users
  - Sensor readings to the CIS
  - Organisations
- Automation layer, which includes
  - CIS Software components
- Communication and hardware layer, which includes
  - Communication network
  - Mobile devices

Trust from the Emergency Organisation perspective is rooted, when an authorised employee-possibly administrator- of the organisation obtains and installs an official CIS-Software Package (CIS-SW) (see sequence diagram). An official CIS-SW package should be signed and certified of its integrity and would provide assurance that it does what it is supposed to be doing. It should also include root certificates to manage future updates and patches. The official CIS-SW includes all the necessary CIS software components, which are needed by an Organisation to connect its External Tools to the CIS and exchange information and services with tools connected to the CIS.

Having established a trust relationship between the Emergency Organisation tool and its own CIS components, we need to establish trust between emergency organisations. There can be two possible use cases

- With a CIS Accreditor

This is a trusted entity that is responsible for registering Emergency Organisation and issue them certificates signed by its own key. The CIS accreditor issues certificates to emergency organisation after she is satisfied with authenticity of the registering emergency and the employee acting on its behalf. Emergency organisations can trust each other when each one can present a valid unexpired certificate signed by the CIS-Accreditor. A CIS-Accreditor will provide identity assurance, maintain certificates, and certificate revocation list. There can be more than one CIS-Accreditor, where each CIS-Accreditor's root certificate should be kept by all CIS-members. There can also be federated operators who are signed by a CIS-Accreditor and delegated to authenticate emergency organisation (forming a certificate chain of trust).

- Without a CIS accreditor

This occurs in the absence of a CIS accreditor. This situation may happen when there is no entity that is available to provide this role, or when there is communication breakdown that prevents connecting to a CIS-accreditor. In this case, the trust is established between any two organisations by having an authorised employee from each organisation mutually authenticate each other (for example using the phone, producing their ids in face to face, etc.). There should be guidelines on how authorised employees can authenticate each other. Note that it is still possible that these two organisations once they established a trust between them, can start extend the chain of trust by acting as reliable broker.

Trust between CIS-SW components is established by allowing only signed CIS-SW to interact with each other. As mentioned above signed CIS-SW would provide an assurance of the integrity of the package. This can also be extended to control the trust with not updated and vulnerable CIS-SW components.

So in summary we have established trust between

- Emergency Organisation <-> Authorised Employee

Established through contract and employee is issued id, etc. (Outside scope of the CIS)

- Authorised Employee <-> CIS-SW Components

Established through the Employee obtaining an official CIS-SW Components, while authentication mechanism may be used to authenticate and authorize employee

- CIS-SW components <-> CIS-SW Components

Established through checking that all components are authenticated (hashed and signed) and updated against serious vulnerabilities

- Authorised Employee <-> Authorised Employee

Established directly through the use of i.d., or indirectly through PKI and certificates

- Emergency Organisation <-> Emergency Organisation

Established through CIS-accreditor or through the direct employee to employee

Thus we have a complete and unbroken chain of trust.

5.3.2. Registration and authentication of CIS participants

The details of registration sequence is described in chapter 7.1.

The process of registration involves the following actors

- Emergency Organisation (EmOrg): This is an organisation that wants to use CIS technology. It consists of the following actors and components
    - Organisation employee: An employee of the emergency organisation who may be authorised to perform one or all of the following:
        - register his/her EmOrg with CISAdminOrg
        - install CIS-Software (CIS-SW) package,
        - configure sharing and access controls to the services connected to the CIS (i.e. selects which other organisations can access the services and information provided by his/her organisation), and
        - selects which services and information his/her organisation wants to receive,
        - creates and administers Cooperation Groups Online Rooms (CGOR)
- External tool: This is a tool that provides and receives information and comprises the CIS components installed in the local server:

- CIS Connector
- CIS Core
- CIS Distributor
- Security Box. This is responsible for
  - keeping the private-key and public certificate of the EmOrg,
  - Keeping the private-key and public certificate of the tool
  - Keeping the root certificate of the CISAdminOrg
  - generating and managing session group keys used in encryption and decryption,
  - message integrity checking,
  - encryption and decryption
  - selecting the security parameters and algorithms used
- Local CIS-Directory Agent: This is the EmOrg local copy of the CIS-DA. It contains information about CIS members and the services they provide
- Local CIS-Directory Structure: This is the EmOrg local copy of the CIS-DS. It contains information about CGORs and their members.

- CIS Administration Organisation (CISAdminOrg). This is a trusted entity that is provides registration service and acting as a certificate authority. It has the following actors and components
  - CIS Administrators (CISAdmin): An authorised employee, who is responsible for some or all of the following
    - validating the registration application for Emergency Organisations
    - Provide pre-authorisation for tools and devices to interface with the CIS
    - Administer the CIS
  - CIS-DA
  - CIS-DS
- Certificate Authority: This is a component responsible for signing certificates for CIS-registered members and maintaining certificate revocation list

In a PKI scenario, each EmOrg registers with CISAdminOrg to become a CIS-member. During a successful registration process, an EmOrg generates a private and public key pair, and have the CA of the CISAdminOrg producing a certificate by signing the public key. The EmOrg receives a copy of the CISAdminOrg root certificate to use it when validating other EmOrg certificates. i.e. to be able to prove that certificates presented by other EmOrg is indeed signed by the CISAdminOrg.

We assume that there is one-to-one mapping between External Tool to CIS connector, CIS connector to CIS core, and CIS core to CIS distributor. (External tool <-> CIS Connector <-> CIS Core <-> CIS Distributor). An EmOrg may issue a certificate for its external tool

In the event of multiple external tools per EmOrg, each tool is given a private key-public certificate, where the certificate is signed by the EmOrg that owns the tool.

The certificate chain will look like

- Certificate of External tool is signed by
  - EmOrg private key, whose certificate is signed by
    - CISAdminOrg private key

5.3.3. Securing the CIS-components inside the EmOrg

A CIS-SW involves several components communicating with each other inside the boundary of the EmOrg. They are

- External Tool
- CIS-Connector
- CIS-Core
- CIS-Distributor
- CIS-Translation box
- CIS-configuration web-service

Access to these components should be controlled such that only authenticated and authorised components can connect to each other. There are a number of possible solutions to provide authentication and authorisation inside the EmOrg. They are briefly summarised

- Pre-shared keys and ACL

  In this method each components shares a secret key with each component it is authorised to access it. The secret key is used to prove the authenticity of the subject requesting the service. It may also be used in generating a session key between them. Each component will have its own access control list (ACL) that defines authorised services for each subject. This may be a suitable solution for the link between equal peers.

- Trust server

  A trusted server is responsible for managing access control to all components through issuing tickets or session keys. There are many standard protocols that can provide authentication and authorization such as RADIUS, Diameter, Kerberos, TACACS+.

5.3.4. CGOR – Cooperation Group Online Room

Not every information is intended for sharing with all participants in the Common Information Space for several reasons. Personal data are liable to the European and national data protection directives, requiring the “need to know” principle. Confidential and sensitive data have to be restricted to a dedicated list of recipients and must be protected by design.

While the responsibility for the processing, classification and release of data always stays with the user of the owning organisation, the Common Information Space has to provide technical concepts fulfilling the data protection requirements. The basic idea is the segmentation of CIS participants in groups that actually need to cooperate in a mission and to share information. The virtual online room for sharing data between all members of a cooperation group is called CGOR. It is comparable to a mailing list in a secured e-mail system, containing a list of recipients and enables group-specific data encryption.

A global CGOR for sharing public information is available by default, and every tool that is registered for participation in CIS becomes automatically a member of the global CGOR. Specific CGORs can be created by an administrator of any CIS member. Intended participants are invited by the CGOR owner and have to confirm the participation, meaning that any information sent to the CGOR is shared with all other members of the Cooperation Group. Data wrapping and encryption ensures that the information is only accessible by CGOR members.

The tools sending information to the CIS have to classify the sent information in a way that the CIS Connector is able to determine the appropriate CGOR in an unambiguous way. Otherwise, only the global CGOR can be used for sending public information. Nevertheless, receiving information in a CGOR is always possible.

Examples:

Location info of police units must be limited to security forces only. A corresponding CGOR is established with the determining parameter "POLICE". The Connector checks the availability and sends MLP messages only to this CGOR; otherwise the information sharing is denied.

A railway company joins the CIS, providing public traffic information (location and status of trains by EMSI resource messages), and information on incidents (alerts by CAP) that shall only be shared with police and fire brigades. The connector allocates EMSI messages to the Global CGOR but CAP messages only to the corresponding CGOR.

In case of a large incident, several relief organisations have to cooperate for a limited period of time. A CGOR is created for that incident. The Connectors of every organisation are configured accordingly, and every message with the incident ID is posted in the CGOR. After the response phase, the CGOR is closed. There must be a default rule for sharing information related to an incident without corresponding CGOR.

An administration tool will be provided by the CIS supplier. The set-up of CGORs, invitation and confirmation are – in a first approach – done by the administrators of the participating organisations, based on trust and cooperation agreements. When the need for cooperation has ended, every organisation can abandon the CGOR membership, and the owner can close the CGOR.

A more advanced approach might comprise the automated generation of CGORs based on tool functions and defined rules.

Example:

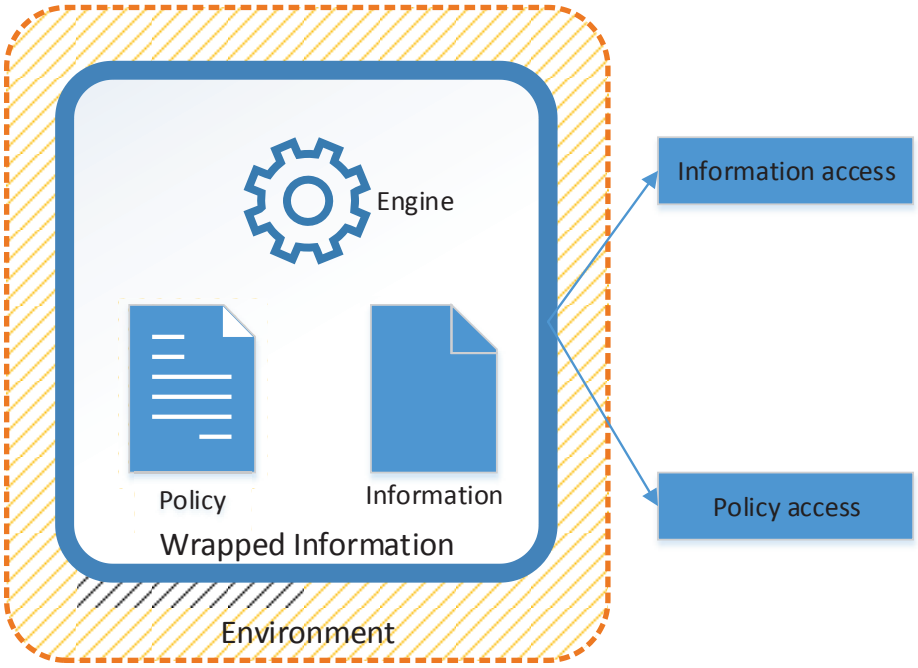
The COP tool of the Civil Protection Authority (CPA) automatically generates a CGOR if an incident is created in the tool, and invites all organisations assigned to the incident. The tools (Connectors) of the addressed organisations automatically accept invitations from CPA. When the CPA closes the incident, the corresponding CGOR is also closed automatically.

This approach requires that the participating tools are prepared for this way of working. At least the leading CPA COP has to support the functions triggering the CGOR management, and the tools of participating organisations must be able to assign automatically their shared information to the generated CGORs.

5.4. Wrapped Information

Wrapped information is defined as a containment of information set and its context-dependent management policy. It consists of an information set that is "wrapped" and a policy that is the wrapping. The policy may include functional capabilities, Figure 14.

Figure 14: Wrapped Information



What wrapped information may offer

It may offer the following features:

- Access control to the information set, which include
  - Read
  - Update
  - Delete
  - Execute
- Protective monitoring
- Access control of the management policy
- Confidentiality and Integrity
- Assure wrapping

The concept provides confidentiality and integrity to the authorised recipients of the information set especially when information is transmitted over unsecure network.

When wrapped information is used

It can be used when sharing information while maintaining a self-contained non-centralised access controls and management settings irrespective of the sharing platforms and/or domain settings.

How wrapped information can be realised

Depending on the required capabilities, W.I. can be realised by a combination of

- Attribute based access control
- Attribute based encryption
- Traditional public key and symmetric key encryption
- Lightweight containers

What wrapped information consists of

W.I. consists of information set and a policy.

Why is it called Wrapped Information

To imply that the information set and the policy are attached together.

What does the policy consist of

The policy consists of rules of access conditions and obligations in a similar way but not limited to the conditions and obligations in XACML standard. The access conditions specify the conditions that should be satisfied to allow access to the information set. This could be in the form of:

Allow Access If subject_Id == police_id

On the other hand, obligations can specify what actions are required to be done. These could be for example, logging an access, or sending notifications when the information set is updated.

What is a dynamic consent/Policy

A dynamic policy is a feature of wrapped information that enables the owner of the information to update the information set, or update the access policy after the information has been wrapped and distributed. For example, the owner of the information set can revoke a previous consent.

How is the wrapped information is integrated in EPISECC

The wrapped information concept is originally designed to provide a stand-alone independent solution of the EPISECC components. In order for the wrapped information to be integrated to EPISECC, the management policy of the wrapped information should allow information access to authorised CIS-Core components.

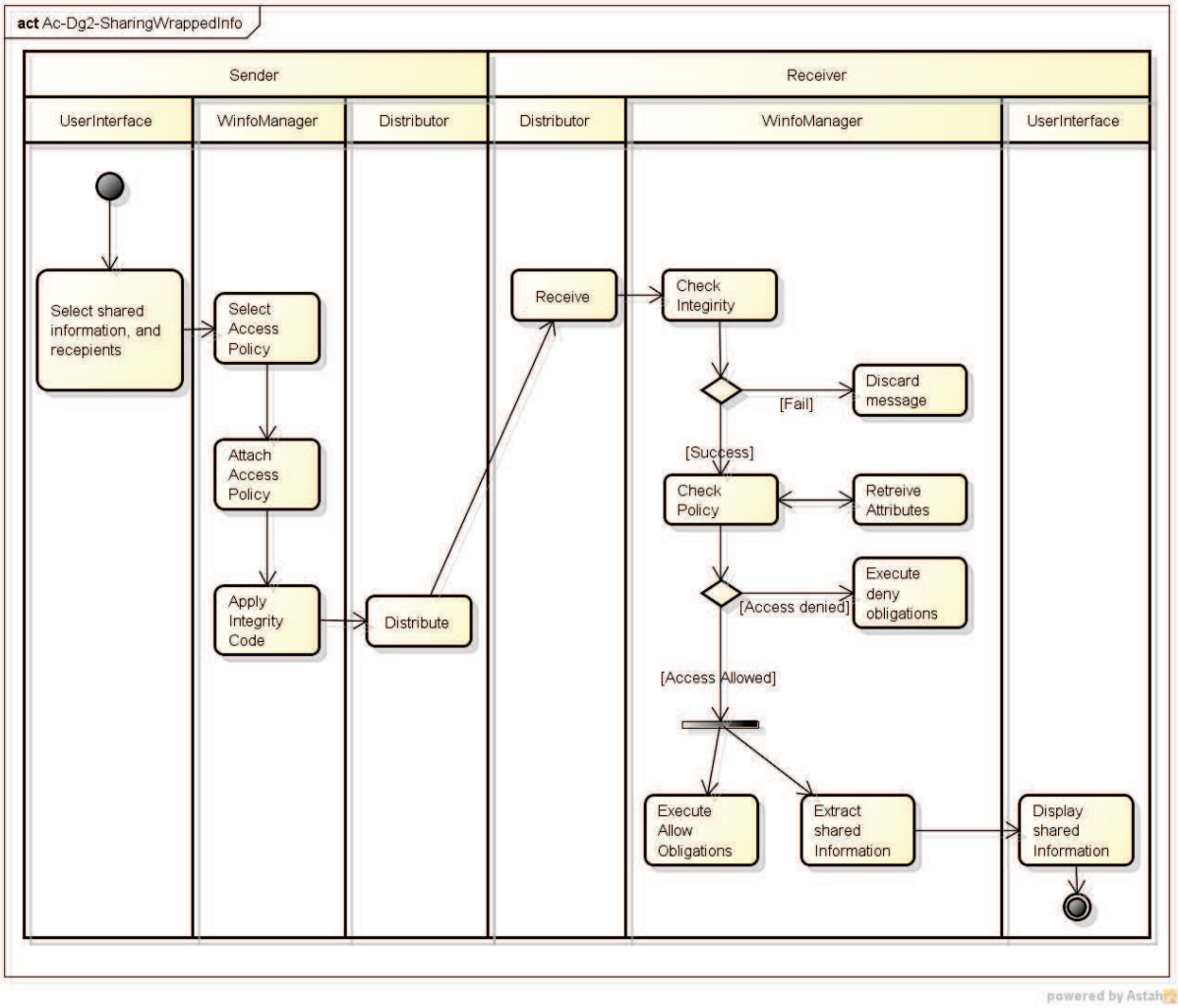


Figure 15: Activity diagram for sharing wrapped information

The figure (Figure 15) is an activity diagram for the use case of sharing information between two users. This is a stand-alone use case that assumes that the WinfoManager- the component that does the wrapping and unwrapping- is integrated with the External tool. Note that for legacy systems the external tool will be oblivious of the wrapping. In that case, the WinfoManager will reside inside the CIS core only. This is clearly shown in the next activity diagram.

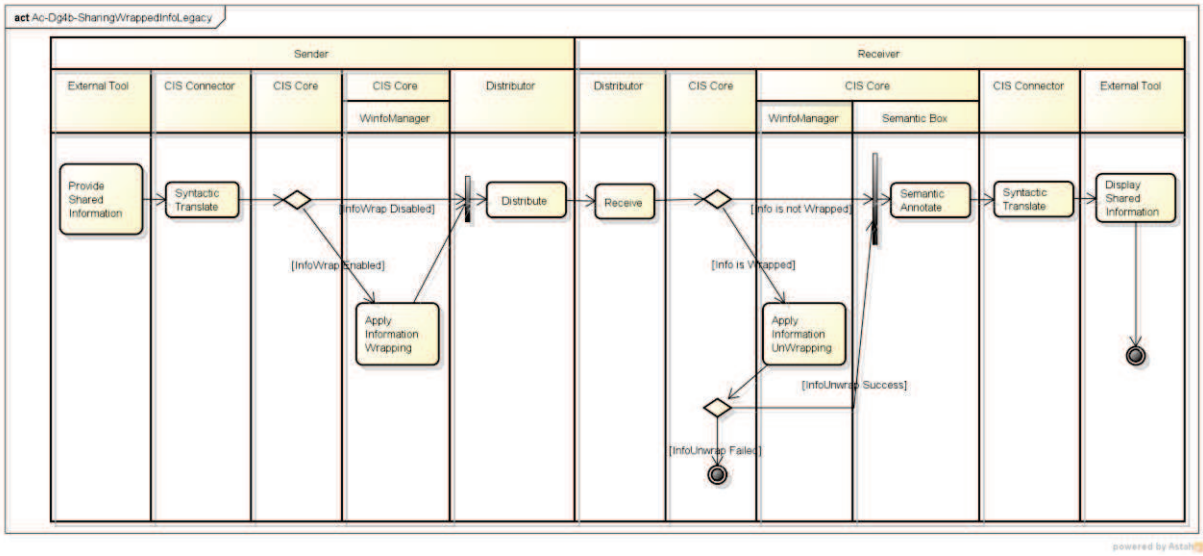


Figure 16: Activity diagram for sharing wrapped information between legacy tools within CIS

The above shows the activity diagram for sharing information between two legacy tools that are using the CIS. Note that activity that is called "Apply Information Wrapping" is divided into three activities "Select Access Policy", "Attach Access Policy", and "Apply Integrity Code" in the previous figure. Similarly, the activity "Apply Information Unwrapping" is decomposed into activities: "Check Integrity", "Check Policy", "Retrieve attributes", "Execute Allow/Deny obligations" and "extract shared information", which can be found previous figure.

5.4.1. Wrapped Information Requirements

The functional and non-functional requirements of the W.I are summarised in the table below. In the following, *subject* is the entity that can perform an action on the system.

Object is the entity that represent resources to which access need to be controlled

Action includes access and can include Create, Read, Update, Delete, and Execute. For the scope of EPISECC, the term is limited to Read.

	Requirement	Type
1.	Secure attachment of the information set to its management policy	Functional
2.	Verify applicability of the management policy to the information set	Functional
3.	Keep the information set and its management policy attached in transit	Functional
4.	Keep the information set and its management policy attached after access	Functional
5.	Keep the information set and its management policy attached at host	Functional
6.	Access to the information set	Functional

7.	Allow only authorised access[2] to the information set as defined by the management policy. Prevent unauthorised access to the information set	Functional
8.	Verify access conditions are met and access obligations are executed (see secure monitoring requirements)	Functional
9.	Verify the ability of the host to execute the management policy	Functional
10.	Allow Break-Glass Access Control feature	Functional
11.	Allow Implicit Deny Access Control feature	Functional
12.	Allow remote access to the information set. (After the wrapped information is sent)	Functional
13.	Gather and verify access subject attributes and access context attributes	Functional
14.	Verify host processes integrity	Functional
15.	Verify subject (process, component or user) authenticity and authorisation credentials	Functional
16.	Access to the management policy	Functional
17.	Allow only authorised access i.e. Create, Read, Update, Delete, and Execute (CRUDE) to the management policy. Deny any unauthorised CRUDE to the management policy.	Functional
18.	Allow remote dynamic management policy features which include revoking a consent by CRUDE an access condition or CRUDE an access obligation	Functional
19.	Enforce parameter checking for secure management policy creation	Functional
20.	Monitoring	Functional
21.	Execute the information access policy obligation: append log service	Functional
22.	Execute the information access policy obligation: send notifications	Functional
23.	Enforce the information access policy obligation: notify the authorised recipient of terms of access	Functional
24.	Enforce the information access policy obligation: request an authorised consent to access information set	Functional
25.	Monitor Unauthorised subject behaviour such as unauthorised log file editing	Functional
26.	Allow authorised remote checking for the log-file	Functional

[2] Information set access: refers to CREATE, READ, UPDATE, DELETE and EXECUTE (CRUDE). For EPISECC we shall confine the term to only READ

27.	Manage and distribute access subject attributes	Functional
28.	Interoperability with legacy processes and systems	Non-functional
29.	Compliance with privacy acts regarding personal information	Non-functional
30.	Mapping policy language to simple understandable human language	Non-functional
31.	Interoperability with distributed design	Non-functional
32.	Modularity of the design	Non-functional
33.	Allowing extending new obligations and conditions	Non-functional
34.	Performance : Allow execution on mobile, tablets and platforms with limited processing power and limited battery power	Non-functional
35.	Ensure confidentiality and integrity of shared information	Non-functional
36.	Ensure integrity of the access policy	Non-functional
37.	Protect against conditions and obligations abuse	Non-functional

Table 3: Functional and Non-functional requirements of the W.I.

5.5. Value added services

(The hereafter described concepts of value added services will not be implemented in the EPISECC prototype).

The Distributor stores all sent and received messages in a local data base. This message pool can be used for various statistics and reports. The Core offers a plug-in interface that enables access to this data. The plug-ins themselves have to be developed and installed by the tool owner based on templates provided with the installation package, similar to the Connector.

The Plug-in is triggered by a Rest service interface and uses a defined set of query features exposed by the Distributor. The queries can use attributes of the EDXL DE envelope but not the message content. The fetched messages have to pass the Security Box and the Semantic Box for decryption and semantic annotations before they are handed-over to the plug-in.

The value added services are specific to the tool owner and have access only to the information which is available for the particular tool. There is no central repository or service that accesses all messages communicated in CIS.

Examples for value added services:

Legal recording: all trafficked messages are listed by sender, receiver CGOR, time stamp, and additional descriptive information if available in the message envelope. The details and content of the messages can be retrieved by the query plug-in.

Message traffic statistics: statistics of outbound/inbound messages per type and time slot.

Query plug-in: messages can be retrieved by querying the attributes of EDXL DE, e.g. message ID, time stamps, sender, receiver etc. The message content is decrypted and semantically annotated in the same way as the original message was handed-over between Connector and Core. Filtering of the message content is possible but not supported by the plug-in template. It has to be implemented individually by the tool owner.

5.6. Rollout concept and system administration

This chapter describes aspects and considerations needed for new participants of the Common Information Space. We assume that a new organisation Org_{new} wants to join the EPISECC Network and wants to install its own adapter to connect its tool T_{new} to CIS. This chapter describes the needed steps to get the things to start.

5.6.1. Assumption for the considerations

The CIS networks is built up as symmetric architecture (peer to peer) with no central point of function, as a repository or a server, which is needed during normal operation.

Two Exceptions are allowed for central service:

- A central repository for booting the global semantics, which afterwards will be distributed to the other nodes
- A service for registering the own node by getting in contact with a central authentication authority.

5.6.2. Migration and packaging of CIS-Adapter

The software of CIS adapters can run on any hardware platform. Prerequisite environment are a Java Development and Runtime environment.

The CIS adapter will be provided as binary packages from the EPPISEC server and consists of the sub packages CIS Core, CIS Distributor, CIS Connector.

CIS Core and CIS Distributor will be provided as binary libraries.

For the CIS Connector only templates and sample implementations will be provided. Only an EPISECC-(Null) Connector – implementing the EPISECC protocol – will be provided.

Each participating partner has to program his migration of interface to the own tools, based on the EPISECC Connector template.

This “own libraries”, implementing the CIS Connector interfaces, has to be added to the provided CIS application bundle.

Together the CIS package should be deployed to runtime environment of the CIS adapter.

Following bundling is feasible:

- CIS-Adapter is installed in an extra server
- CIS-Adapter is an addition to the tool of the emergency organisation (deep integration)

- CIS-Connector is an addition to the tool of the emergency organisation, CIS-Core + CIS-Distributor are installed in an extra server
- CIS-Connector + CIS-Core are an addition to the tool of the emergency organisation, CIS-Distributor is installed in an extra server.

“Extra server” means a separate installation on any server of the organisation inside the secured intranet, or a virtual machine or even dedicated hardware like a router. It is addressed by web services only.

5.6.3. Management console

One additional package is needed for configuration of the CIS Adapter, which is called "Management Console".

The management console is only needed at system start up and as support interface, if some features are not supported by the Emergency tools (like creating CGORs).

The management console offers an HTML protocol at a well-defined port number.

Following functions are supported by the management console:

- Set Master password
- Editing additional users and their roles
- Start Registration process
- Administration of CGORs (START/JOIN/EDIT/EXIT)
- Configuring Translation Box
- Diagnostic and test tools

5.6.4. Start the registration process

Communication can only be established between CIS Adaptors, if the systems trust each other (Authentication).

The CIS Adaptor must have a public accessible port to the internet and a known unique name (default: Internet name) to build a connection from outside.

One CIS Adapter has a well-known address (Boot Node), which is preconfigured in the CIS Distributor. This well-known address is only needed in the boot phase. (Alternatively one known partner internet name can be configured at startup).

5.6.5. Authentication of the CIS adaptor

A new partner has to fill a registration form and submit it to authorisation authority, which has to approve the correctness of this data and give the enabling of the trust. This clearance is logged by the system.

If trust is enabled, this new added node is fully entitled member of the network.

The data in the Registration form describes the meta-information for this Connector instance.

Which data are needed?

- Name of the responsible Person
- Purpose of usage
- Name of the node
- Login name and credential for the primary local administrator (admin)
- Defining other users and their roles.
- Creating or assigning the **Organisation** structure, to which this node belongs to. The creator is the owner and can change the content.

The Organisation metadata are the same as in the questionnaire.

If all mandatory fields are inserted the data record will be propagated.

If trust is enabled, this new added node is fully entitled member of the network. (EPISECC-participant)

A Public/private key pair is generated for this new CIS-Adapter.

5.6.6. Local administration of users and roles

All administration features of the management console can be bound to needed roles of authenticated users or you can configure the system to be public accessible to anyone.

The following concepts should be implemented at least:

- Defining users with their passwords
- Defining Groups and associating users to groups
- Reserved Groups are PUBLIC, LOCADMIN and GLOBADMIN
- Assigning Access rights to “Objects or Methods” to Groups or users
- Access rights are: READ, WRITE, EXECUTE

The Users, Roles and Access rights can be managed by the users having the LOCADMIN roles assigned.

5.6.7. Browsing all Organisations, Adapters and Missions/CGORs

If a user has the right to read the meta information for all connected systems everyone can get a list of all organisations and there meta data.

If a user has the right to read the meta information for all missions these information can also be accesses at a summary level.

5.6.8. Management of missions/CGORs

Everyone having the right to start a new mission (or CGOR) can create a new mission belonging to the local organisation. He/she has to describe meta information for the mission and invite one or more other “organisation” to participate.

A special CGOR (called "PUBLIC") always exists, which includes all participating members of CIS network.

Following operations are supported:

- START of a new CGOR and invite participants
- JOIN to a CGOR to which I was invited
- EDIT configuration of a CGOR created by me
- EXIT of a CGOR started or joint by me
- CLOSE a CGOR started by me

5.6.9. Global administration

Global administration can influence the behaviour the whole system, also other CIS adaptors.

The role “GLOBADMIN” can only be assigned by another user having “GLOBADMIN” right.

Possible actions for these users are:

- Changing global system parameters
- Blacklisting another CIS Adapter, if affected.

5.6.10. System Backup and Restore

As all the information is replicated to all other systems, all systems contain roughly the same information. A failed system can be restored by the implemented synchronisation mechanism. So no special backup and restore strategy is needed.

Only information, which is kept only locally, should be saved as backup. At this time this is only the case for the “private key” for encryption.

5.6.11. System Updates

System Update can easily be implemented by downloading a new release of CIS software and by combining this with the updates of the own Connector modules. The new built application can be deployed again.

Local Data stores for Directory Agent and Segmentation are organised separately from the application and are updated dynamically during operation.

For each release of the Semantic Repository an update script for the local data store has to be provided.

5.6.12. Logging

There exist two ways of logging:

- Local logging of the software modules for tracing the system (like Log4J). This information is not replicated.
- Global logging, which traces important events in the network. Examples are:
    - Authentication event
    - Starting of a mission
    - Ending of a mission

6. Legal and ethical aspects

6.1. General legal considerations

In order to ensure that the general architecture of the CIS is in compliance with the applicable data protection rules and enjoys trust of all parties involved, the CIS should guarantee an adequate level of security that is appropriate to the risks associated with its data processing activities.

The main objective of the CIS is to provide an open-ended exchange tool enabling the interaction and coordination of civil protection assistance in case of a cross-border disaster. Whilst for the purposes of the proof of concept (D6.1) it has been agreed to narrow down the number of connected tools to a defined set of data sharing mechanisms, the general architecture of the CIS does not limit the types and the amount of data which will be exchanged. As a consequence it is highly likely that among all various types of processed data also personal data (e.g. MDG includes location data) will be exchanged throughout the CIS platform. The processing of personal data triggers the applicability of European data protection legislation. Therefore the general architecture should adhere to the requirements set forth by the applicable data protection legislation and more in particular to its security standards.

While the currently applicable regulatory framework is defined by the EU Data Protection Directive (Directive 95/46/EC), it is recommendable to anticipate the more stringent security requirements of the future General Data Protection Regulation (GDPR) which will enter into force on the 25th of May 2018. By doing so the CIS's architecture will remain a state-of-the-art privacy complying mechanism when the GDPR becomes enforceable. Article 32 of the GDPR imposes a general obligation, for controllers and processors alike, to implement appropriate technical and organisational measures in order to ensure an adequate level of security for the data-processing. In addition to the legal framework that imposes certain security requirements, data security and integrity are also indispensable to build trust in the reliability and confidentiality of CIS. Sufficient confidence in the security properties of the CIS is of key importance in order to convince organisations to share their information through the CIS.

6.2. Requirements according data protection regulation

The requirements below stem from a combined reading of the current EU Data Protection Directive and the future General Data Protection Regulation. The table indicates for each of the identified requirements to which extent the CIS is affected and how the requirement is addressed in the CIS architecture.

No.	Requirement description	Measure type
1.	Implement appropriate organisational measures to ensure a level of security appropriate to the risk *Is in the responsibility of the owner (provider) of a CIS. D5.3 proposes appropriate procedures and measures in order to make a CIS instance safe.*	Policy, procedure
2.	Implement appropriate technical measures to ensure a level of security appropriate to the risk. *Is essential part of the architectural design in this document D5.2 (security box, trust handling and registration) and will be implemented in an exemplary way in the CIS prototype D6.2.*	Procedure, practice
3.	Implement appropriate organisational measures for pseudonymisation and encryption of personal data. *All participating organisations and their tools need to be registered and trusted by the owner of CIS (e.g. LEMA).* *The CIS itself doesn't care about the transmitted data content – this has to be covered by the connected tools and the tool users.*	Policy, procedure
4.	Implement appropriate technical measures for pseudonymisation and encryption of personal data. *All data sent to the CIS are encrypted in order to limit messages to the intended recipients. Specific handling of personal data (e.g. pseudonymisation) is outside the scope of CIS, and stays in responsibility of the connected tools.*	Procedure, practice
5.	Implement appropriate organisational measures ensuring the ability to ensure the ongoing confidentiality, integrity, availability and resilience of processing systems and service *The ownership of information shared in CIS always stays with the producer (sender) of the information. Messages sent to CIS are logged and stored locally, on the participants' servers and can be analysed by the owning organisations.*	Policy, procedure

6.	Implement appropriate technical measures ensuring the ability to ensure the ongoing confidentiality, integrity, availability and resilience of processing systems and service *The concept of secured CGOR (Cooperation Group Online Room) and the related encryption of all messages is designed for ensuring confidentiality and integrity.* *Availability and resilience are considered by the distributed CIS structure, hosted at the participants' servers, and the peer-to-peer message distribution and synchronisation design. This architecture allows the CIS to continue working even if the connectivity is partly down, and to resume the full information after re-connection.*	Procedure, practice
7.	Implement appropriate organisational measures ensuring the ability to restore the availability and access to personal data in a timely manner in the event of a physical or technical incident *No specific handling of personal data in CIS itself – it is the responsibility of the participating organisations (see also RQ 3, 5)*	Policy, procedure
8.	Implement appropriate technical measures ensuring the ability to restore the availability and access to personal data in a timely manner in the event of a physical or technical incident *No specific handling of personal data in CIS itself – it is the responsibility of the connected tools (see also RQ 4, 6)*	Procedure, practice
9.	Implement appropriate organisational measures ensuring a process for regularly testing, assessing and evaluating the effectiveness of technical and organisational measures for ensuring the security of the processing *Not applicable for the EPISECC research project (we are not a "real" CIS provider). The experience gained in the proof of concept exercise will be analysed accordingly in the lessons-learned reports D6.3 and D5.4.*	Policy, procedure
10.	Implement appropriate technical measures ensuring a process for regularly testing, assessing and evaluating the effectiveness of technical and organisational measures for ensuring the security of the processing *The data traffic in CIS is completely logged in the participants' servers. These log-files will be analysed after the POC exercise. Specific security tests might be included in the POC (t.b.d.).*	Procedure, practice

11.	Take steps to ensure that any natural person who has access to personal data does not process them except on instructions from the controller, unless he or she is required to do so by Union or Member State law. *No specific handling of personal data in CIS itself – it is the responsibility of the participating organisations.* *The CIS is designed for the communication of systems (servers) rather than persons. Personal credentials of users have to be checked by the connected tools. If a tool is a trusted partner, the users of this tool are considered as trusted, too.*	Policy, procedure, practice

Table 4: Legal requirements

6.3. Legal and ethical impacts on CIS

The common information space as designed in this document is neither a data store or inventory nor an information system processing the data. It is rather a middleware establishing the connection between IT systems owning and processing the information. It can be compared with a telephone system connecting the dialogue partners but agnostic about the dialogue content.

While the ethically and legally compliant handling of sensitive information stays in the responsibility of the CIS participants (trusted organisations and integrated tools), the CIS software has to guarantee the secure transmission of information. That comprises the protection of all transmitted data against unauthorised access and tampering, the monitoring of due delivery of information, and mechanisms for recovering interrupted connections and lost data packets.

In the CIS architecture, all components of a tool adaptor are located within the secured network (intranet) of the tool owner. That means that all internal service calls between the adaptor components are considered as safe. Before the transmission of a message via internet, the message content is encrypted, specific to the intended recipients (see 5.3).

In addition, the CIS security procedures provide features that allow the configuration of closed cooperation groups (CGOR) whose members are given privileged access to restricted information. The secure handling of trust, certificates and CGORs is further elaborated on two levels, technical (this D5.2) and procedural (D5.3).

7. CIS Security Procedures

In this section, we select 7 important use cases and display in most cases sequence diagrams for them. Sequence diagrams are important tools to show how system components and actors interact and behave.

7.1. Initialisation and Registration of CIS participants

a. A contractual agreement is signed between legal representatives of the organisation exchanging information. This should cover all legal aspects of information sharing. Two basic scenarios are possible
    i. An Emergency Organisation representative signs agreement with other emergency organisation, she wants to exchange information with
    ii. An Emergency Organisation representative signs agreement with a central authority that will form the legal basis for information exchange and usage.

Details of this step (1.a) *are out-of-scope of this document. However, the legal basis for information exchange is essential to be set before all the following technical sequences can start.*

The CIS architecture should support information sharing between organisations in absence of a CIS Administration Organisation, as well as, information sharing in the presence of a CIS-Administration Organisation.

A CIS-Administration Organisation can act as a certificate authority CA for signing and issuing certificates. A Public Key Infrastructure would provide identification and authentication, as well as non-repudiation.

If no PKI is available, an alternative authentication method can be used. This is a viable solution for peer-to-peer communication and small groups communication. However, this option may not be scalable enough when many organisation are involved

b. An Emergency Organisation Authorised Employee (EmOrgAdmin) (Most likely an Administrator) obtains the official CIS Software package in a secure way.
c. EmOrgAdmin starts the installation process of the CIS-SW package
d. An authorised EmOrg employee completes a registration request form to register her organisation with the CIS-AdminOrg. The application form includes relevant and important information about the applicant organisation
e. An authorised employee of EmOrgAdmin validates the registration request and if satisfied approves the application
f. An automated registration process starts where a client side (EmOrg) generates a public and secret keys pair
g. The EmOrg client sends the public key part of the pair to the CIS-AdminOrg server.
h. The CIS-AdminOrg automated component generates a public key certificate where it binds the EmOrg identifying information to the public key.

i. The CIS-AdminOrg automated component updates the certificate directory (CIS Directory Agent CIS-DA), where information about the new registered EmOrg and lists her public key certificates
j. The CIS-AdminOrg sends securely its root public certificate to the EmOrg
k. The CIS-AdminOrg sends the updated CIS-DA to the EmOrg and all distributors
l. The CIS-AdminOrg sends the updated CIS-DS to the EmOrg and all distributors

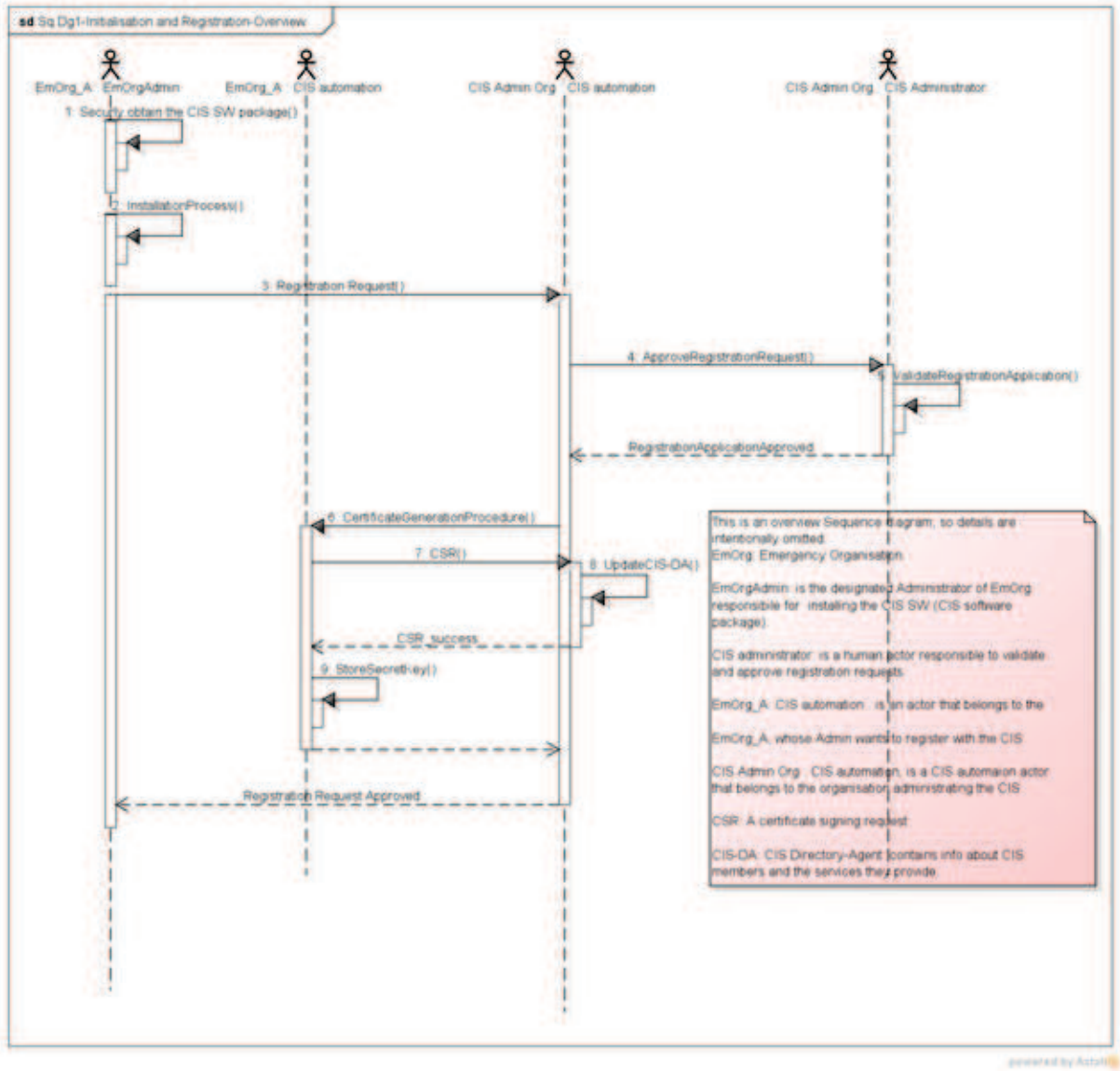


Figure 17: Sequence Diagram 1 – Initialisation and Registration

The trust model is implicit in the above sequence diagram. For the Emergency Organisation, the root of the trust starts with obtaining an official CIS-SW package. This may include the root public key certificate of the CIS-Administration, which will enable authentication of the CIS-Administration Organisation. The CIS-SW package is hashed and signed to provide assurance of the integrity and behaviour of the SW package. For the CIS-Administration Organisation, the trust in the identity and the authenticity of the registering emergency organisation is rooted on the judgement of the

authorised employee of the EmOrgAdmin, who will need clear guidelines on validating and registration forms.

In case of absence of an EmOrgAdmin, we have a (a peer-to-peer) model where authorised employees of two emergency organisations need to mutually identify, and authenticate each other.

7.2. Cooperation Group Online Room (CGOR) administration

7.2.1. Search for Active CGOR

a. An authorised employee of an emergency organisation logs on securely through to a CIS-web-service
b. After successful login the web service is granted access to the local CIS-Directory Structure and local CIS-Directory Agent. The Local CIS-DA contains information about CIS members and the services they provide. The local CIS-DS contains information about CGORs and their members.

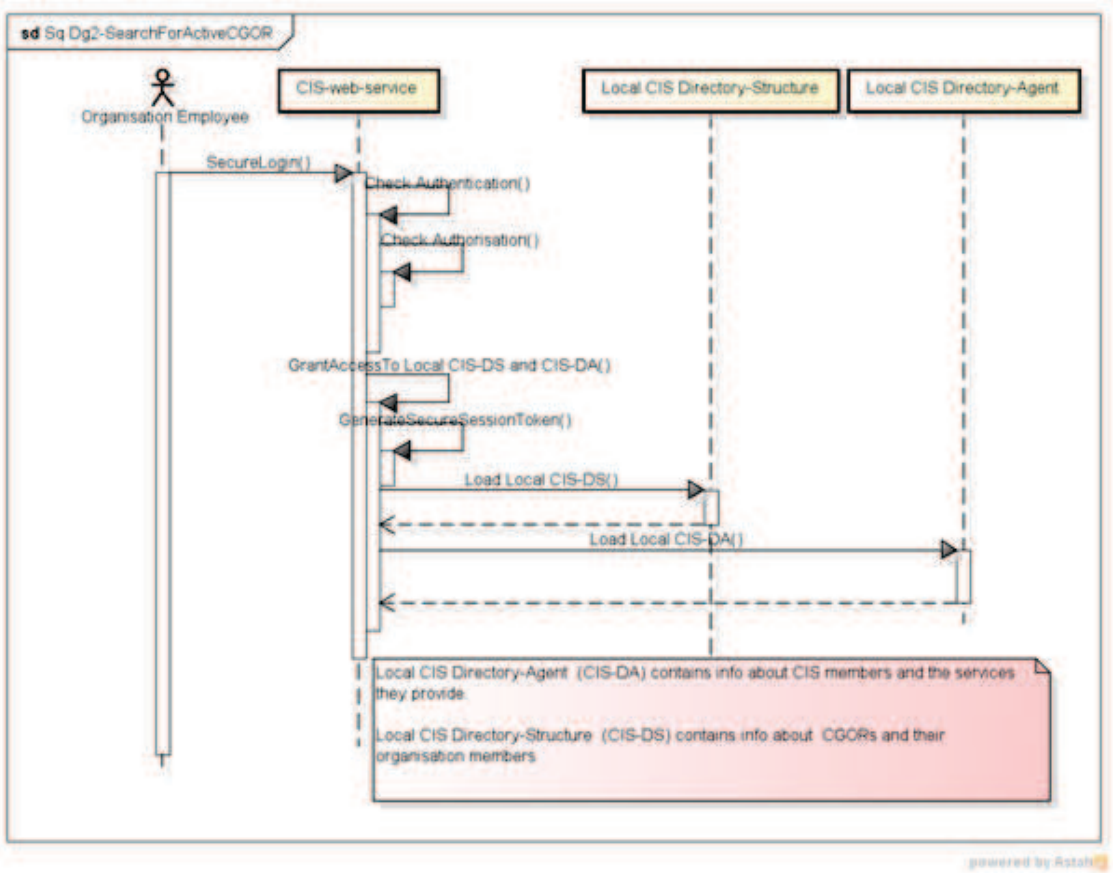


Figure 18: Sequence Diagram 2 – Search for Active CGOR

Local CIS-DA and local CIS-DS are constantly updated and synchronised with other CIS-DA and CIS-DS. This guarantees that the employee is presented with an up-to-date information.

The CIS-web-service is an administrative tool.

The above sequence is independent of the presence of a CIS-Administration-Organisation

7.2.2. Starting a CGOR

a. The authorised employee creates a CGOR, through an option given through the Admin GUI provided by the CIS-Web-service.
b. "Create a CGOR Request" is completed through filling a form containing relevant information about the CGOR, such as description of the CGOR purpose, invited members, provided services if any, security settings, etc.
c. The CGOR is created successfully and the local CIS-DS is updated to include the newly created CGOR.
d. The local CIS-DS updates other CIS-DS

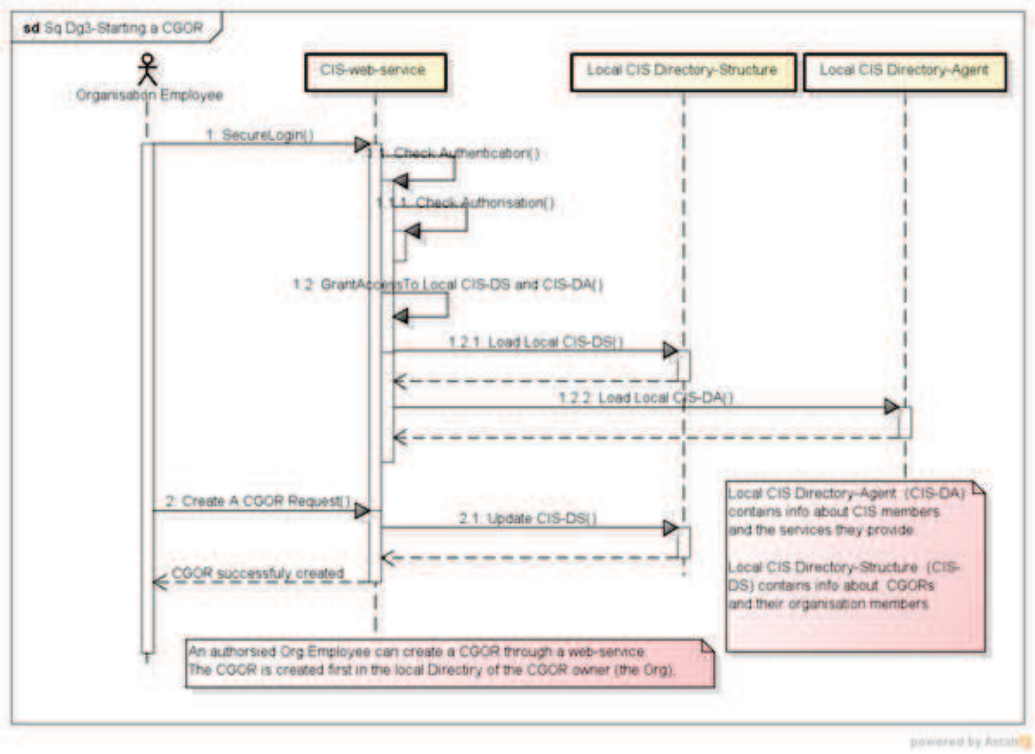


Figure 19: Sequence Diagram 3 – Starting a CGOR

In the above scenario, we assume no need for further approval from CIS-Administration-Organisation. It may be possible to include an option of Official Endorsement of the CGOR by the CIS-Admin-Org. The benefit of endorsement is that it will prevent abuse done by creating too many CGORs, and provides filtering between endorsed and non-endorsed CGORs. However, since only authorised employees of registered emergency organisations are entitled to create CGORs, the risk of abuse is very negligible. Hence, official CGOR endorsement will not be considered any further.

7.2.3. Inviting CIS-Members to the CGOR

a. The authorised employee selects CIS-members to invite to join his CGOR.

The employee should be authorised to invite members to the CGOR. The authorisation is automatically granted to the creator of the CGOR. It may also be delegated to selected CGOR-members. For the time being, we will assume that the authorised employee in the above step is the creator of the CGOR

b. An invitation to join the CGOR is forwarded to the invited organisation. This can be done through several scenarios. The following is one way of achieving this

   i. The invitation to join a CGOR is created on the local CIS-DA

   ii. The local CIS-DA of the inviting organisation updates the CIS-DA of the invited organisation

   iii. An authorised employee of the invited organisation will see the notification when she logs in to her local CIS-DA

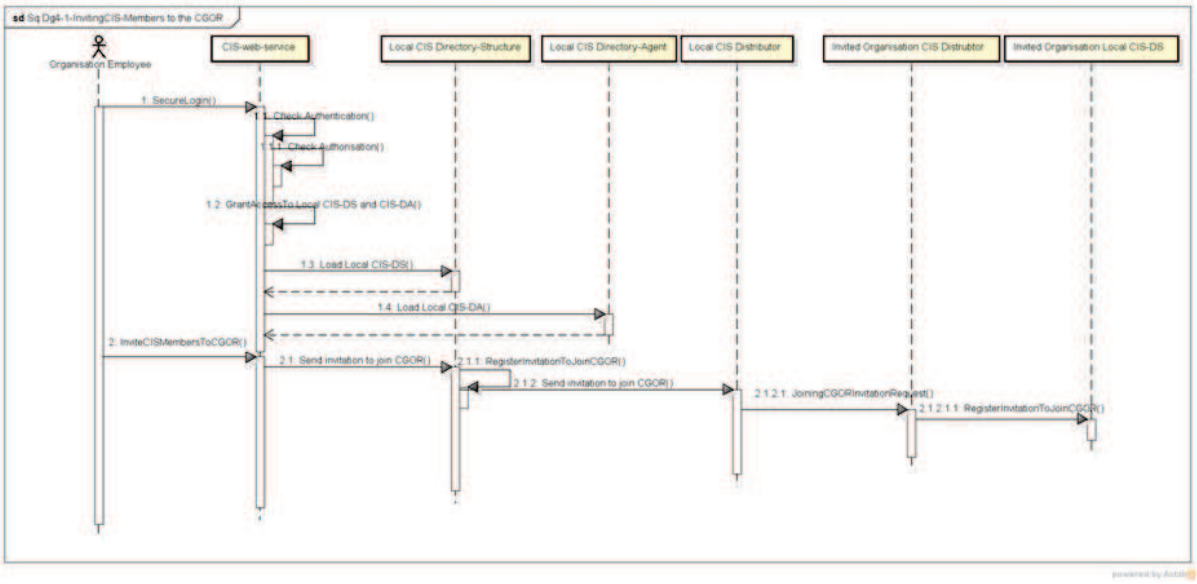


Figure 20: Sequence Diagram 4 – Inviting CIS-Members to the CGOR

This approach may cause delay of a response from the invited organisation, if the employee is not logged in. Alternative solutions may include sending a notification email. However, going any further is beyond scope

c. An authorised employee of the invited organisation views the request to join the CGOR and accepts the invitation

It is possible to avoid going through this step altogether. In that case, emergency organisations can be automatically added to any CGOR if the CGOR creator invites them. i.e. invitations are automatically accepted. This will still be viable option given that the environment is of secure, trained, authenticated, authorised and accountable employees of emergency organisations. The disadvantage of an automatic invitation acceptance may be that the system may be vulnerable to a flood attack by being member of several CGORs. However as pointed out, since invitations should be signed by CIS-members, this threat is very unlikely. No need for further discussion

d. The Invitation acceptance is saved on the local CIS-DA of the invited organisation and acknowledgment of acceptance is sent back to the inviting organisation CIS-DA.

e. After receiving invitation acceptance, the inviting Emergency-Organisation update the CGOR member list and exchanges the CGOR security parameters with the new CGOR member. The security parameters will include security keys, algorithms and -if needed- signatures requirements.

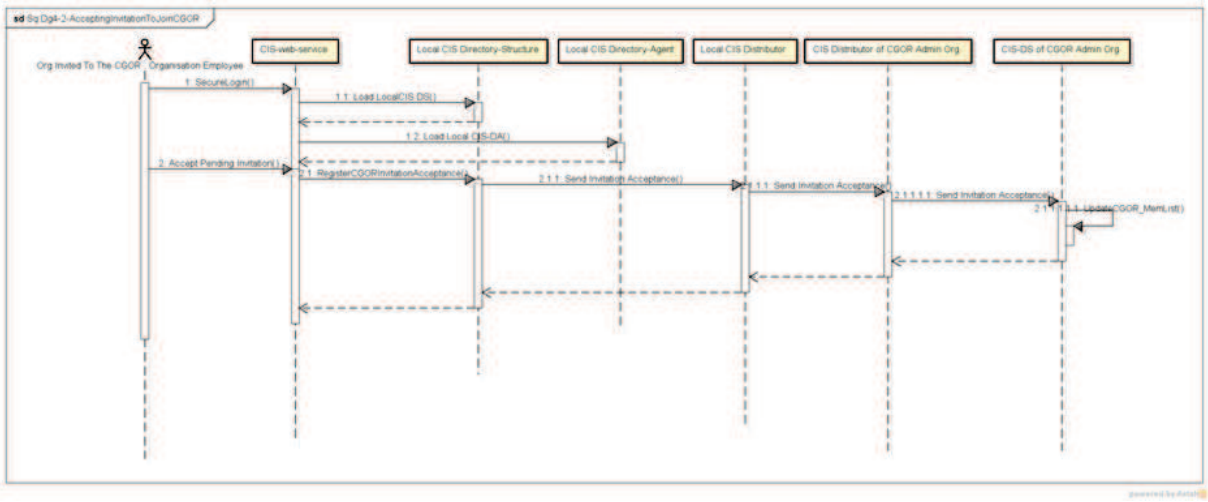


Figure 21: Sequence Diagram 5 – Accepting an Invitation to Join a CGOR

CGOR member list contains information on the CGOR members' identity, services and distributors' addresses of the CGOR members. It can be edited only by the Emergency Organisation created the CGOR or delegated members. CGOR member list is distributed via the distributor every time it is edited. It is hashed and signed by the CGOR creator to ensure integrity and authentication.

The current design assumes that symmetric key encryption is used in encrypting data shared on a CGOR, as session keys are used. In this case, the key management of the CGOR will be carried by the CGOR-creating Organisation, which will be responsible for creating session keys, and distributing them on CGOR members. This solution is valid given that we assume a PKI.

We shall describe other key management options in the absence of PKI and certificate authority.

7.2.4. Request to become a CGOR member

In the previous sequence we described a scenario where the CGOR creator invites other CIS-members to become members of the CGOR. In this scenario, we describe a situation where a CIS-member wants to join a CGOR.

a. The authorised employee of the emergency organisation makes a request to become member. This can possibly be done by either
    i. An option given through the CIS-web-server.
    ii. By contacting the authorised person from the CGOR creating organisation and request to be added to the CGOR. In this case the process described in the previous scenario (Inviting CIS-Members to the CGOR) is triggered

The sequence in the diagram assumes the first option (i) and is described in the following steps.

b. The CGOR- admin organisation receives the request to join the CGOR.
c. The authenticity of the request is checked
d. If the request is accepted, the new organisation is added to the CGOR-member list and the updated CGOR-member list is signed and saved in the local –DA
e. The new CGOR member is notified of the result of the request.

f. The security parameters of the CGOR is sent to the new member so that it can decrypt the shared data on the CGOR
g. The updated CGOR member list is distributed to the CGOR members

7.3. Sharing Information

7.3.1. Sending Information

a. An external tool sends the information to be shared to the CIS-Connector
b. The Connector receives the information and perform a syntactic translation (possibly annotation) if there is a need to
c. The Connector sends the annotated syntactically-translated information to the CIS-Core
d. The CIS-Core sends the information to the translation box (Semantic adapter in the figure) for semantic translation and annotation
e. The Translation box performs semantic translation and sends back the annotated semantically-translated information to the CIS-Core
f. The CIS-Core sends the information to the Security box (Information wrapper) , with the security parameters.
g. The Security-box encrypts the information using the appropriate keys and sent the ciphertext back to the CIS-Core
h. The CIS core sends the ciphertext to the Data-distributor along with information about the authorised recipients
i. The data distributor sends the information to authorised recipients

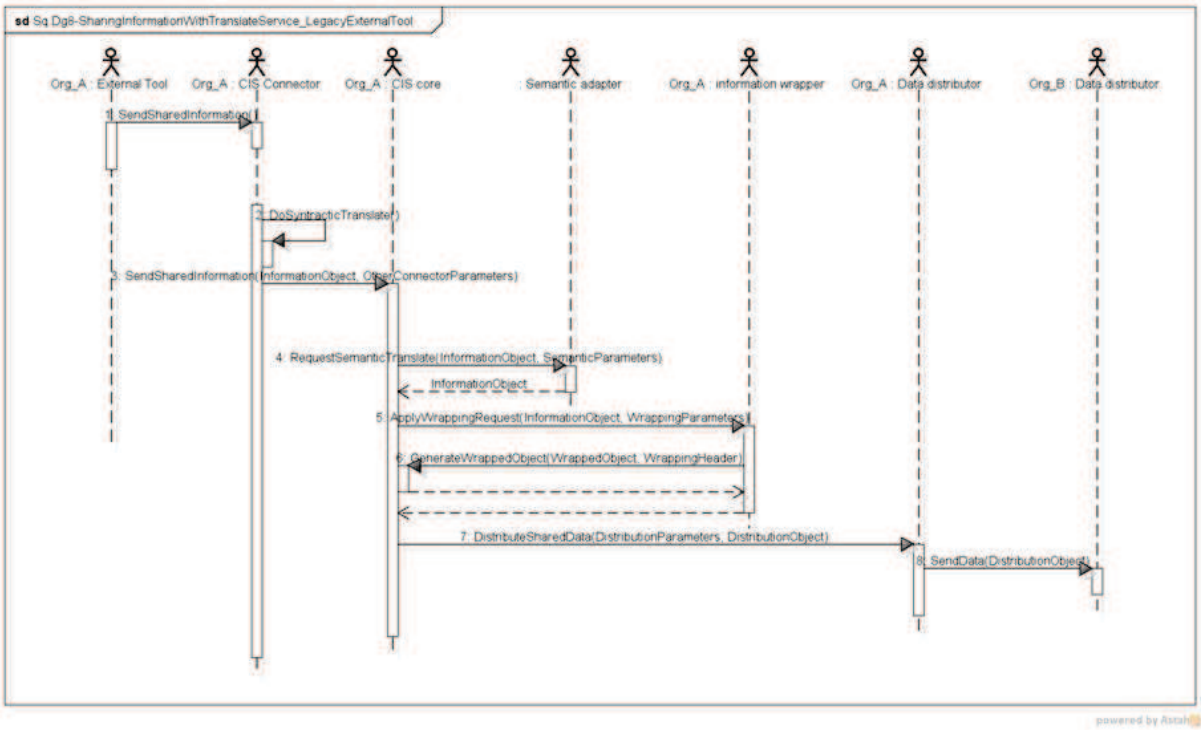


Figure 22: Sequence Diagram 6 – Sharing Information-Sending Information

In this sequence, it is assumed that the Semantic translation occurs at the Sender's. This is one of the options. Possible options are; translation at the sender and at the receiver, translation at the receiver only, and translation at the sender only. It has been agreed to focus on developing and implementing the option of Translation at the receiver only. Main reasons are to reduce the complexity and potential of errors resulting from double translation, while placing the responsibility of translation on the receiver and at his own discretion.

The Security-box provides confidentiality and possibly integrity check to the shared information. The security box need to apply the suitable encryption keys and algorithms that are used for the CGOR.

7.3.2. Receiving Information

a. The Data-distributor receives information coming from other distributor
b. The Data-distributor sends the received information to the CIS-core
c. The CIS-Core checks the authenticity of the information and sends it to the security-box for decryption
d. The Security-box receives the ciphertext, along with headers explaining the context (such as CGOR-id, security parameters used)
e. The security-box checks the access conditions and if satisfied decrypts the received information and returns a plaintext to the CIS-Core.

The access conditions are rules set by the information sender/originator/owner that must be met to grant access to the information. For example, updating a log file that monitors when the information is accessed, or notifying a third party, or the sender before accessing the information, etc.

f. The CIS-Core sends the plaintext to the Translation-box for sematic translation

g. The Translation box performs semantic translation and returns the annotated semantically-translated information to the CIS-Core
h. The CIS-Core sends the information to the Connector
i. The Connector performs a syntactic translation (possibly annotation) if there is a need to. The Connector sends the syntactically translated information to the tool
j. The tool uses/processes/displays the message to the human-operator

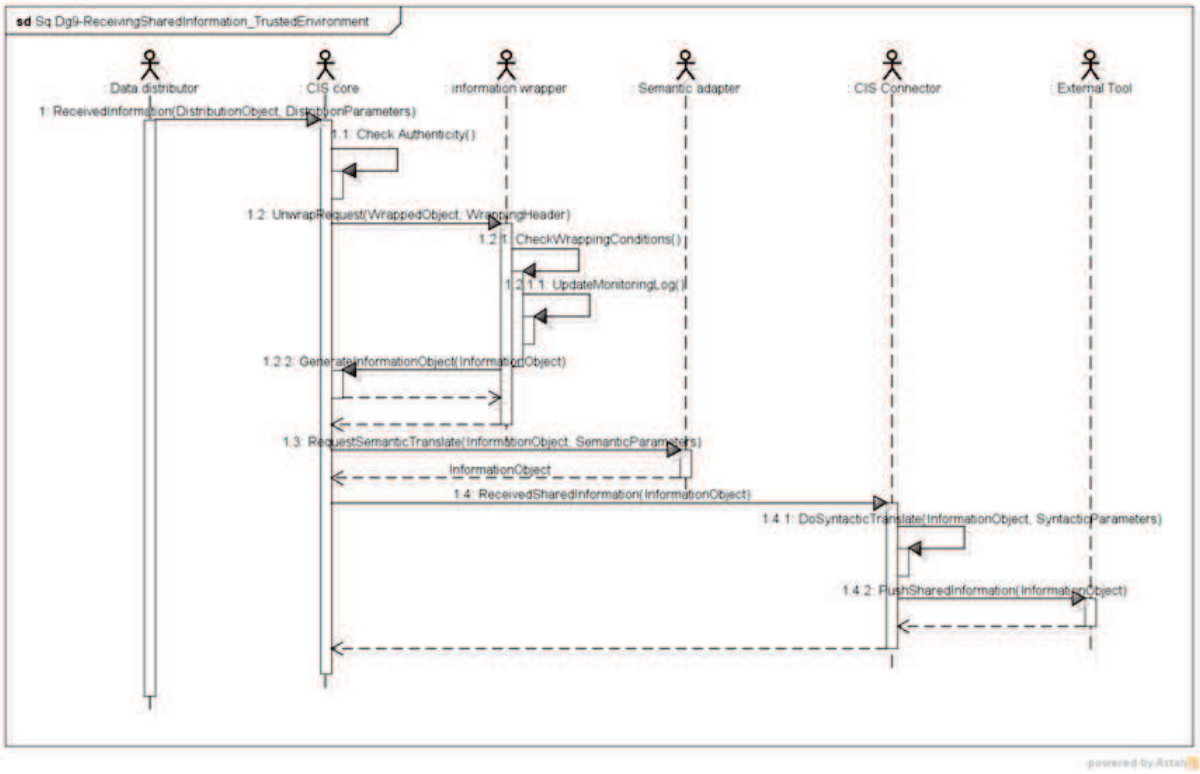


Figure 23: Sequence Diagram 7 – Receiving Shared Information

Bibliography

[1] EPISECC deliverable D2.1 – PPDR Information Space – Status quo of commercial, research and governmental projects and applications, 2015

[2] EPISECC deliverable D3.4 – Pan-European inventory of disasters, 2015

[3] EPISECC deliverable D4.2 – Taxonomy model, 2016

[4] EPISECC deliverable D5.1 – Protocol & Network Interoperability

[5] FP7 project DRIVER, Experiments Strengthened Responders, http://driver-project.eu/experiments_SR [accessed 8 Sept. 2016]

[6] OASIS open standards consortium https://www.oasis-open.org/ [accessed 8 Sept. 2016]

[7] OASIS emergency management technical committee https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=emergency [accessed 8 Sept. 2016]

[8] OASIS Emergency Management TC, EDXL DE V 2.0 standard draft, 2012 http://docs.oasis-open.org/emergency/edxl-de/v2.0/csprd02/edxl-de-v2.0-csprd02.odt

[9] OASIS standard, EDXL DE V 1.0, 2006 http://docs.oasis-open.org/emergency/edxl-de/v1.0/EDXL-DE_Spec_v1.0.pdf [accessed 8 Sept. 2016]

[10] OASIS Standard, Common Alerting Protocol Version 1.2, 2010 http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.doc [accessed 8 Sept. 2016]

[11] CEN, "CEN Workshop agreement CWA 15931, Disaster and emergency management-Shared Situation Awareness", 2009.

[12] ISO, TR 22351 Technical Report: Societal security - Emergency management - Message structure for exchange of information http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=57384 [accessed 8 Sept. 2016]

[13] OGC (Open Geospatial Consortium) http://www.opengeospatial.org [accessed 8 Sept. 2016]

[14] Open Source Geospatial Foundation, GeoServer software server
http://geoserver.org/
[accessed 8 Sept. 2016]

[15] Ka-Ping Yee, PFIF 1.4 Specification, 2012,
http://zesty.ca/pfif/1.4
[accessed 8 Sept. 2016]

[16] Protégé free, open-source platform
http://protege.stanford.edu/products.php
[accessed 8 Sept. 2016]

[17] Apache Jena open source Java framework
https://jena.apache.org
[accessed 8 Sept. 2016]

[18] Endsley, M. R., “Design and evaluation for situation awareness enhancement”.
In Proceedings of the Human Factors Society 32nd Annual Meeting (pp. 97–101). Santa Monica, CA: Human Factors Society, 1988

[19] Federal Emergency Management Agency (FEMA) USA, National Response Framework,
https://www.fema.gov/national-response-framework
[accessed 8 Sept. 2016]

[20] FEMA, National Incident Support Manual
http://www.fema.gov/media-library/assets/documents/24921
[accessed 8 Sept. 2016]