



EnCoRe Project Deliverable

Title: Technical Architecture arising from the second Case Study

Identifier: D2.2

Version: 1.0

Date: 3 May 2011

Status: Final

Authors: Pete Bramhall, Marco Casassa Mont, Gina Kouna

Editor: Azzedine Benameur

Reviewers: The entire EnCoRe project team

Class: Public

Summary

This document is a formal deliverable of the EnCoRe project and contains the definition of the EnCoRe Technical Architecture arising from the project's second Case Study, which is focused on Biobanks.

About the EnCoRe project

The EnCoRe project is an inter-disciplinary research project into informational privacy, undertaken collaboratively by UK industry and academia, and partially funded by the Technology Strategy Board (TP/12/NS/P0501A), the Engineering and Physical Sciences Research Council and the Economic and Social Research Council (EP/G002541/1).

Enabling individual citizens and consumers to retain control over their personal data can be achieved in a number of very different ways that are based on different trust models. When an individual discloses his or her personal data to commercial and other entities, he or she also grants, sometimes implicitly, consent for it to be used for one or more purposes. Subsequent control over the storage, use and onward sharing of that data relies on the notion of trust that the given consent will be respected.

Currently, many organisations' personal data management operations do not justify the trust placed in them by the data subjects. This is for a variety of legal, regulatory, process, economic and technical reasons.

The EnCoRe project's aims are:

- To enable business to adopt scalable, cost effective and robust consent and revocation methods for controlling the usage, storage, location and dissemination of personal data.
- To benefit individuals by providing meaningful, intuitive mechanisms which will allow them to control the use of their personal information held by others.
- To help restore individual confidence in participating in the digital economy and so, in turn, benefit the wider society.

The overall vision of the project is to make giving consent as reliable and easy as turning on a tap and revoking that consent as reliable and easy as turning it off again.

The project partners are:

- Hewlett–Packard Laboratories
- HW Communications Ltd
- London School of Economics and Political Science
- QinetiQ
- HeLEX Centre for Health, Law and Emerging Technologies; University of Oxford
- Warwick Digital Laboratory, University of Warwick

The EnCoRe project runs from June 2008 to February 2012.

Its website is www.encore-project.info and it tweets at www.twitter.com/encore_project

Executive Summary

This document is a formal deliverable of the EnCoRe project. It contains the definition of the EnCoRe Technical Architecture arising from the second Case Study: a BioBank scenario. It also describes a set of functional use cases that can be used in a BioBank. The requirements which guided the design of the architecture were gathered and defined by the legal and social science research within the EnCoRe project through various focus groups and by direct discussions with the Oxford Radcliffe BioBank.

The scope of the EnCoRe Technical Architecture for second Case Study encompasses all the technical functions required for the management of data subjects' privacy preferences and choices and the enforcement of individuals' consents that are pertinent to the Case Study's scenario. The technical architecture is the block-level design of the necessary technical system, at the level of functional blocks (i.e., software and service components) and the data flows between them and to/from humans, other technical systems, compliance and other business processes and regulatory environments. Its goal is to provide the basis for an EnCoRe reference implementation that validates the approach and the technology. To that end this document's approach is to start with contextual information and overviews, and incrementally refine the level of detail and ground it to the scenario's usage by presenting functional use cases. Obligation policies are further detailed in the Appendix.

Document History

Doc ID	Description	Date
V1.0	Final Document	3 May 2011

Acknowledgements

The editor and authors wish to thank those who reviewed the drafts and also those who contributed ideas and material to this deliverable, especially Dave Lund and George Mourikas from HW Communications Ltd.

Table of Contents

Executive Summary	iii
Document History	iii
Acknowledgements	iii
Table of Contents	iv
List of Figures	vi
Abbreviations	vii
1. Introduction	1
2. Changes from the first Technical Architecture	2
3. High Level Principles & the Technical Architecture	3
3.1 High Level principles	3
3.2 High Level overview of the Technical Architecture	4
3.3 Refined description of the Technical Architecture	6
3.3.1 Internal Workflow Manager	8
3.3.2 External Workflow Manager	9
3.3.2.1 Flexible Sticky Policy Management	9
3.3.3 Cross-Organisational Workflow Manager	10
3.3.4 Policy Negotiation Manager	11
3.3.5 Privacy Consent & Revocation Assistant	11
3.3.6 Obligation Management System	11
3.3.7 Event Manager	14
3.3.8 Risk Assessment	15
3.3.9 Compliance Checking	15
3.3.10 Architectural Security Principles	16
4. Privacy-Aware Access Control Policies and Obligations	17
4.1 Privacy-aware Access Control Policies	17
No changes are introduced in terms of privacy-aware access control policies. The	17
4.2 Obligation Policies	17
5. Design and Deployment Options	19
6. New Use Cases	20
6.1 Functional Use Cases	20
6.1.1 General Functional Use Cases	20
6.1.2 Detailed Functional Use Cases	21
6.1.2.1 FUC 0 - EnCoRe Semantic Compatibility Specification	21
6.1.2.2 FUC 1- Disclosing personal data to another enterprise	24
6.1.2.3 FUC2- Data Sharing Chain	25
6.1.2.4 FUC3- Revocation Request within an Enterprise	26
6.1.2.5 FUC4- Revocation Request from a direct partner	26
6.1.2.6 FUC5- Revocation Request from a partner down the chain	26
6.1.2.7 FUC6- Interaction through a Helpdesk	27

References.....	28
Appendix A: Parametric Obligation Policies.....	29
A.1. Target	29
A.2. Metadata.....	31
A.3. Events.....	31
A.4. Actions	33
A.5. On Violation Actions	34

List of Figures

Figure 1. EnCoRe Second Technical Architecture	4
Figure 2. Refined Architecture	6
Figure 3. Privacy Consent & Revocation Assistant.....	11
Figure 4. Model of the Obligation Management Framework	12
Figure 5. High-level Architecture of the Obligation Management System.....	13
Figure 6. Revised "UI User Views"	22
Figure 7. Semantic diagram with usage rules	23
Figure 8. Disclosing personal data to another enterprise	25
Figure 9. FUC3 Revocation Request within an enterprise	26

Abbreviations

The following abbreviations are used frequently in this document:

C&R	Consent(s) and Revocation(s)
GUI	Graphical User Interface
HCI	Human-Computer Interaction
ICO	Information Commissioner's Office
PD	Personal Data
PDP	Policy Decision Point
PEP	Policy Enforcement Point
OMS	Obligation Management System

1. Introduction

The aim of this document is to present and discuss the second EnCoRe Technical Architecture. The first EnCoRe Technical Architecture [1] was designed to fulfil the requirements of the first EnCoRe case study, centred on employee data and focusing on an organisational context. The second EnCoRe case study, based on a Biobank scenario, adds new challenges. These include the need to support more flexible and compelling privacy-aware policies beyond access control, these including obligation policies and sticky policies. This second architecture aims at providing support for: long-term running transactions involving personal data and related management of consent and revocation; enabling privacy-aware disclosure of personal data to third parties; the notion of privacy-aware workflows involving the processing and manipulation of personal data; the need to map real business operations onto a protected data structure.

This document provides details and a description of the updated architecture, and focuses on the changes from the first architecture. The reader is directed to [1] for context, background and some detailed information.

Although inspired by, and focused on, the specifics of the second EnCoRe case study, this architecture is much more widely applicable than to just those, being suitable for use in other scenarios where an individual (the data subject) discloses his or her personal data to an organisation, which may wish to disclose it to other organisations. Its legal ability to do so may depend on the specific details of the consent, granted by the data subject at the time of disclosure. At that time, the data subject may not be fully aware of the implications of granting consent, and/or may select the simplest consent options offered by the organisation. Later, perhaps after becoming more aware of these implications, or having just changed her mind, the data subject may wish to revoke the previously granted consents and be sure that her new wishes will be respected by all the organisations that have (or have access to) copies of the personal data she disclosed. In order for this to be, a complex set of interactions, between and within the involved organisations, is required. This architecture provides the framework for these.

As with the first EnCoRe Technical Architecture, the organisations considered here are assumed to be non-state organisations, and hence this document uses the terms “organisation” and “enterprise” to refer to these equivalently. However, many of the points made in this document are also equally applicable to state organisations.

2. Changes from the first Technical Architecture

The major changes from the first technical architecture include but are not limited to:

- *Removal of the Notification Manager:* this component's initial role was to offer basic obligation support, primarily based on the need to notify data subjects whenever their data was used or shared. It is now replaced by a fully fledged Obligation Management System (OMS). This new component is able to manage more complex obligations in the context of long-term running transactions, as needed in the second case study;
- *Enhanced Consent and Revocation Provisioning:* this effectively provides internal workflow coordination when handling Personal Data (PD), data subjects' preferences and choices, and data disclosure;
- *Cross-Organizational Workflow Manager:* this component enables privacy-aware interactions of an organisation with other organisations when disclosing Personal Data and handling changes in consent and/or revocation choices;
- *Flexible Sticky Policies Management:* this component handles the disclosure of Personal Data between parties in a way that is consistent with agreed policies, obligations and consent/revocation choices. Relevant high-level policies (along with choices) are "attached" to Personal Data, potentially with varying degrees of stickiness, and then sent across boundaries between parties via the new Cross-Organization Workflow Manager component.
- *Policy Negotiation Module:* this component offers a way for data subjects to negotiate fine-grained choices on their attributes, based on the data subjects' preferences and what the organisation is willing to support.
- *Virtual Data Registry:* this component is almost the same as described in the first technical architecture, except for one feature. It now keeps track not only of where the data are disclosed, but also their provenance to support the revocation features introduced by Case Study 2.
- *Compliance Checking Component:* this component checks compliance with legal, regulatory and/or organisational requirements.
- *Semantic Configuration Component:* this component is a configuration component that enables EnCoRe systems to interact and share the same semantics while protecting the inner structure of their databases from visibility by others.

3. High Level Principles & the Technical Architecture

In this section we consider firstly the high level principles that guide the design of this second EnCoRe Technical Architecture, and then the technical architecture itself.

3.1 High Level principles

The principles that guide the design of this architecture are the following:

- Individuals do not necessarily have to share all their personal data during interactions with organisations. They can agree or refuse to disclose certain personal attributes during a negotiation phase, this being driven by policies supported by the organisation. These policies can be flexibly configured by the organisation to enable different types of data disclosure and negotiations with data subjects;
- Individuals may express fine-grained preferences on their personal data that dictate privacy obligations to be fulfilled by the organisation. These include: imposing limitations on access and disclosure of data; the request to be notified in specific circumstances; the duty of sanitising, minimising or deleting data after predefined periods of time. An Obligation Management System is introduced to enable the organisation to deal with these duties. Specifically, the data subject can express relevant preferences that affect and constrain predefined types of obligations, supported by the organisation;
- Data subjects' consent and revocation choices, along with access control constraints and obligations, are also enforced when transmitting personal data across organisational boundaries. Sticky policies, along with operational workflows, have been introduced to support this capability. Ideally, this principle can be easily supported in the case when multiple EnCoRe systems are deployed across the involved organisations. However this is not a strong requirement, as organisations might deploy their own solutions, so long as these are EnCoRe-compliant in terms of how they handle relevant policies and privacy choices associated to disclosed personal data .

3.2 High Level overview of the Technical Architecture

The diagram below provides a high level overview.

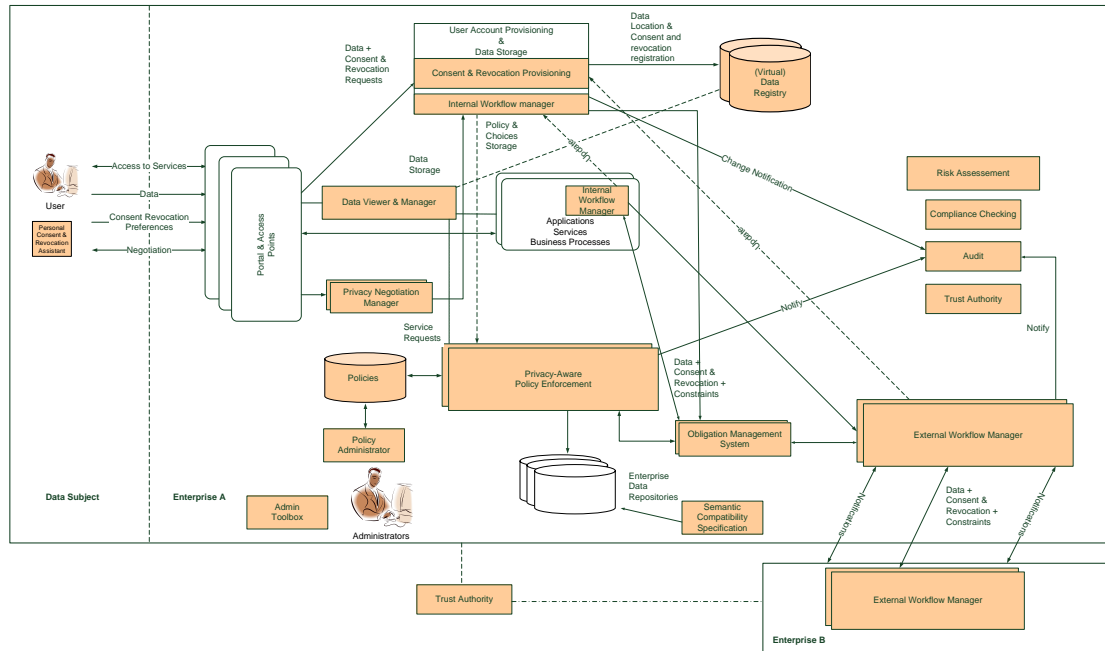


Figure 1. EnCoRe Second Technical Architecture

Compared to the first EnCoRe Technical Architecture, new components have been added to address the new requirements, namely:

Internal Workflow Manager: this component orchestrates the sequence of privacy management tasks needed to be executed on personal data and choices within the organisation. This includes the initial disclosure of personal data, preferences and choices, along with any subsequent changes. It also interacts with the External Workflow Manager component when personal data and related choices are transmitted across the external boundaries of the enterprise.

Semantic Compatibility Specification: this component is used in the deployment phase of an EnCoRe system. It allows mapping high-level terms such as name, date, time, etc, to database tables/rows. In this way two EnCoRe systems communicate without disclosing any information about the inner structure of their databases, thus reducing security risks.

Admin Toolbox: this component is used by EnCoRe administrators to deal with EnCoRe configuration matters, such as defining consent and revocation options, authoring policy templates, etc.

External Workflow Manager: this component manages the interactions and collaborations with third parties. One essential function is to ensure that the agreed privacy choices, access control policies and obligations associated to personal data are

also enforced by third parties. This component deals, *inter alia*, with the management of sticky policies, e.g., the binding of the personal data with cryptographic mechanisms to ensure that data subject choices are fully respected even when their personal data goes across the boundaries of the enterprise.

Obligation Management System: this component manages the privacy lifecycle of personal data, i.e., it enables the organisation to deal with duties and expectations set by the data subject on how to handle personal data during a potentially long period of time. It enables the organisation to explicitly define obligation policies and enforce them, driven by data subjects' choices of the available privacy options. Specifically, it ensures the proper fulfilment of the privacy obligation choices expressed by the data subject regarding notifications, deletion and data minimisation, etc.

Privacy Negotiation Manager: this component allows data subjects to negotiate selected privacy preference attributes, both at the time of initially disclosing personal information and subsequently, in the case when the data subject decides to change them. This negotiation, resulting in the data subject's choices of offered options that provide the best fit to their preferences, is driven by an organisation's privacy-aware policies, and also provides for degrees of further customisation. This makes the overall specification of privacy policies (which depend on these choices) more flexible. It also embeds a knowledge-based repository to reduce unnecessary interactions by taking advantage of past negotiations, and keeps a history of the elements previously shared to raise warnings where appropriate.

Risk Assessment: this component assesses security and privacy risks related to the management of personal data within and across organisations. In this architecture we just introduce a placeholder for this functionality. The exact set of capabilities supported by this component will be subsequently defined by other activities within the EnCoRe project.

Compliance Checking: this component checks compliance with legal, regulatory and/or organisational requirements. In this architecture we just introduce a placeholder for this functionality. The exact set of capabilities supported by this component will be subsequently defined by other activities within the EnCoRe project.

The main interaction flow involved in this architecture is consistent with that defined in [1]. However, a few additional steps have now been introduced:

1. At the time of disclosure of personal data, the data subject has some degree of negotiation over which attributes to release and which privacy preferences' values to associate to them, based on the organisation's needs and supported access control and obligation policies;
2. The Consent and Revocation Provisioning component is now explicitly driven by the Internal Workflow Manager. This has knowledge of the various steps and processes to be followed to: store personal data and related choices;

configure the involved privacy-aware access control policies; instantiate and configure the involved obligation policies;

3. The Obligation Management System is now part the overall interaction flow. It is configured based on the constraints defined by the organisation’s policies and on the choices expressed by data subjects at the time of disclosing their personal data and/or in case of changes or revocation of consent. This happens by customising template obligation policies based on data subjects’ choices, to handle the specific duties and constraints on their personal data. Part of the interaction flow is affected by the Obligation Management System in the cases when relevant events are detected (e.g., time, data accesses, disclosures, attacks, etc.); in these cases the system ensures that the relevant obligations (e.g., involving notifications, minimisation or deletion of data, etc.) are enforced and subsequently monitored. These activities might further involve interactions with the data subjects, e.g., for notifications or explicit authorizations, as expressed in their privacy choices;
4. Any interaction with the external world involving disclosure of personal data will be mediated by the External Workflow Manager. This will be also in charge of dealing with the instantiation and management of sticky policies.

3.3 Refined description of the Technical Architecture

The following figure shows a refined representation of the second EnCoRe technical architecture. The core principles and underlying mechanisms are the same as for the first EnCoRe technical architecture, but with the differences mentioned in the previous section.

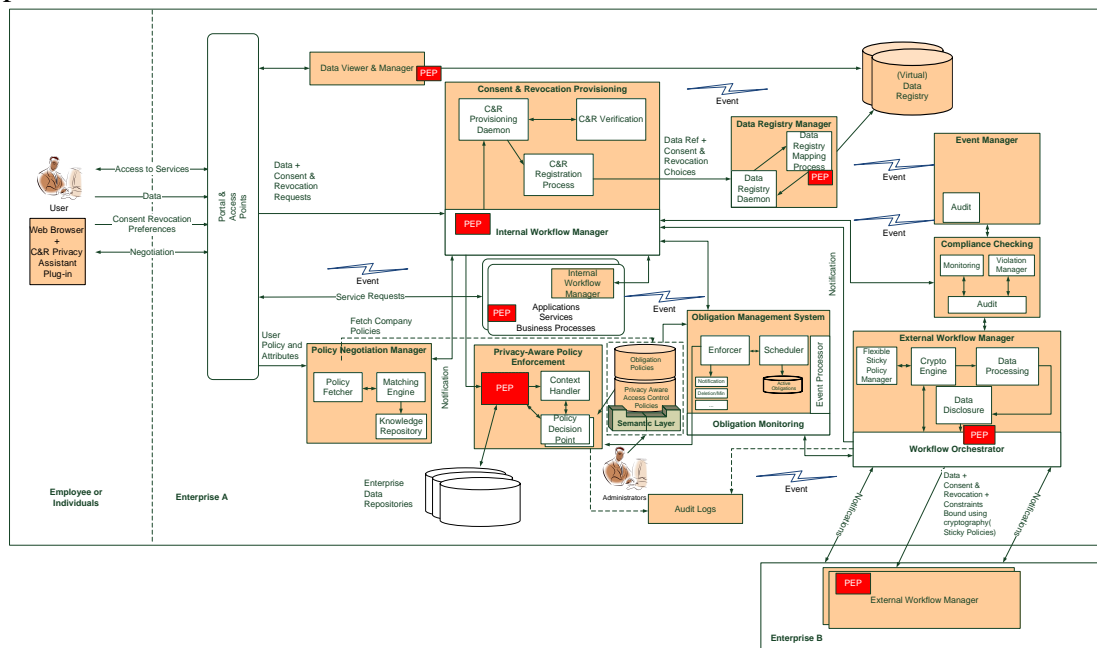


Figure 2. Refined Architecture

The following list contains the general assumptions (and requirements) which were made for the first EnCoRe technical architecture:

1. The second EnCoRe Technical Architecture must be flexible, extensible and general purpose. It must be widely re-usable for EnCoRe Case Study 3.
2. We make the assumption that all components of the architecture are secured following good security practice and can be used as standalone service components.
3. We make the assumption that the data subject's system (e.g., web browser) that accesses EnCoRe is secured and cannot be subject to cross-site scripting or cross-site request forgery attacks.
4. We make the assumption that the platforms and systems running the back-end EnCoRe components are secured and observe good security practices.
5. An EnCoRe compliant system shall conform to the EnCoRe architecture and high level principles.
6. Proper operational security mechanisms are in place to provide secure communication channels between EnCoRe components and for communications with third parties
7. Data security mechanisms are in place to ensure different degrees of binding of policies to data. This will be dictated by business, security and privacy needs. It might involve the use of cryptographic solutions for confidentiality and non-repudiation purposes

The following list contains the specific assumptions (and requirements) of relevance for this second EnCoRe technical architecture:

1. A data subject can minimize the set of shared personal data using a negotiation protocol via a web browser.
2. A data subject can express fine grained consent and revocation preferences and choices, the latter dictating agreed obligations and duties to the organisations.
3. Data subjects will manage their data-related consents and choices only by interacting with the system to which they originally disclosed their information.
4. The data subject's preferences and choices (and a trace of consent/revocation changes) will be stored locally within his/her web

browser (using an HTML5-enabled browser) and/or in any trusted location (e.g., network file system, service in the cloud, etc.) chosen by him/her.

5. A data subject will be given privacy advice and support, by their local Privacy Assistant component, regarding potential violation (by organisations) of agreed privacy choices and historical information about these.
6. System-wide events can be used to support compliance checking and risk management.
7. The interaction is limited to an EnCoRe compliant system or a legacy system which uses an EnCoRe compliant external workflow manager

The remaining part of this section provides additional details about the new components and concepts introduced in this second version of the architecture.

3.3.1 Internal Workflow Manager

This component manages the internal workflows, and interacts with the external workflow manager when data are sent/received from/to another EnCoRe-enabled system (or a legacy system that relies on an EnCoRe-compliant external workflow manager component). It consists of the following types of workflows:

- **iSendDataToExternal:** This workflow is triggered by applications when data need to be disclosed outside the boundaries of the system. It triggers the external workflow manager to prepare disclosure of the data using eSendDataToExternal and eDisclosure, and wait for its response to update the Data Registry using iUpdateRegistry.
- **iReceiveDataFromExternal:** This workflow is triggered by applications which triggers the external workflow manager (eReceiveDataFromExternal) when data are received from outside the system. It then updates the Data Registry using the iUpdateDataRegistry workflow.
- **iUpdateDataRegistry:** This workflow is triggered when a data registry update is needed.
- **iRevocationRequest:** This workflow is triggered when a revocation request is made, it invokes the C&R provisioning to do the necessary tasks to deactivate/delete data, access the data registry to find out if any other partner has this data if it is the case executes eSendRevocationRequest to propagate this change down the chain.
- **iViolation:** Handles violations such as notification to administrators or data subjects that a violation occurred. Details of this are dependent on the EnCoRe compliance framework, to be published at a later date.
- **iViolationReceive:** Receive violations. More details will be added when the EnCoRe compliance framework document is published.

3.3.2 External Workflow Manager

This component manages the external workflows to enable interaction with EnCoRe-compliant third party or legacy systems which rely on EnCoRe-compliant external workflow component. It consists of the following types of workflows:

- **eSendDataToExternal:** This workflow is triggered by the internal workflow manager (iSendDataToExternal) when applications are about to disclose data outside the boundaries of the system. It first executes the eDisclosure workflow that prepares the data prior sending it and then calls the internal workflow manager to execute iUpdateDataRegistry.
- **eReceiveDataFromExternal:** This workflow is triggered by the corresponding external workflow manager when data are received from outside the system. It then triggers iReceiveDataFromExternal workflow.
- **eDisclosure:** This workflow manages the data preparation prior sending. Its main function is to bind the data, policy and metadata using cryptographic operations.
- **eRevocationRequest:** This workflow is triggered by the corresponding external workflow manager when applications are requesting a revocation (deletion, update ...). It then calls the iRevocationRequest workflow.
- **eSendRevocationRequest:** This workflow is triggered by internal workflow manager, to propagate a revocation request down the chain.
- **eNotify:** This workflow is triggered by the corresponding external workflow manager to notify that a data sharing has appended further down the chain or that revocation has occurred. It invokes iUpdateDateRegistry.
- **eViolation:** Propagates violations to external partners such as notification that a violation occurred. Details of this are dependent on the EnCoRe compliance framework, to be published at a later date.
- **eViolationReceive:** Receive violations from external partners such as notification that a violation occurred. Details of this are dependent on the EnCoRe compliance framework, to be published at a later date.

3.3.2.1 Flexible Sticky Policy Management

The external workflow manager supports the management of sticky policies associated to data to be disclosed to third parties. This functionality is provided by a specific component in the architecture, the *Flexible Sticky Policy Manager* within the external workflow manager. This component is instructed by the external workflow manager about the type of binding to be enforced between the personal data and meta-data (inclusive of privacy choices and a suitable abstraction of the access control and obligation policies). We envisage at least two degrees of stickiness: weak and strong. The latter would involve mediation by trusted authority/trusted third party (which

may also be the organisation disclosing the data) to enable the access to the actual data, based on the fulfilment of agreed policies.

The actual level of stickiness of policies required is dictated by business, security and privacy requirements that might vary, based on the organisation's needs, legal framework, contractual agreements with third parties and related level of trust. The *Flexible Sticky Policy Manager* supports different levels of configuration to support different types of stickiness. More details follow.

Weak Stickiness:

The implementation of sticky policies does not necessarily rely on a cryptographic mechanism; a weak version of stickiness can be provided just at the logical level to bind together a set of two elements: the data and meta-data (e.g., abstracted policies, choices, etc.) [2]. The logical binding can be provided by means of a standardised data format to convey the association of these elements in order to form a single package containing all the required information. For example, this might use a blob (Binary Large Object) file. These packages are to be transmitted using best security practice to ensure that the communication channel is secured. It is important to notice that weak stickiness is primarily meant to support the association and links between all relevant data, not really mitigate any threat related to the possibility of breaking them. After having received the personal data, nothing prevents the third party from using these data without enforcing the associated privacy choices. Additional support in this direction is provided by Strong Stickiness.

Strong Stickiness:

Strong stickiness mandates usage of cryptography. This version makes sure that only the intended recipients can access the data and the policies. At least it ensures the data owner that the receiver of the encrypted package will have to fulfil all the requirements and needs mandated by the associated policies - along with issuing non-repudiable statements certifying this - before getting access to the data.

Different mechanisms can be used to achieve this. Cryptographic mechanisms, such as encryption, can be used to bind personal data to privacy choices [3]. If the binding can only be unbound with an encryption key that is disclosed to the third party only after it has agreed to fulfil the privacy choices, then it is possible to obtain a proof that the third party has agreed to enforce the choices before giving it the access to the personal data. But then again, after the decryption key has been issued nothing prevents the third party from using the personal data without enforcing the privacy choices. If stronger mechanisms are needed, the approaches such as those proposed by Zuo and O'Keefe [3] could be considered. However, this type of approach is currently beyond the scope of this architecture.

3.3.3 Cross-Organisational Workflow Manager

This is not a component *per se*, but an abstraction of two existing components: the internal and external workflow managers. In order to co-ordinate cross-organisational workflow, the combination of these two components is needed, as detailed in section 6.1. We refer to this combination as the cross-organisational workflow manager.

3.3.4 Policy Negotiation Manager

In this version we offer a very simplified approach, driven by the organisation and involving negotiation of the level of granularity of different settings. It offers the possibility for the data subject to expand on the granularity of preferences that can be handled by a system.

Supported preferences are stated by the organisation, but the granularity could vary from coarse-grained to fine-grained. The policy negotiation manager provides different views for setting the preferences, giving data subjects control of which level of abstraction to focus on.

3.3.5 Privacy Consent & Revocation Assistant

The Privacy C&R Assistant is a client-side component that supports data subjects in defining and storing their privacy preferences. It keeps a local copy of the choices and preferences expressed for different EnCoRe-enabled services, in order to identify discrepancies that may occur between expected behaviours and actual behaviours (e.g., requiring the deletion of certain personal data after a predefined time period, as part of expressed privacy choices, and later discovering that this data is still in existence). Moreover, depending on the implementation it could be used in the context of providing synchronous and asynchronous notifications to data subjects. The component can store data subjects' choices and preferences on a variety of different media, including: the local file system, an external storage (such as network folders) and a cloud service provider. The latter is designed to handle the case of using multiple client devices (e.g., laptop, home PC and smartphone) and therefore it is necessary to keep synchronized all the clients history by leveraging a cloud synchronisation service.

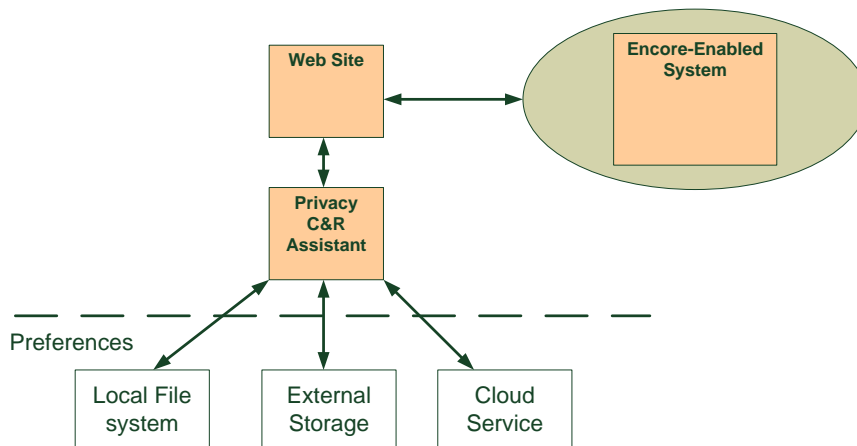


Figure 3. Privacy Consent & Revocation Assistant

3.3.6 Obligation Management System

The Obligation Management System (OMS) is in charge of implementing privacy-aware lifecycle management of data.

This involves enforcing constraints and duties that have been defined both by organisational policies and data subjects' preferences. This includes obligations about

notifying data subjects about usage or disclosures of their personal data; dealing with transformation and minimisation of personal data; dealing with deletions; etc.

It is important to notice that the obligation policies handled by this system do not necessarily only depend on access control events, but could also depend on time events, business process events (e.g., disclosure of data), security events (e.g., detected attacks), etc. This is required to take into account and enforce a wide range of situations and requirements, driven both by organisations and by the needs and preferences of data subjects. As such, the OMS framework and system used in EnCoRe does not suffer from the limitations of existing approaches and frameworks, such as the XACML framework [5].

The architecture of the OMS system (and underlying principles) leverages the capabilities and functionalities developed by HP Labs in previous work [6,7]. Figure 4 illustrates the conceptual obligation management framework:

Model of the Obligation Management Framework

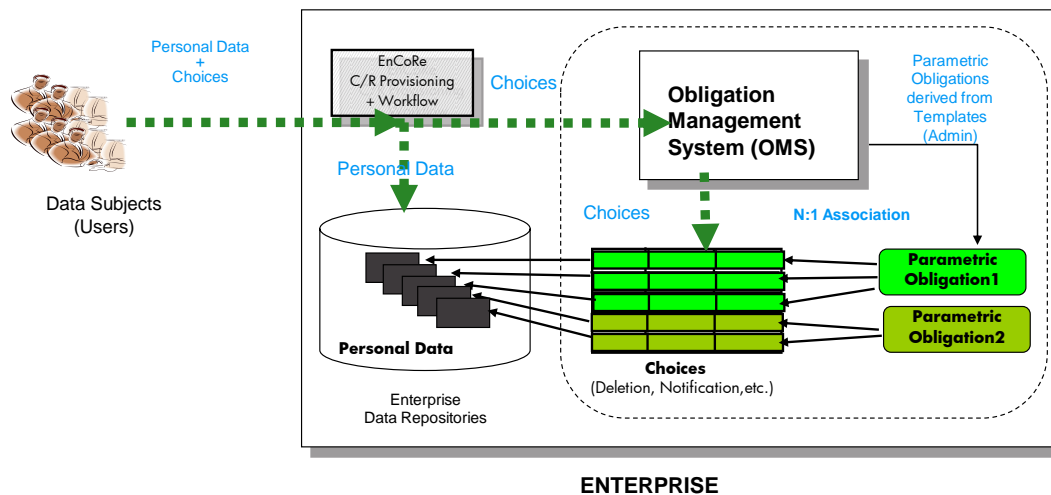


Figure 4. Model of the Obligation Management Framework

The architecture enables the organisation’s privacy administrators to explicitly define obligation policy templates that can be used to capture a set of duties and requirements on how to handle personal data. These templates specify the types of preferences that can be customised by data subjects (e.g., notification requests, time to delete data, criteria for minimising/deleting data, etc.) and default values. Administrators can instantiate templates into parametric obligations that are actively enforced by the OMS system.

At the time of disclosing their personal data (and subsequently when changing consent), data subjects can customise the relevant privacy choices. The OMS system will associate them to the relevant parametric obligations and their specific personal

violations (e.g., failure in notifying a data subject), the obligation monitoring system will take the necessary steps to remediate them, as specified in the obligation policy.

- *Obligation Policy Repository*: this contains a list of obligation templates as well as instantiated obligations, i.e., obligations associated to specific data items, and driven by data subjects' choices.

3.3.7 Event Manager

The Event Manager is a new component, handling and processing events of relevance to the various EnCoRe components, beyond pure access control events. This component supports a subscriber/provider model.

It receives events of relevance from a variety of internal and external sources. It publicises the list of supported events.

Each EnCoRe component can subscribe to a set of these events. When an event happens, the Event Manager will notify the relevant components.

The Event Manager is fundamental to ensure that the Obligation Management System (by means of the Events Handler – see Figure 5) receives a broad set of events, of relevance to trigger the supported privacy obligations.

Below is a table of the events that can be raised by various encore components:

Component	Event	Emits	Receives	Relevance to Compliance
Privacy Negotiation Manager	negotiation_started	X		low
	negotiation_finished	X		low
	negotiation_change	X		
Obligation Management Systems	obligation_triggered	X		medium
	obligation_stored	X		high
	obligation_enforced	X		medium
	obligation_violated	X		high
	obligation_monitored	X		
	obligation_change	X	X	
Internal Workflow Manager	update_registry	X	X	medium
	revocation_request_received	X	X	high
	change_request	X	X	
	external_data_received	X	X	high
	external_data_send	X		high
External	external_data_received	X	X	high

Workflow Manager	external_data_send	X		high
	secure_data_disclosure	X		medium
	change_request	X	X	
	revocation_request_received	X	X	
	3rd_party_notification	X	X	medium
System Wide	time_HH_DD_MM_YY	X	X	medium
	compliance_violation	X	X	high
	attack_detected	X	X	high
	component_failure	X	X	high

3.3.8 Risk Assessment

The Risk Assessment component is not yet fully specified. This is primarily a placeholder to be fully specified in the third EnCoRe Technical Architecture. Overall, it assesses security and privacy risks. It will require the specification of multiple technical components to enable this capability. For example, on the client side, a component could provide information to the end users about potential involved risks and how they are mitigated. On the server side, a component could provide ways to assess overall security and privacy risks, risk exposure and how the EnCoRe architecture mitigates them. It could also assess risk related to sharing of information across parties: understanding the involved risks and carrying out assessment of the trustworthiness of third parties.

3.3.9 Compliance Checking

The Compliance Checking component is not yet fully specified. This is primarily a placeholder to be fully specified in the third EnCoRe Technical Architecture. Overall, it checks compliance with legal, regulatory and/or organisational requirements. This could have an implication for data subjects, in the sense that some assurance needs to be provided that ideally should differentiate from a standard privacy seal. It also has implications for both organisations and the third parties with whom they share information, in the sense that compliance must be checked for all these entities in some way.

At the current stage we envisage this component including of the following core modules:

- **Monitoring:** this component monitors the queue of past events stored in the audit component and present events fed by the Event Manager.
- **Violation Manager:** this component identifies violation patterns, store them in the audit components and finally trigger the appropriate workflows when a violation occurs. It can propagate violation to external partner by interacting with the external workflow manager (that will trigger eViolation workflow).
- **Audit:** this component is used by the two component described above.

The compliance checking component interacts with the internal and external workflow manager to propagate or be notified that a violation occurred.

3.3.10 Architectural Security Principles

EnCoRe makes a distinction between two different aspects of security:

- Operational Security
- Data Security

Operational security ensures that: secure communication channels (e.g., SSL/TSL) are present to secure communications between internal component of an EnCoRe system and across the involved parties. This is necessary for the transmission of data and policies. Confidentiality and non-repudiation techniques (e.g., encryption and digital signature) need to be in place when transmitting information across organisations' external boundaries. It also ensures that security best practices are implemented, including proper setting of access control to the hosting systems, patching and updates of the underlying software solutions, etc.

Data Security primarily builds on top of operational security and relies on the additional sticky policies mechanism to ensure that data do not leave the system without their associated policies. While the term sticky policies might imply cryptographic binding it is not necessarily always so. As discussed, a weak sticky policy can simply be a logical binding of policy, data and meta-data, where the glue is a simple logical binding.

Refer to [1] for further information about this.

4. Privacy-Aware Access Control Policies and Obligations

This section provides an overview of the different types of policies that are handled by the components of the second EnCoRe Technical Architecture. It is beyond the scope of this document specify these policies. However, the aim it is to create awareness about the types of constraints that are managed and how they relate to data subjects' preferences and different types of enforcement and monitoring requirements.

4.1 Privacy-aware Access Control Policies

No changes are introduced in terms of privacy-aware access control policies. The specifications and requirements described in [1] are still valid, and are used in this architecture.

4.2 Obligation Policies

This second EnCoRe Technical Architecture introduces the concept of Obligation Policies, and specifically the core concepts at the base of Parametric Obligation Policies, to be managed and handled by the OMS system.

A Parametric Obligation Policy can be conceptually represented as:

FOR: Target

WHEN Events(Refs)

THEN EXECUTE [Actions(Refs)]

ON VIOLATION:

EXECUTE [Violation-Actions(Refs)]

For a given target (personal data), when specific (parametric) events happens then a set of (parametric) actions are executed. In case of violation of the policy a set of on-violation actions are specified to remediate the issues.

More specifically:

- A Parametric Obligation Policy can be associated to a potentially large set of personal data (i.e., no multiple instantiations) and, at the same time, it can dictate customized obligation constraints (based on data subjects' privacy choices) on each data item;
- A Parametric Obligation Policy does not embed privacy choices in its Events and Actions sections. Instead, this policy contains explicit references to these choices, which are stored elsewhere (in data repositories);
- The Target section of Parametric Obligation Policies explicitly model and describe the data repositories that will contain privacy choice values

pointed by these references - in addition to repositories containing personal data;

- An On-Violation section automates the process of remediation of violated obligations.

The key feature of parametric obligations is that privacy choices are stored separately from parametric obligation policies: references are used to retrieve these choices. This ensures that a parametric obligation policy can apply to a potentially large set of personal data – as defined in its Target element – and, at the same time, allows the “customization” of its Events and Actions based on references to external privacy choices.

A set of parametric obligation policies can be created by a privacy administrator to dictate the criteria by which personal data should be handled. The referencing mechanism (coupled to appropriate data descriptions in the Target section) ensures that these policies are instantiated on-the-fly by the Obligation Management System, based on associated privacy choices, and enforced and monitored on a potentially large set of managed data.

From an operational perspective a parametric obligation policy can be represented as an XML format, as a reactive rule. XML can be used for its specification because of its versatility and suitability to extensions. The XML skeleton of a parametric obligation policy is:

```
<?xml version="1.0"?>
<obligation oid="">
  <target> ..... </target>
  <metadata> ... </metadata>
  <events> .....</events>
  <actions> .....</actions>
  <onViolation> ...</onViolation>
</obligation>
```

More details are available in Appendix A.

5. Design and Deployment Options

The design and deployment options for this second EnCoRe Technical Architecture are the almost the same as for the first. The only difference is that to enable interoperability with other EnCoRe systems the administrator defines the semantics by the EnCoRe Semantic Compatibility Specification as described in 6.1.2.1.

Specifically, this architecture strongly assumes compliance to the best security practices, as discussed in section 3.3.10. This includes designing and deploying all the core EnCoRe components as secured, self contained services along with a clear definition of their APIs and secured interaction protocols. This approach enables a flexible deployment of the architecture, based on actual needs dictated by the target case study, together with business and security requirements.

6. New Use Cases

This architecture has been designed by factoring in the requirements of the following use cases which are additional to those considered in [1].

- A data subject interacting with an EnCoRe-enabled system can now define a subset of his/her personal data. This subset represents the data he/she would absolutely not want to share. Moreover, he/she can express a sharing option on each attribute of his/her personal data. In the first EnCoRe Technical Architecture, an EnCoRe-enabled system had a Default Policy, and the data subject sets his privacy choices that are encapsulated in its User Policy. But if these two do not match, the interaction with the system stops at this stage. However, this architecture enables flexible policy negotiation;
- Enterprises can now not only manage the access to personal data in a privacy-aware way but also support a privacy-aware lifecycle management of this data;
- Enterprises can now send data across their external boundaries and ensure that external entities respect the choices expressed by the data subject in the first place.

The remaining part of this section discusses the functional use cases.

6.1 Functional Use Cases

To illustrate the core functionalities of this architecture, we present a set of functional use cases. We consider three enterprises A, B and C. Enterprise A is where the data subject sets his choices and has his first contact with an EnCoRe system. Enterprise B is a direct partner of A. Enterprise C is a direct partner of B. Enterprise A, B and C are all using EnCoRe-compliant systems. These use cases are based on the specific assumptions and requirements listed in section 3.3.

6.1.1 General Functional Use Cases

We consider a set of general scenarios, involving a data subject and enterprises A, B and C, based on the assumptions mentioned previously.

Scenario: A data subject interacts with enterprise A, and sets his consent and revocation choices on his personal data. For this purpose, the EnCoRe system of enterprise A uses the policy negotiation manager to help him make his choices, and invokes the internal workflow manager which triggers a workflow involving the C&R provisioning component to call the appropriate services (Data Registry, Access control, Obligation Management...). At some later point the data subject decides to revoke his granted consents to the use of his personal data, the internal workflow manager initiates the appropriate actions to fulfil this request.

Scenario: A data subject has given consent to enterprise A to use his data and to share it with one partner: enterprise B. When he asks enterprise A to revoke usage of his

personal data, enterprise A contacts the external workflow manager of enterprise B and asks for revocation, and when done A is notified and proceeds to action the revocation within its system.

Scenario: A data subject has given consent to enterprise A to use his data and to share it with N partners. Enterprise A has shared his data with enterprise B and C. When the data subject asks for revocation, enterprise A contacts enterprise B and asks for revocation. When done, A is notified and asks C for revocation. When done, A proceeds with revocation within its internal system.

These scenarios can be decomposed into four functional use cases detailed in the next section.

6.1.2 Detailed Functional Use Cases

6.1.2.1 FUC 0 - EnCoRe Semantic Compatibility Specification

This is part of the EnCoRe system installation in enterprise A.

Enterprise A is where the Data Subject (DS) is going to place his/her trust regarding the management of his/her data, privacy preferences and choices. Before the DS starts disclosing data to enterprise A, enterprise A needs to become EnCoRe-compliant and therefore adopt a baseline semantic convention to form:

- Enterprise Policies (formed by the Data Controller – see Figure 6),
- Obligation technical policies (i.e., the actual implementation of the parametric obligation policies) and the Access Control technical policies used to specify the Enterprise Policies by the EnCoRe IT admin – see Figure 6.

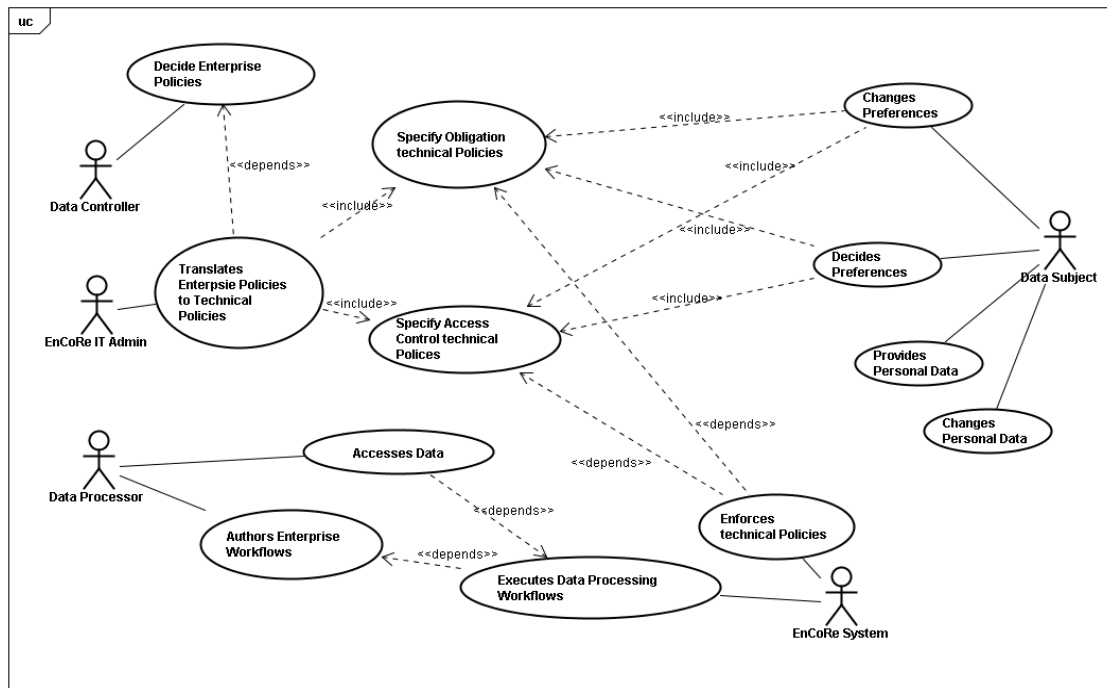


Figure 6. Revised "UI User Views"

The EnCoRe Semantic Compatibility Specification, used by enterprises in initial EnCoRe setup, is part of the Common Specifications component. The Common Specifications component may be a component in the EnCoRe architecture outside the Enterprise components alongside the Trust Authority. The EnCoRe Semantic Compatibility Specification includes semantics used to form the baseline terminology for:

- the Administrator, to build the different repositories within the Enterprise according to Semantics included in the EnCoRe Semantic Compatibility Specification and
- the Extended Semantics, which will be the local Semantic repository to each Enterprise and it is updated according to the individual Enterprise Semantics,

The “Obligation technical policies” and “Access Control technical policies” natural language editors and all GUI’s interactions will use semantics within the Extended Semantics which is built by the enterprise’s EnCoRe IT Admin.

The External Workflow Manager of each Enterprise will have to use this Semantics component when enterprise-to-enterprise sharing is needed, as shown in Figure 7.

Extended Semantics (local to any enterprise) is the “Semantics” web-based file from the EnCoRe Semantic Compatibility Specification. It is kept as a local copy and extended according to the internal Enterprise data items types in the Data Registry, Data Repository, available DS privacy options and enterprise Obligation Policy.

The terms in the Extended Semantics are used as a naming convention for the:

- Client browser to display the privacy options available to the DS (that form eventually “Access Control technical policies” and “Obligation technical policies”),
- “Access Control technical policies” editor natural language terms marked with $\langle \rangle$,
- “Obligation technical policies” editor the natural language terms marked with \diamond
- Data sharing between different enterprises.

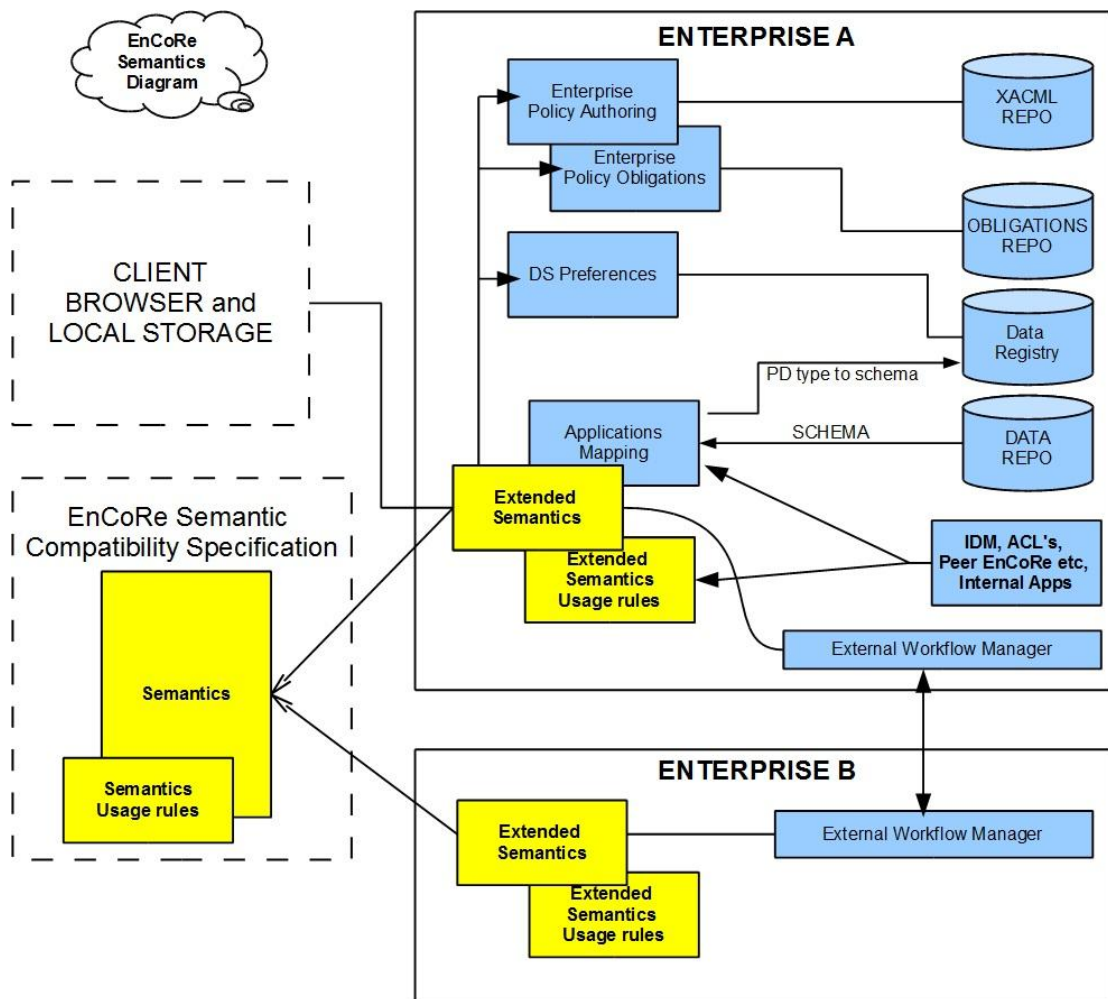


Figure 7. Semantic diagram with usage rules

To clarify the difference between core Semantics and local Extended Semantics, the list of purposes in the core Semantics may be

1. data sharing with external organisations
2. data sharing within other systems in the same enterprise
3. internal processing. This list may be extended in a local Extended Semantic of an enterprise.
4. statistical marketing research

The first three purposes will be common on all local Extended Semantics blocks of all EnCoRe compliant enterprises.

The EnCoRe Semantic Compatibility Specification can be viewed as an architecture component outside the enterprise within a Common Specification block that will include specifications on how to include the EnCoRe architecture into an existing enterprise architecture.

The EnCoRe Semantic Compatibility Specification will interface with the Extended Semantics (which is local class to every EnCoRe equipped Enterprise).

The Extended Semantics class is interfaced within the Enterprise' s GUI interacting with the Data Subject, the Data Controller, the EnCoRe IT administrator and the External Workflow Manager so that the collaboration between EnCoRe-enabled systems has a common set of semantics.

6.1.2.2 FUC 1- Disclosing personal data to another enterprise

A data subject, after setting his privacy choices, interacts with an application within enterprise A. This application is about to disclose data outside the boundaries of A. The following events occur:

1. The application calls the internal workflow manager to trigger `iSendDataToExternal`.
2. The internal workflow manager calls the external workflow manager to trigger `eSendDatatoExternal`.
3. The external workflow manager prepares the data and sends them using `eDisclosure` workflow to enable sticky policies.
4. Finally the external workflow manager calls the internal workflow `iUpdateDataRegistry` to update the data registry.

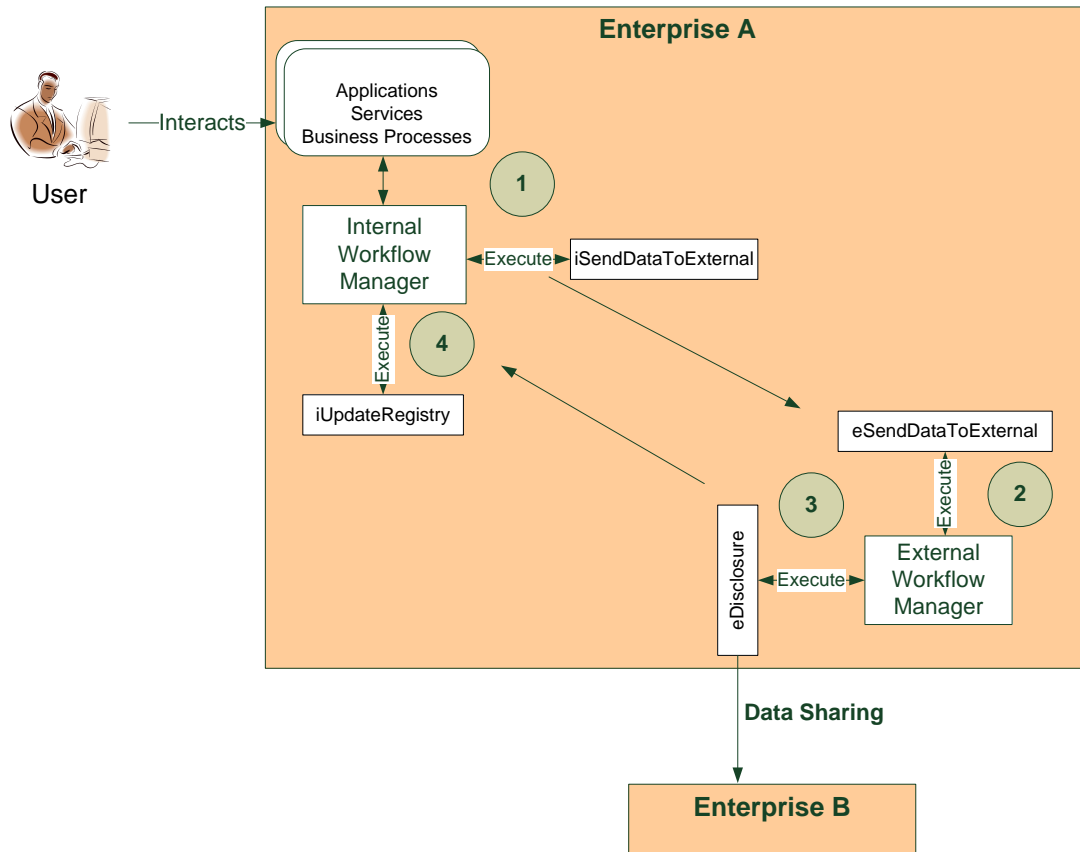


Figure 8. Disclosing personal data to another enterprise

As can be seen, data disclosure is performed according to a pull model; however a push model can be defined as well using the existing workflows.

6.1.2.3 FUC2- Data Sharing Chain

Enterprise B wants to share personal data with enterprise C, provided that the data and the choices expressed by the data subject to which it refers allow it. The data originates from enterprise A where the data subject had his first interaction with an EnCoRe-compliant system. The following events occur:

1. An application within enterprise B calls the internal workflow manager to trigger iSendDataToExternal workflow.
2. The internal workflow manager calls the external workflow manager to trigger eSendDataToExternal.
3. The external workflow manager prepares the data and sends them using eDisclosure workflow.
4. The external workflow manager calls the internal workflow iUpdateDataRegistry to update the data registry.
5. The external workflow manager from Enterprise B invokes the external workflow manager from enterprise A to execute the eNotify workflow (which will execute the internal workflow iUpdateRegistry)

6.1.2.4 FUC3- Revocation Request within an Enterprise

Within enterprise A, a data subject wants to revoke a previously given consent to the usage of his data. The following events occur:

1. The application calls the internal workflow iRevocationRequest.
2. The application invokes the internal workflow iUpdateDataRegistry.

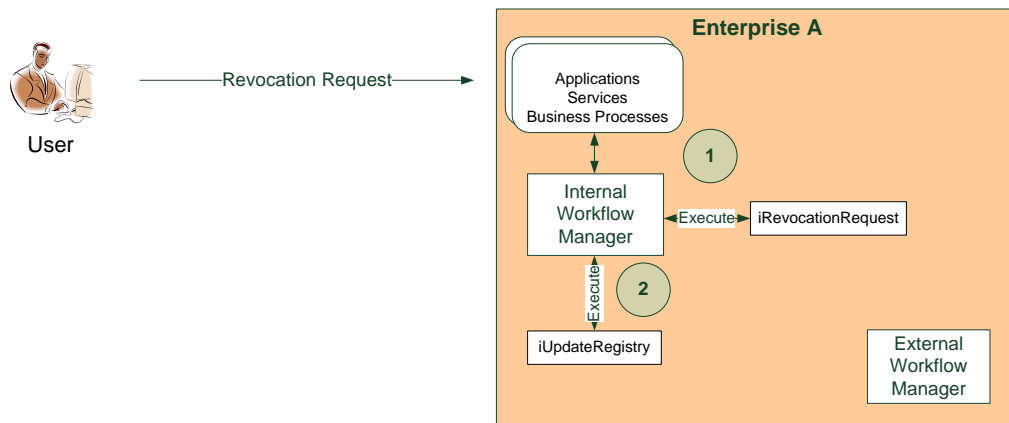


Figure 9. FUC3 Revocation Request within an enterprise

6.1.2.5 FUC4- Revocation Request from a direct partner

Enterprise B is a direct partner of enterprise A. A data subject accessing an application within enterprise A revokes a previously given consent to the usage of his personal data. The following events occur:

1. The application within A calls the internal workflow iRevocationRequest.
2. The internal workflow manager calls the external workflow manager to execute eSendRevocationRequest.
3. The external workflow manager from enterprise A invokes eRevocationRequest workflow from the external workflow of enterprise B.
4. The external workflow manager from enterprise B invokes the internal workflow iUpdateDataRegistry.
5. The external workflow manager from enterprise B invokes the external workflow manager from enterprise A to execute eNotify.

6.1.2.6 FUC5- Revocation Request from a partner down the chain

Enterprise C received data from enterprise B which originates from enterprise A. A data subject accessing an application within enterprise A revokes a previously given consent to the usage of his personal data. The following events occur:

1. The application within A calls the internal workflow iRevocationRequest.
2. The internal workflow manager calls the external workflow manager to execute eSendRevocationRequest.

3. The external workflow manager from enterprise A invokes eRevocationRequest workflow from the external workflow of enterprise B.
4. The external workflow manager from enterprise B invokes the internal workflow iUpdateDataRegistry.
5. The external workflow manager from enterprise B invokes the external workflow manager from enterprise A to execute eNotify.
6. The internal workflow manager from enterprise A calls the external workflow manager to execute eSendRevocationRequest.
7. The external workflow manager from enterprise A invokes eRevocationRequest workflow from the external workflow of enterprise B.
8. The external workflow manager from enterprise C invokes the internal workflow iUpdateDataRegistry.
9. The external workflow manager from enterprise C invokes the external workflow manager from enterprise A to execute eNotify.

6.1.2.7 FUC6- Interaction through a Helpdesk

In this use case we consider the situation where a data subject needs to perform an operation on his personal data and/or consent, mediated by a helpdesk officer over the phone. For the purpose of this architecture we consider this use case as a specific example of dealing with Privacy-Aware Access Control where:

- A new role is introduced: Helpdesk Officer.
- Preferences can be expressed by a data subject about which activities can be carried out by an Helpdesk Officer on data subject's personal data.
- The data subject can grant or revoke consent on activities which can be carried out by Helpdesk Officers.

In this context, this use case is a specific example of the use cases presented in [1] involving personal data disclosure, settings of consent or revocation and access of data by a third party mediated by the enforcement of Privacy Aware Access control policies. As such, no additional architecture components, beyond the ones already described in this document and in [1], are required to address it.

References

- [1] D2.1 Technical Architecture for the first realized Case Study, http://www.encore-project.info/deliverables_material/D2.1%20EnCoRe%20Architecture%20V1.0.pdf
- [2] G. Karjoth , M. Schunter , M. Waidner: Platform for Enterprise Privacy Practices: Privacy-enabled Management of Customer Data, <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.20.4522>
- [3] M. Casassa Mont, S. Pearson, P. Bramhall: Towards Accountable Management of Identity and Privacy: Sticky Policies and Enforceable Tracing Services, <http://www.hpl.hp.com/techreports/2003/HPL-2003-49.pdf>
- [4] Y. Zuo & T. O'Keefe, Post-release information privacy protection: A framework and next-generation privacy-enhanced operating system, <http://www.springerlink.com/content/03718003288553u5/>
- [5] eXtensible Access Control Markup Language (XACML), <http://www.oasis-open.org/committees/download.php/2406/oasis-xacml-1.0.pdf>
- [6] M. Casassa Mont: A System to Handle Privacy Obligations in Enterprises, <http://www.hpl.hp.com/techreports/2005/HPL-2005-180.pdf>
- [7] M. Casassa Mont, F. Beato: On Parametric Obligation Policies: Enabling Privacy-aware Information Lifecycle Management in Enterprises, <http://www.hpl.hp.com/techreports/2007/HPL-2007-7.pdf>
- [8] LabVantage, Sapphire solution, <http://www.labvantage.com/index.aspx>

Appendix A: Parametric Obligation Policies

For illustration purposes we consider a simplified scenario to illustrate, in more details, the specification of parametric obligation policies. Additional information is available in [7].

This scenario consists of an enterprise site that collects personal data about data subjects, their privacy preferences and choices, and stores this information in database tables. In this context, we consider a very simple parametric obligation policy dictating that: for each piece of managed personal data (*Target*), credit card information must be deleted (*Parametric Action*) based on time-based deadlines specified by data subjects via their privacy choice (*Parametric Event*). When this happens the corresponding data subject must be notified (*Parametric Action*). Should the enforcement of any of these actions fail, the obligation management system should try to reinforce them and notify an administrator (*“On Violation” Actions*).

A.1. *Target*

The *Target* section of a parametric obligation policy is used to provide the following information:

- **A description of data repositories containing (personal) data** that is subject to privacy obligations. In this context one or more data repositories can be described (e.g., RDBMS database or LDAP directory, etc.). A data repository description includes location and name of the data repository, data schema structures (e.g., database tables) and primary keys. It is important to notice, that by default, all data stored in these repositories will be affected by this obligation policy. A more selective choice of which data items must be managed can be made by instantiating a “Conditions” sub-section (e.g., by testing properties/values of the stored data). Each data repository is identified by a unique *alias* that is used as a shortcut in other parts of the parametric obligation. If multiple data repositories are described, it is possible to specify any relationship (i.e., links between primary keys) existing on data stored in these repositories;
- **A description of data repositories used to store privacy choices.** The definition of this sub-section is identical to the previous one, with the exception that it refers to repositories storing privacy choices/parameters, e.g., the EnCoRe Data Registry. These choices are associated to the managed personal data and used to customize other sections of the privacy obligations;
- **A cross-links sub-section** defining how to link *choices* to *personal data*, by using relevant keys defined in the other two sub-sections.

The XML skeleton of the *Target* section (low-level details have been omitted for space reasons) follows:

```

<target>
  <DataRepositories>
    <Repositories>
      <DataRepository alias= “...”>
        <DRType> ... </DRType>
        <DBname> ... </DBname>
        <TableName>.....</TableName>
        <Conditions>
          <Condition> .....</Condition>
        </Conditions>
        <UniqueIdentifier>
          <References> ... </References>
        </UniqueIdentifier>
      </DataRepository>
    <InternalLinks>
      <Link> .....</Link>
    </InternalLinks>
  </DataRepositories>
  <ChoiceRepositories>
    <Repositories> ... </Repositories>
    <InternalLinks>
      <Link>..... </Link>
    </InternalLinks>
  </ChoiceRepositories>
  <CrossLinks> ..... </CrossLinks>
</target>

```

In case of our simple example of privacy obligation policy, the above skeleton could be instantiated with the following information: (1) a *data repository* entry, containing the database and table names where personal data is stored, the table’s “primary key”

(e.g., `UserId`) and an alias (e.g., `DataRepAlias`) for this repository; (2) a *choice repository* entry, containing the database and table names where privacy choices are stored, the table's "primary key" name (e.g., `PrefId`) and an alias for this repository (e.g., `PrefRepAlias`). A field in this table, for example called `TimeChoice`, could be used to store data subjects' choices about *deletion time* of Credit Card details; (3) a description (in the "Cross link" sub-section) of how to link personal data to choices (e.g., `DataRepAlias.UserId = PrefRepAlias.PrefId`)

A.2. Metadata

The Metadata section of a parametric obligation policy describes: (1) Type of obligation policy (e.g., "Parametric"); (2) Natural language description of the obligation, presented to users and/or administrators. The XML skeleton of the Metadata section follows:

```
<metadata>
  <type>Parametric</type>
  <description> ... </description>
</metadata>
```

A.3. Events

The *Events* section of a parametric obligation policy describes "parametric" events that must occur to trigger the obligation. These events can contain references to personal data and choices described in the *Target* section. The high level XML skeleton of the *Events* section follows:

```
<events operator="AND/OR/NOT ">
  <event id="e1">
    <type> ...</type>
  </event>
</events>
```

One or more *event* or *events* sub-sections can be described in this section, in a recursive way, combined via logical AND/OR/NOT operators. Each "event" subsection has a unique, local identifier. The actual definition of these events depends on their types. Currently managed types of events are:

- **Time based event:** it describes a condition that checks the current time (NOW) against a stated time. The "stated time" can be retrieved via a reference (e.g., to a field in a Privacy Choices data repository) ;
- **Data Access event:** it describes a condition on how many times a specified user's data item(s) can be accessed in a predefined period of time. The actual information (data subject's personal data item, number of accesses and period of time) can be retrieved via references to values stored somewhere;

- **Data Deletion event:** it describes a condition that is true when a specified piece of data has been deleted (by an external system). The location of this data can be specified via a reference.
- **Context-based event:** it describes conditions on contextual information (e.g., system attributes, OS or application-based information). References to this information can be used.

In our example of a privacy obligation policy, a simple time-based event is described as follows:

```
<events operator=" ">
  <event id="e1">
    <type>TIMEOUT</type>
    <date">
      NOW > [#ref] PrefRepAlias.TimeChoice
    </date>
  </event>
</events>
```

In our example, the “*NOW > [#ref] PrefRepAlias.TimeChoice*” condition is verified if the current time (NOW) is greater than a time accessible via the “[#ref] PrefRepAlias.TimeChoice” reference. This reference points to information stored in the Privacy Choices repository (having the PrefRepAlias alias) in the “TimeChoice” field, as declared in the Target (see the Target example in section A.1). It is important to notice that, in our example, each piece of data has an associated choice value - specified by the user and stored in the Choice Repository (“TimeChoice” field).

At this declarative stage, this *reference* is a generic *reference* to potentially many values stored in the Choice Repository. It must be *contextualized* to each specific “piece of data” the policy applies to. This happens at runtime, during the interpretation of *events*. The Obligation Management System will achieve this by using the Target section of this policy: for each targeted piece of data it will retrieve the associated Choices based on the specified reference (e.g., “TimeChoice” value in the Choice Repository) and check any related condition in the events section (in our simple example it is a simple time-based condition). This is done in an efficient way, via a few SQL queries to databases. In our example, when the time-based condition is satisfied for a given piece of data and an associated choice, the system triggers the enforcement of related actions on that piece of data.

A.4. Actions

The *Actions* section of a parametric obligation policy describes “parametric” actions to be enforced when an obligation is triggered by its events. These actions can contain references to data and choices consistently with the definitions in the Target section. A high level XML skeleton of the *Actions* section follows:

```

<actions>
  <action id="a1">
    <type>.....</type>
    <onCondition> ... </onCondition>
    ...
  </action>
</actions>

```

One or more *action* sub-sections might be defined in this section. Each action sub-section has a unique, local identifier. Actions are executed in a sequence, potentially subject to the satisfaction of (optional) conditions (e.g., constraints on privacy choices. By default these conditions are TRUE, i.e., actions are just executed). The actual definition of these actions depends on their types. Currently managed types of actions are:

- **Notification Action:** this action sends a notification to an entity. The e-mail address of this entity can actually be a reference to a value in the Data Repository;
- **Deletion Action:** this action deletes a piece of personal data or some of its attributes. A reference can be used to identify this piece of data;
- **Command Execution Action:** this action executes an external application or service (e.g., a workflow application to process a piece of data or transform it). References to personal data or privacy choices can be passed as parameters;
- **Logging Action:** this actions logs information (including referenced information) for auditing purposes.

In our example of privacy obligation policy, two actions are defined, to delete the data subject’s credit card details and notify him/her:

```

<actions>
  <action id="a1">
    <type>DELETE</type>
    <data attr="part">
      <item>
        [#ref] DataRepAlias.CreditCardRef
      </item>
    </data>
  </action>

```

```

<item>
  [#ref]DataRepAlias.CreditCardNumber
</item>
</data>
</action>
<action id="a2">
  <type>NOTIFY</type>
  <method>EMAIL</method>
  <to> [#ref] DataRepAlias.Email </to>
  <text> some e-mail text here </text>
</action>
</actions>

```

These actions contain references to personal data (credit card details and e-mail address). The same observations made in the “Events” section apply here. These references are “solved” at runtime, based on contextual information i.e. specific pieces of personal data for which obligations have been triggered.

A.5. *On Violation Actions*

The “*On Violation*” section of a parametric obligation policy describes “parametric” actions to be executed in case an enforced policy is violated, i.e., if any of its enforced actions fail. The XML skeleton follows:

```

<onViolation>
  <ovAction id="oval">
    <type> .....</type>
    <onCondition> ... </onCondition>
    ...
  </ovAction>
</onViolation>

```

An action can fail either at the enforcement time or afterwards (e.g., deleted data could reappear because of wrong database synchronisation): this latter case is detected by the monitoring component of our obligation management system. All actions described in the “Actions” section can be used in the “OnViolation” section. A specific “RE-ENFORCE” action has been introduced just for the “OnViolation” section: when used, it requires the system to re-enforce just the actions that have failed (in the Actions section).