Requirements for privacy-enhancing Service-oriented architectures

Editors:	Sebastian Meissner (ULD)
	Jan Schallaböck (ULD)
Reviewers:	Carine Bournez, (W3C)
	Claudio Ardagna, (UniMi)

Identifier:	H6.3.1
Type:	Heartbeat
Class:	Public
Date:	February 27, 2009

Abstract

Service-oriented architectures expose new chances and challenges for privacy and data protection. The potentially increased distribution of personal information across multiple domains make subject access requests difficult to handle. Which service did process what data? Whom to address for liability issues? At the same time, the service orientation offers a new approach for the granularity of data processing, allowing clearer responsibilities and better auditing.

This deliverable develops a comprehensive set of requirements for Service-oriented architectures. If applied in the construction of Service-oriented architectures, legal compliance with privacy legislation should be facilitated. Even more, they offer additional support for privacy enhancing Service-oriented architectures.

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 216483.

Members of the PrimeLife Consortium

1.	IBM Research GmbH	IBM	Switzerland
2.	Unabhängiges Landeszentrum für Datenschutz	ULD	Germany
3.	Technische Universität Dresden	TUD	Germany
4.	Karlstads Universitet	KAU	Sweden
5.	Università degli Studi di Milano	UNIMI	Italy
6.	Johann Wolfgang Goethe – Universität Frankfurt am Main	GUF	Germany
7.	Stichting Katholieke Universiteit Brabant	TILT	Netherlands
8.	GEIE ERCIM	W3C	France
9.	Katholieke Universiteit Leuven	K.U.Leuven	Belgium
10.	Università degli Studi di Bergamo	UNIBG	Italy
11.	Giesecke & Devrient GmbH	GD	Germany
12.	Center for Usability Research & Engineering	CURE	Austria
13.	Europäisches Microsoft Innovations Center GmbH	EMIC	Germany
14.	SAP AG	SAP	Germany
15.	Brown University	UBR	USA

List of Contributors

This deliverable has been jointly authored by multiple PrimeLife partner organisations. The following list presents the contributors for the individual parts of this deliverable.

Chapter	Author(s)
Executive Summary	Sebastian Meissner, Jan Schallaböck (both ULD)
Introduction	Sebastian Meissner (ULD)
Legal Framework	Sebastian Meissner, Jan Schallaböck (both ULD)
Technical Framework	Laurent Bussard, Uli Pinsdorf (both EMIC), Stuart Short (SAP)
Requirements	Sebastian Meissner, Jan Schallaböck (both ULD)
Scenario	Uli Pinsdorf (EMIC), Sascha Koschinat (GUF)
Outlook	Jan Schallaböck (ULD)

Executive Summary

This requirement document addresses the privacy risks and chances resulting from the implementation of Service-oriented architectures (SOA) within one organisation or rather across different organisations ("cross-domain service composition"). In particular cross-domain service composition involves new privacy risks. The involvement of different legal entities may lead to the situation that data subjects are no longer aware of what data relating to them are handled by what entity for what purpose and the use of standardised formats and interfaces across different security domains facilitates linkage of data sets and thus profiling of data subjects.

On the other hand, SOAs also provide some options to achieve an enhanced level of privacy. First, one typical attribute of an SOA is that each single service can be mapped on to specific purposes. This circumstance facilitates the implementation of an automated review of adherence to the privacy principle of purpose limitation. Second, the tailoring of single services to specific purposes also simplifies determination of personal data that are really needed for the respective service and thus eases adherence to the privacy principles of collection, use, and disclosure limitation as well as obedience to the principle of data minimization. Third, as the technical integration of a SOA nowadays typically is taking place on the basis of web services and XML, it provides some possibilities for the implementation of an automated data protection management.

The document at hand aims at listing relevant requirements that facilitate the design of privacy-enhancing Service-oriented architectures.

The structure of the introduced set of requirements relates to the different functions of services (or rather: their activities) as depicted in the technical framework chapter. Thus a threefolded structure comes into play, roughly mapping to the three states, before (policy), during (logging), and after processing (access to primary information). For each of these structural elements, the same types of requirements reoccur: the need for formalisation, non-repudiation, and accessibility, and the answer to the questions: "Who processed the data?", "What data is processed?", "How is the data processed?" and "To whom is the data transferred?"

The document does not amount to nothing more than listing the requirements. Rather, it offers possible technical solutions to meet each of the requirements. In particular, it proposes to make use of the following techniques and instruments:

- Policy matching and aggregation (enabled by development of a formal language)
- Sticky policies (policies of service providers are attached to the personal data)
- Sticky logging (log data are attached to the personal data)
- Privacy4DRM (DRM mechanisms are used to enforce adherence to privacy policies)
- Certifications (e.g., privacy seals could be used as a trust anchor)
- Anonymity measurement techniques

All of these techniques and instruments are briefly described as possible solutions in the requirement section of this document. With regard to most of them, quite some additional research has to be done before they will be mature enough to be implemented in existing SOAs.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

SOA is a technology-independent architecture concept adhering to the principle of service-orientation. It aims at enabling the development and usage of applications that are built by combining autonomous, interoperable, discoverable, and potentially reusable services. These services jointly fulfil a higher-level operation through communication. One core principle of SOA is the loose coupling of partial services: Single services are not permanently bound to each other, but their binding happens at run-time enabling a dynamic composition of services. Moreover, it is even feasible to dynamically bind services hosted in different security domains and by different legal entities ("cross-domain service composition"). One prominent example for this is the "service chain" that comprises of several partial services offered by different organisations. To facilitate the use of such services, usually one legal entity might serve as single point of contact for (potential) users. In times of the Internet, places of business of organisations providing partial services for one high-level service can be widely distributed around the globe.

In many cases, an SOA might involve the processing of personal data and thus raise risks for the privacy of data subjects concerned. Two specific risks can be identified with regard to cross-domain service composition. The first of them concerns transparency of processing of personal data: The involvement of different legal entities may lead to the situation that data subjects are no longer aware of what data relating to them are handled by what entity for what purpose. This is particularly true if services in fact are bound dynamically at run-time. In case a high-level service is performed by different organisations, but offered only by one of them, customers even might not be aware of the involvement of further legal entities at all. The second risk concerns the issue of linkability of data: The use of standardised formats and interfaces within an SOA facilitates linkage of systems and data sets. Without the implementation of appropriate technical and organisational measures, organisations could be able to link different sets of personal data and generate profiles on data subjects.

However, the implementation of an SOA also provides some options to achieve a high level of privacy for the data subjects concerned. First, one can realize that each single service that forms part of an SOA usually serves a specific purpose (e.g., authentication, payment). In combination with privacy-compliant logging techniques, this circumstance can be used to implement an automated review of adherence to the privacy principle of purpose limitation (for thorough description of legal principles see Chapter 2). Second, tailoring of single services to specific purposes simplifies determination of personal data that are really needed for the implementation of the respective service. This circumstance facilitates adherence to the privacy principles of

collection, use, and disclosure limitation as well as obedience to the principle of data minimization. Third, SOA provides some possibilities for the implementation of an automated data protection management. This results from the fact that technical integration of a SOA nowadays typically is taking place on the basis of web services and XML. As the same holds true for existing and emerging standards for an automated data protection management (e.g., P3P, EPAL, and WS-Privacy), these standards could easily be applied within an SOA.

A general goal of Work Package 6.3 is the identification of requirements for privacy-compliant or rather privacy-enhancing Service-oriented architectures. At this, current research is to be realigned to a European level, taking into account the different legal frameworks. Moreover, WP6.3 aims at going further towards a better understanding of how to architect a SOA whilst keeping privacy in mind. It is closely connected with Activity 5 and WP3.4: A5 will provide a language to express data handling policies and preferences. Relationships between A5 and WP6.3 are in place to ensure that the resulting data handling policy will be usable within Activity 6. WP6.3 will use the data handling language provided by A5 and build tools to use it during modelling and run-time of cross-domain service composition. WP5.2 will provide for better understanding of privacy implications of Service-oriented architectures and empirical evaluation of current SOAs. Finally, WP3.4 will take the result of WP6.3 into standardisation.

This document aims at identifying fundamental requirements for privacy-enhancing or (at least) privacy-compliant Service-oriented architectures. It is organized as follows: Chapter 2 introduces the legal framework for the design and implementation of privacy-compliant SOAs. In particular, it presents core principles of EU data protection law that are to be abided by developers and operators of Service-oriented architectures. Chapter 3 provides an overview on the technical framework for privacy-aware service composition. After describing the state of the art, it identifies static and dynamic software analysis as the core of privacy-aware service composition and introduces a holistic approach on such a service composition covering views on structure, activities, and patterns. Chapter 4 lists the requirements for privacy-compliant or rather – enhancing SOAs that have been identified on the basis of the two previous chapters. It is divided into five chapters dealing with requirements for (1) policies, (2) logging, (3) access, (4) cross-domain service composition, and (5) additional mechanisms. Chapter 5 presents a scenario dealing with an online career portal allowing for creation and use of an electronic CV as well as for the use of attached claims issued by different legal entities. This scenario exemplifies aspects of requirements listed in the previous chapter. Finally, Chapter 6 provides an outlook on future research about privacy-enhancing service composition.

Chapter 2

Legal Framework

The protection of personal data in the sense of a right to informational self-determination is EU-widely recognized as a fundamental right. It is explicitly acknowledged by many constitutions in the European Union and is similarly one core element of the right to respect for private and family life guaranteed by Article 8 of the European Convention of Human Rights. It is also acknowledged by the jurisdiction of the European Court of Justice and specifically addressed in Article 8 of the European Union's Charter of Fundamental Rights. This charter will be legally binding as soon as the Treaty of Lisbon will come into force.

The central regulatory instruments within the EU for data protection issues are the

- Directive 95/46/EC on the protection of individuals with regard to the processing of personal data and on the free movement of such data (General *Data Protection Directive*)[1] and the
- Directive 2002/58/EC on privacy and electronic communications (*E-Privacy Directive*)[2].

The Directives provide fundamental legal principles for this area of regulation that will be introduced in higher detail in the following (1.-11.). As a legal instrument European Directives are addressed towards the member states for national implementation. Therefore 95/46/EC and 2002/58/EC are not directly applicable, but had to be implemented in national laws by the EU Member States, which all member states did.

In this document, only the above-mentioned EU Directives are taken into account. National implementation is not considered, but needs to remain within the regulation prescribed by the directives. Thus, an equal level of protection is ensured and variations – at least on the level of principles, as laid out here – are limited.

Applicability however of the abovementioned regulation is only given if *personal data* are processed by a *data controller* or a *data processor*.

Article 2 lit. (a) of Directive 95/46/EC defines personal data as

> "any information relating to an identified or identifiable natural person (*data subject*)".

The term *identifiable* natural person is defined in the same provision:

> "An identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his

> physical, physiological, mental, economic, cultural or social identity" (see also Recital 26 of Directive 95/46/EC).

A thorough definition of the term ‘personal data’ is provided by the Article 29 Working Party in its Opinion 4/2007 on the concept of personal data [3].

Subject to the legal obligations stipulated by EU data protection law is the *controller* of the processing. According to Article 2 lit. (d) of Directive 95/46/EC the controller is

> “the natural or legal person, public authority, agency or any other body which alone or jointly with others determines the purposes and means of the processing of personal data.”

However, one has to be aware of the fact that the processing of personal data by a natural person in the course of a *purely personal or household activity* is out of scope of the Data Protection Directives (cf. Article 3 paragraph (2) of Directive 95/46/EC).

The controller has to be distinguished from the processor who processes personal data on behalf of the controller (cf. Article 2 lit. (e) and Article 17 paragraphs (2)-(4) of Directive 95/46/EC). Even if the processor handles personal data, the controller remains liable for this processing.

Directives 95/46/EC and 2002/58/EC follow a set of well established legal principles introduced below.

2.1 Lawfulness

Article 6 paragraph (1) lit. (a) of the Data Protection Directive stipulates that personal data must be processed fairly and lawfully. According to European law, lawfulness requires the existence of a legal basis for each step of processing of personal data (cf. Article 7 of Directive 95/46/EC). Handling of personal data without a legal basis is illegal and may result in penalties and further sanctions (see Article 24 of the Data Protection Directive). A legal basis for the processing can be either provided by law directly (e.g., Article 7 lit. (b)-(f) of Directive 95/46/EC) or by the consent of the data subject.

According to Article 7 lit. (a) of the Data Protection Directive, precondition of a valid consent is that it is given in an unambiguous manner. This usually requires a conscious act of some kind by the data subject, such as actively ticking a box. This so-called “opt-in consent” is of particular importance in the online world. Consent must be sufficiently specific and freely given (cf. Article 2 lit. (h) of Directive 95/46/EC). Furthermore, the controller has to inform the data subject about all relevant aspects of the processing so that he or she is able to fully understand all of its implications ("*informed consent*"). The information has to be provided before the consent is given. As a consequence to it being given voluntarily, the consent may also be withdrawn at any later date or time.

Implementers of SOAs must therefore thoroughly check whether there is a legal basis for each processing of personal data. Apart from the consent of the data subject, mainly Article 7 lit. (b) and (f) of the Data Protection Directive might come into consideration to provide a legal basis for the processing. According to the former provision, processing is permitted if it is necessary for the performance of a contract to which the data subject is party. The latter provision provides for legitimacy of the processing if it is necessary for the purposes of legitimate interests pursued by the controller, except where such interests are overridden by the interests of the data subject.

In many cases it might be quite a complex issue to check the legitimacy of each processing step taking place within an SOA. This specifically applies to SOAs that are designed for virtual organisations consisting of several legally independent parties. In case of such an SOA, the parties have to identify in a first step, which of them are controllers and which are processors in the sense of Article 2 lit. (d) and (e) of the Data Protection Directive. In a second step, each controller has to

identify all processing steps that fall within his or her responsibility and to check whether there is a legal basis for each of them.

2.2 Purpose Specification

Lawfulness of processing of personal data does not only require a legal basis for the processing, but also compliance with all principles of European data protection law. Fundamental to this is the concept of purpose specification. Article 6 paragraph (1) lit. (b) of the Data Protection Directive stipulates that personal *data must be collected for specified, explicit and legitimate purposes*. This especially requires the controller of the processing to determine the purposes of the processing prior to the collection of data (also cf. Recital 28 of Directive 95/46/EC). Furthermore, the purposes have to be communicated to the data subjects before or at the time the data are collected (cf. Article 10 lit. (b) of Data Protection Directive; see also Article 11 lit. (b) thereof). The principle of purpose specification is closely related to the concept of purpose limitation (see below at 4.).

Designing an SOA, the purpose of each processing step of personal data must be clearly determined. The controller of the processing is only in line with the principle of purpose specification if he or she determines sufficiently specific purposes. It is not enough to stipulate a general purpose such as "for the purpose of customer relationship management". Rather, purposes must be defined in a manner that does not allow any doubt about what is meant by them.

2.3 Collection Limitation

Another fundamental principle of European data protection law is the concept of collection limitation. Article 6 lit. (c) of the Data Protection Directive provides that *personal data must be adequate, relevant and not excessive* in relation to the purposes for which they are collected and/or further processed. From this follows that the collection of personal data must be limited to data that are really necessary for the specified purposes. The principle of collection limitation is closely related to the concept of data minimization (see below at 5.).

Implementers of an SOA must thoroughly check whether all personal data they intend to collect are really needed to realize the previously defined purposes. That implies the careful examination of the necessity of every single data type that constitutes or can be linked to other personal data. It is not sufficient if data are just useful to achieve the specified purposes. In fact, data are only necessary if the intended purpose could not be realized without them.

2.4 Use, Retention and Disclosure Limitation

Consequently not only the collection of data, but also the use, retention and disclosure of data have to be limited according to European data protection law. Pursuant to Article 6 lit. (b) of the Data Protection Directive personal data

“must not be further processed in a way incompatible with the purposes originally specified.”

The described concept is known as the *principle of purpose limitation* and is closely related to the principle of purpose specification that was already introduced above.

Article 6 lit. (e) of Directive 95/46/EC also provides that personal data must be kept in a form that permits identification of data subjects for no longer than is necessary for the purposes for which the data were collected or for which they are further processed. Put simply, *data have to be erased* or at least truly anonymized *as soon as they are not longer needed* for the achievement of the pursued purposes. This principle is closely related to the principle of data minimization.

Occasionally, the controller is obliged to keep personal data despite expiration of the pursued purpose (e.g., in case of a legally stipulated retention period). In this instance, the respective data have to be blocked, i.e. be marked in a manner that assures their exemption from further use.

As it holds true for every other processing of personal data, the disclosure of personal data to third parties is only legitimate if there is a legal basis for it (cf. Article 7 of Directive 95/46/EC in conjunction with Article 2 lit. (b) thereof). Beyond that, the internal disclosure of personal data within the entity of the controller has to be limited as well. It has to be ensured that data are only disclosed internally to people who really *need to know* them in order to achieve the pursued purpose(s) of the processing. Article 17 paragraph (1) stipulates that the controller has to implement appropriate technical and organisational measures to protect personal data against unauthorized disclosure.

When designing an SOA, it has to be kept in mind that collected data may not be used for purposes incompatible with the original ones and that data have to be erased immediately after cessation of the pursued purposes. Furthermore, implementers of an SOA have to be aware of the fact that data may only be disclosed externally if there is a legal basis for it. The latter aspect is of particular importance if one deals with an SOA that is designed for virtual organisations consisting of several controllers in the sense of Article 2 lit. (d) of the Data Protection Directive.

2.5 Data Minimisation

Article 6 paragraph (1) lit. (b) of the Data Protection Directive provides that personal data must only be collected for legitimate purposes, while lit. (c) thereof states that data must be adequate, relevant and not excessive in relation to the purposes for which they are collected and/or further processed. In addition, Article 6 paragraph (1) lit. (e) provides that personal data must not be stored longer than necessary for the purposes, for which the data is collected or for which it is further processed. So implementers of SOAs must have identified which data is really needed for the service, and must not collect any data, that is not needed. Equally any SOA needs to include mechanisms to delete data, which is not needed.

But data minimization for a privacy-enhanced SOA can go even further in some jurisdictions. For example in Paragraph 3 of the German data protection act, the goal of data minimization and data avoidance is described in more detail with the current draft ISO Privacy Framework [ISO 20084] following a similar approach: Data avoidance addresses the idea to design software in such a way that as little personal information as possible is needed.

This goal may be supported by all layers of the architecture. This may be achieved via Privacy Enhancing Technologies (PET) -mechanisms, such as pseudonymous credentials but also by simply avoiding the unnecessary creation of temporary shadow files, unnecessary logging or by filtering out data that is not needed.

2.6 Accuracy and Quality

Data protection legislation also requires mechanisms to ensure accuracy and quality of personal data (cf. Article 6 paragraph (1) lit. (d) of Directive 95/46/EC). A data controller has to ensure, that those data he or she is processing is duly ensured for its accuracy, one reason for this being that inaccurate data may lead to false conclusions. This includes the obligation to undertake "every reasonable step" to keep data up to date, and that inaccurate data needs to be erased or rectified.

To support such accuracy, as well the current draft ISO Privacy Framework, give preference to collecting personal data directly from the data subject, wherever that is reasonable. Some experts in ISO even recommend to discard those data, whose accuracy cannot be assured, and to

implement mechanisms to frequently check the accuracy and quality of collected personally identifiable information (PII).

2.7 Openness, Transparency and Notice

Openness and transparency are core principles of data protection. The law provides a number of mechanisms to support transparency (a) before the data is collected and (b) thereafter.

(a) When collecting personal data, the data controller has to provide information on

- his or her identity,
- the purpose of the collection,
- the specific data collected, and
- the identities or types of recipients of the personal data.

In those cases where the data is not collected directly from the data subject, the data controller is obliged to inform the user of such collection and its circumstances as above (Article 11of the Data Protection Directive). Any such information must be provided in a comprehensible manner; otherwise the legal basis for the collection may be questionable, possibly leading to unlawful collection of personal data. This information may be abstracted into layers. This is especially acceptable, when following the specific recommendations of the Article 29 Working Party in this regard[5].

(b) For the once collected data, the data subject has the right to access any data on him or her. This, again, has to include the list of information regarding purpose, types of data and, and - since the data has already been processed - also the identity of recipients that the data already has been forwarded to (Article 12 of Directive 95/46/EC).

2.8 Individual Participation and Access

Beyond simple access as part of the transparency mentioned above, the data subject has a right to rectification of incorrect information, and in some cases to deletion and blocking. The general right to access is governed by Article 12 of the Data Protection Directive with lit. (b) referring to the *right to rectification, blocking and erasure*. Erasure can always be an option, where the data controller has never had, or has lost a legal basis for the storage of personal data. This can especially be the case, if the data subject withdraws his or her consent, and there is no other legal basis (e.g., contractual reasons) for further collection of the data. Blocking of personal data (Article 14 of Directive 95/46/EC) applies especially in those cases, where the data needs to be stored for legal purposes (e.g., tax validation), but the user has chosen not to allow usage for any other purposes, such as advertising purposes. Finally lit. (c) of the above mentioned article mandates the data controller to inform all other parties, which he or she has previously passed the information on to, of the erasure, blocking or correction.

2.9 Accountability

Any data controller must include mechanisms to ensure the accountability for the data he or she processes. This especially includes security safeguards within its own organisation (see below 10.), but also goes beyond his or her organisation when passing the data on to entities processing the data on his or her behalf. In those cases the data controller remains responsible for a lawful and diligent handling of the personal data, which he or she must ensure through effective contractual mechanisms and other means (see also Article 17 paragraphs (2) and (3)).

2.10 Security Safeguards

A data controller has to implement appropriate technical and organisational protection mechanisms, to

"protect personal data against accidental or unlawful destruction or accidental loss, alteration, unauthorized disclosure or access, in particular where the processing involves the transmission of data over a network, and against all other unlawful forms of processing." (cf. Article 17 of Directive 95/46/EC)

This means that state of the art technology has to be taken into account, where reasonable established standards have to be applied.

2.11 Further Compliance Issues

2.11.1 Special Categories of Data

The European legislator identified a special need for protection of some types of data that are listed in Article 8 of the Data Protection Directive. In particular, the Directive foresees a higher level of protection for all data types that are listed in Article 8 lit. (1): Personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, trade-union membership, and the processing of data concerning health or sex life. These so-called *sensitive data* may only be processed under the preconditions of Article 8 paragraphs (2)-(4) of Directive 95/46/EC. Paragraphs 5 and 7 of this provision introduce further special categories of personal data such as data relating to offences or criminal convictions and national identification numbers.

2.11.2 Automated individual decisions

The EU legislator also identified special risks for data subjects that can result from automated individual decisions. Thus, Article 15 paragraph (1) of the Data Protection Directive grants the right to every person not to be subject to a decision which produces legal effects concerning him or her or significantly effecting him or her, and which is based solely on automated processing of data intended to evaluate certain personal aspects relating to him or her, such as his or her performance at work, credit-worthiness, reliability or conduct. Article 15 paragraph (2) of the Data Protection Directive provides two exceptions from that general rule assuming that the legitimate interests of the data subject are sufficiently safeguarded if all of the respective prerequisites are met.

2.11.3 Transfer to Countries outside of the EC

Obviously, in the age of the internet personal data can easily be transferred across borders. As long as data remain inside the territory of the European Union the level of protection guaranteed by national laws does not vary significantly. This results from the fact that the national laws of the Member States were largely harmonized by the implementation of Directives 95/46/EC and 2002/58/EC. The situation is different if data are transferred to countries that are neither a member of the EU nor of the European Economic Area (so-called *third countries*). In case personal data are transferred to a controller situated in a third country, legitimacy of this transfer does not only require a legal basis (as every processing of personal data). Rather, the special requirements of Articles 25 f. of the Data Protection Directive must be met as well.

Chapter 3

Technical Framework

Service-oriented architecture (SOA) has become an important concept when talking about service composition today. It is defined as a form of technology architecture that adheres to the principles of service-orientation [6]. It defines a set of guiding principles to enable the development and usage of applications that are built by combining services. Those services should be autonomous, interoperable, discoverable, potentially reusable, and vendor-neutral. The interactions between these services are formally defined through contracts that are independent of the underlying platform and from the programming language used. SOA establishes the potential to support and promote these principles throughout the business process and automation domains of an enterprise.

This section describes use cases related to service composition where security and/or privacy are important aspects.

3.1 State of the art

Service Oriented Architecture (SOA) is a collection of cooperating services, which jointly fulfil a higher-level operation through communication. They fall in the class of distributed systems [7]. A special attribute of SOA is the loose binding between the services. Typically the binding happens only at run-time, which means that a service learns only at this point in time with which actual service instance it is communicating. This feature is called loose binding and is in fact said to be one of the core characteristics of SOA [8].

In PrimeLife, we define "*cross-domain service composition*" as any combination of existing services and resources hosted in different security domains. This composition can be an application (e.g. Java, C#), a workflow (e.g. Microsoft Work Flow [9], Business Process Execution Language (BPEL) [10]), or a mash-up (e.g. openkapow [11], Microsoft Popfly [12]). Exemplary use cases for cross-domain service composition could, for example, be: Very formal composition (e.g. virtual organisations, hospital workflow), data mash-up created with a visual editor, or more ad-hoc collaborations between multiple parties.

3.1.1 Privacy in existing workflow technologies

A workflow is the movement of documents, information or tasks through a business process and, the technologies used to perform this, allow these processes to be defined, executed, and automated according to a set of procedural rules. In the context of a distributed heterogeneous environment the workflow could contain a set of multiple tasks that would need to be executed in a coordinated manner.

There are three different types of workflow, namely, sequential, state machine and rules-driven workflows. Sequential workflow is a progression from one step to the next and usually there is no return to a previous step; an example of this can be seen in a flow chart. On the other hand, the state machine workflow which is a progression from state to state, allows, if needed, a return to a previous stage. The rules-driven approach is based on a sequential workflow and the progression is dictated by rules. A rule-based approach can ensure that privacy concerns are taken into consideration and workflow technologies permit systems and processes to be rapidly modified to reflect changes that may occur. Furthermore, through Business Process Modeling and Analysis, the potential impact on these changes can be assessed.

Typically a developer builds a service composition by connecting different processes through common workflow logistics. Another name for this process is Orchestration, since the individual services are directed by a central instance, the application that is based on the workflow logistics, just like a director coordinates musicians of an orchestra. An orchestration expresses business process logic that is typically owned and controlled by a single organisation, even if that logic involves interaction with external business partners. An orchestration establishes a business process protocol that formally defines a business process structure. The internal workflow logic is broken down into a series of basic and structured activities that can be organized into sequences and flows. It is important to notice that the workflow acts as a meta-structure which orchestrates the communication between different service nodes. Orchestration is an important part of SOA because it provides the means of expressing business process logic in a standardised and service-oriented way. WS-BPEL [7113], the Web Service Business Process Execution Language is an industry specification that standardizes orchestration.

Another way to create a service composition is to let the individual services organize themselves, so that they act and react with each other. This approach is called choreography. Choreography aims at organizing multiple applications within an organisation and even information exchange between multiple organisations. The goal is to establish a kind of collaboration between services representing different entities (organisations). The participants in choreography act in different roles and have different relationships. Typically, there is no single owner of the collaboration logic. Once the collaboration protocol with the message exchanges has been defined, the choreography is self-organizing. WS-CDL [UPin-M], the Web Service Choreography Description Language, is one of several specifications that deal with choreography.

The big difference between orchestration and choreography is the way the control of data flow is done. While orchestrations are controlled by a central logic that controls how the data flows, the control within choreography is more decentralised and is governed by message exchange patterns. There are certainly overlaps between orchestration and choreography where choreography could be used to connect multiple orchestrations etc.

3.1.2 Privacy in existing mash-up technologies

Mash-ups can be seen as a web-based resource that combines existing resources, whether they contain content, data or application functionality, with more than one resource in different environments by empowering the end-users to create and adapt individual information centric and situational applications. The aggregation and linking of content from different resources can be

achieved in an intuitive manner and usually it is not required to have knowledge of programming languages.

As a web application that combines data from more than one source into a single integrated application, mash-ups make use of APIs published by service providers such as Yahoo!, Google, Amazon, eBay, Microsoft, and others. Those APIs are based on web standards and allow the use of the functionality of the services. The actual invocation of the APIs is quite different; it ranges from programming approaches where programmers consciously take advantage from all details of a given API to more “graphical” invocations which allow service composition without being an expert in programming. For instance, often mash-ups are created using a graphic designer where services can be dragged and dropped as, for example, boxes and connected to each other. Further user input and data can be provided to the mash-up and be consumed/distributed further by the mash-up. Hence, mash-ups can also use other mash-ups as data source. Depending on the technology, mash-ups can be executed either in the browser, on the client side, or on the server side.

Three types of mash-ups can be identified:

- Presentation / Consumer mash-ups bring information from more than one source into a common user interface, such as web portals (e.g. Live.com [14], iGoogle [15], My Yahoo! [16]) displaying the information side by side. Little real integration is involved. This is normally used for private use and permits users to easily combine data elements from multiple sources through a user-friendly graphical interface. Mechanisms such as drag and drop of pre-built widgets or RSS feeds are used.
- Data mash-ups extract data from multiple sources and combine it. They enable cross-referencing and comparison of data, e.g. mix of geographical data with Wi-Fi hotspot locations, house prices, or crime statistics. Extraction of these data might be hard and require programming. Visual development environments exist to build data mash-ups (Yahoo! Pipes [17], Popfly, Dapper [18], openkapow, Serena Software [19], etc.).
- Logic/business mash-ups combine data, people, and processes. They connect two or more applications and automate certain tasks. They always involve programming. Within the enterprise, they overlap with traditional workflow applications and with composite applications, but they should enable rapid customization and adoption (Serena Software).

Mash-ups like the latter show more overlap with SOA than the former two categories where client-side and server-side mash-ups compete with server-side orchestration technologies such as BPEL. As there is a focus on existing back-end systems there is a need for security, quality or availability.

The concept of mash-ups is advancing quickly. So are the ways to use and mix different types of mash-ups. Therefore, the description above gives just an indication of what is being done at the moment. However, further combinations and variations among them should clearly be expected.

Privacy can be a concern as usually mash-ups do not stipulate rules on data usage; in the case of a consumer mash-up it is possible, with certain existing technologies, to easily derive personal identifiable information. This was demonstrated by computer consultant Tom Owad who mashed book wishlists posted by Amazon users with Google Maps. The wishlists often contain the user's full name, as well as the city and state in which they live -enough information to find their full street address from a search site such as Yahoo People Search; this was sufficient to get a satellite image of their home from Google Maps. Owad used this to produce a map of people who liked to read "subversive" books (www.applefritter.com/bannedbooks), showing what level of detail the sites can provide.

3.2 Privacy-aware service composition

The goal of this document is to describe requirements for technology that helps to deal with constraints-based service compositions both at design-time (i.e. during creation of a workflow) and at run-time. At design-time the technology usually supports a developer creating a workflow that considers requirements and warns her if the current workflow structure implies a violation of the constraints. At run-time the technology shall help to enforce the constraints based on the actual service invocation and exchanged data. The PrimeLife deliverable D6.2.1 [20] gives in Sect. 3.3 an overview of service composition with constraints and a list of such constraints related to security and privacy. We will focus here on data handling constraints.

The remaining of this section will mainly focus on data handling constraints:. Activity 5 will provide a language to express data handling policies and preferences. Relationships between A5 and WP6.3 are in place to ensure that the resulting data handling policy will be usable within WP6.3 scenarios. WP6.3 will use the data handling language provided by A5 and built tools to use it during modeling and run-time of cross-domain composite services.

3.2.1 Target groups

Target Group	Benefit from privacy-aware service composition
Workflow designer, programmers	▪ Create privacy-aware services ▪ Avoid common mistakes happening just due to unawareness of programmers
Operator hosting a single service	▪ Enable to make meaningful privacy statements about the service ▪ Privacy-awareness for services becomes an incentive or even a business ▪ Raise users' trust in service
Operator hosting a service composition	▪ Enable to make meaningful privacy statements on service composition ▪ Know in advance the impact of third-party services ▪ Make complex SC transparent and auditable ▪ Raise users' trust in service
User / Data Subject	▪ Know in advance how/where data will be processed or stored ▪ Enable user to choose between 'cheap&evil' and 'expensive&trustworthy' ▪ Raise users' awareness for privacy
Auditor	▪ Easy way to verify that an operator met its promises on privacy ▪ Generate non-repudiation for privacy logs

Table 1: Mapping core Requirements to functions

The solution should address at least five target groups. First there are the workflows designers who are in charge to create a workflow. They want to avoid design mistakes that eventually lead to privacy breaches. The operators of individual services want to make meaningful, measurable privacy statements about their services, e.g. for profiling services. Same is true for the operator of the whole service composition; if they want to know how the actually invoked individual services sum up in terms of privacy. Having measurability makes it even a business case for both types of operators to invest into privacy compliant services. Certainly, the user resp. the data subject takes advantage from a privacy-aware service composition. She knows in advance how PII will be processed and what types of obligations can be promised by the operator of the service composition. Finally, an auditor would have an easier life; she could digest massive log files for all operations invoked on a specific PII or could enable the compliancy of a service in general. The following table summarizes the target groups and the benefits it could draw out of a technology for constraint-aware service composition.

3.2.2 Static analysis vs. dynamic analysis

In order to make a service composition privacy-aware we have to automatically analyze it and display the results in an appropriate way to the user. Hence software analysis is the core of privacy- aware (or even constraint-aware) service composition. In general the software analysis is split in two independent disciplines, which are

- Static code analysis and
- Dynamic code analysis.

Both disciplines complement each other and have a different view on the same system. Static analysis allows an inspection of software already at design time. It generates statements about expected software architecture, data flow, data structures, or interfaces. The discipline of software measurement takes lots of metrics from static analysis. Dynamic analysis observes software during run time and allows conclusions how software responds to given inputs. Conclusions taken from dynamic software analysis are about memory consumption, performance, communication patterns, or code coverage. Dynamic software analysis has strong links to program testing.

Analysis of service compositions should be seen as a special kind of software analysis. It should focus on distributed systems and therefore emphasize the dynamic nature of service bindings. Services may bind themselves to external services only shortly before invocation. The analysis of a service composition should also be split in

- Static analysis of service compositions and
- Dynamic analysis of service compositions.

The static analysis of service composition deduces general statements based e.g. on the structure of the workflow and the type of invoked services. Static analysis is technically challenging to achieve due to relatively few data; in typically service compositions it is not known at design time which concrete service instance will be invoked. Dynamic analysis of service compositions looks at the actually invoked services, data flow patterns, orchestration resp. choreography of services.

3.2.3 Software analysis for privacy-aware service compositions

When we use static analysis for privacy-aware service composition a more or less large part of the data handling checks can be done in a static way. For instance, it is possible to check that data collected for a specific purpose cannot be used for another purpose. Similarly, it is possible to check that a workflow consuming data from a well known service (let's say medical record) with

well known data handling requirements (e.g. personal medical data can only be used for patient treatment or personal data cannot leave Europe) does not share collected data with inappropriate third parties. Note that we consider the fact of defining run-time verifications, e.g. check sticky policy before sending data, as an aspect of the static analysis. The dynamic analysis complements the static analysis in a way that it checks privacy-constraints at run-time. It is somewhat easier to achieve since everything it is done just before invocation of a service when everything is know: the concrete PII, relevant policies, service instances, service's metadata. Dynamic analysis is necessary to enforce data handling policies at run-time. It allows meeting constraints even in scenarios where services are composed on an ad-hoc basis.

Certainly, both approaches have advantages and drawbacks. Usually the static analysis lacks concrete information about the invoked services; nevertheless it can support general decisions about privacy-aspects of each exchanged data item. The dynamic analysis has lots of extra information at hand, such as the concrete data that shall be exchanged, the policy, and the concrete receiver. All these measures help to enforce privacy decisions that were made on the abstract level. Thus, privacy decision ideally come from a prior static analysis taking into account general preferences, such as that data may not be transferred outside Europe.

3.2.4 Holistic approach on privacy-aware service composition

We think that privacy-aware service composition should follow a holistic approach, taking into account multiple approaches for the static and the dynamic analysis. We found three views that should be regarded for service composition both statically and dynamically. Certainly, there may be more views; research is still going on here.

- Structure: deducts information from the general structure of a workflow including its abstractions, searching for hidden data paths, loops, bypasses, and regards communication with external entities.
- Activities: deducts information from looking at activities and groups of activities, looks at what an activity does or how activities in a group of activities relate to each other.
- Patterns: looks at security patterns, common "weak points" in service compositions such as external communication or utilized communication protocols.

It is important to understand that each of the three analysis has to be applied in the static and in the dynamic view of a service composition, in other words at design time and at run-time. That makes a total of six variations that have to be regarded. Figure 1 visualizes the six combinations. This is what we call the *holistic view*. A technology supporting all of the above mentioned target group should consider all six views to gain a maximum of information.

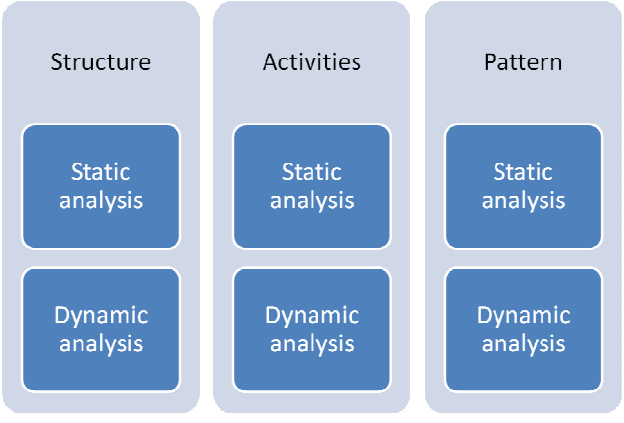

Figure 1: Combination of analysis approach and view.

We will have a look at each of the six combinations at the end of this section and outline what it contributes to the overall picture of privacy for a given service composition (Table 2). Before that we will discuss the three views individually. We will not describe them entirely because research on this topic is still ongoing and this document shall concentrate on the requirements for a technical framework.

3.2.4.1 View on Structure

The structure analysis deals mostly with the structure of the service composition. It is not so important what the activities actually do, but how they relate to each other. One way to express that are data flow diagrams. These diagrams are comparable to directed graph and capture essentially the sequence of activities and external data sinks and sources, such as databases and user input. Data flow diagrams may present different level of abstractions, i.e. an activity can in a recursive way be seen as a service composition. The analysis of the graph structure allows conclusions about the fact which PII can (potentially) reach which activity. This could eventually allow violations like anonymised data is merged with other data coming from the same source.

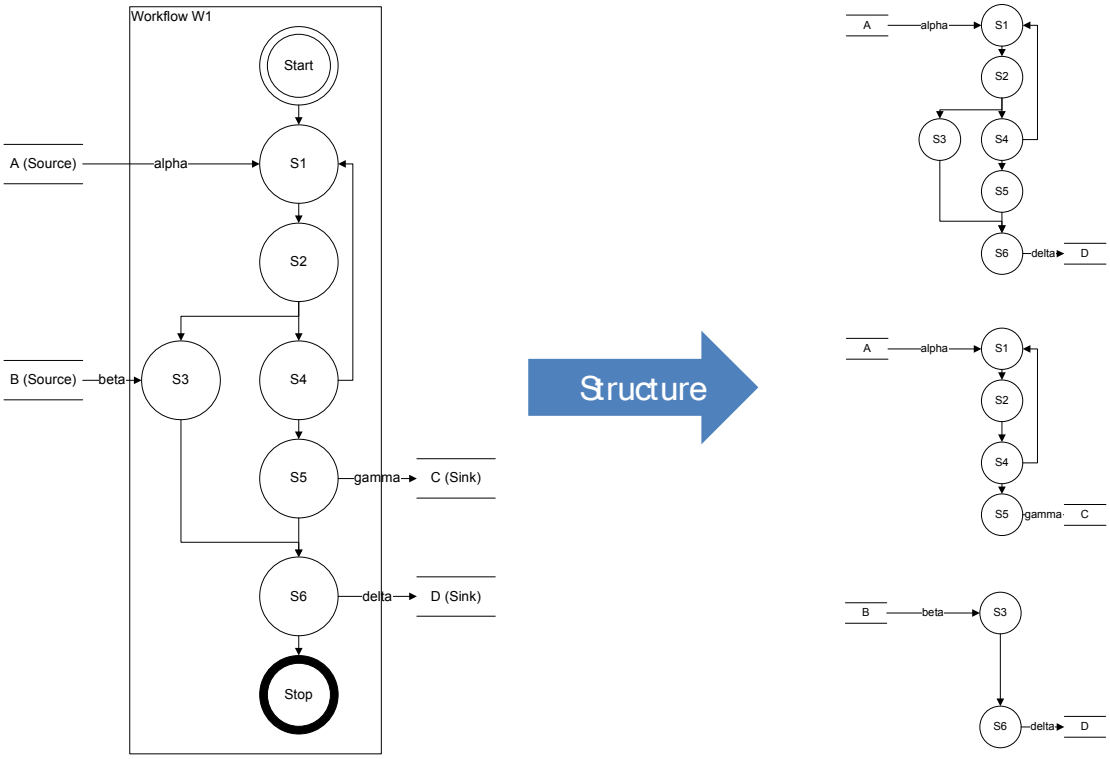

Figure 2: Graph analysis on a sample workflow

Figure 2 illustrates a sample workflow and shows a conclusion that could be drawn from it. The arrow labelled ‘structure’ analyzes the paths through the workflow from each source to each sink. This is an easy way to understand which PII is potentially processed by which activity.

3.2.4.2 View on Activities

The activity view does not look at the entire structure of a workflow but at its building blocks, namely the activities. The view looks into the activities and contributes knowledge about how each activity is handling data. Again, this view can be applied at design time (static analysis) and at run-time (dynamic analysis).

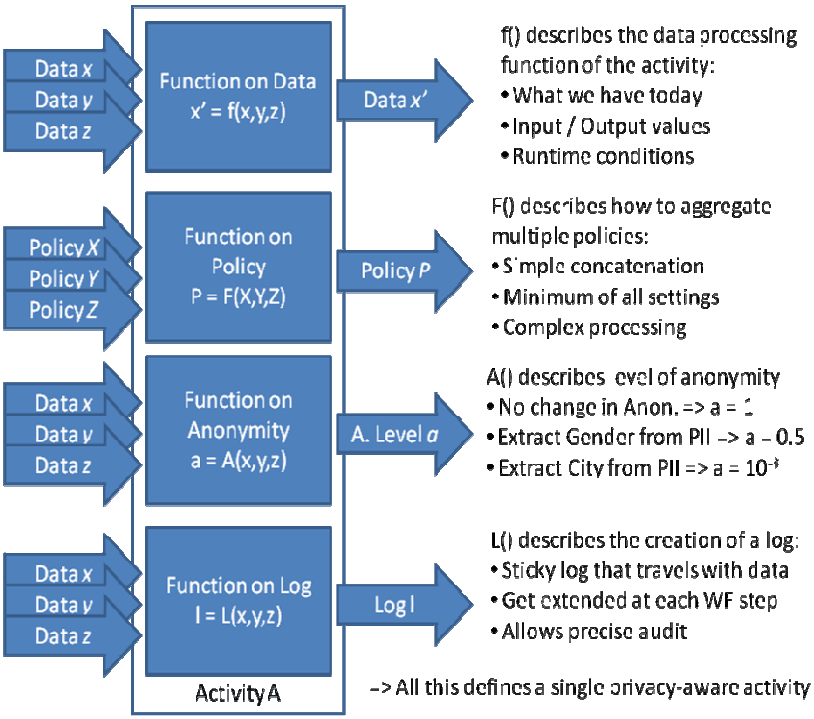

Figure 3: Four functions that define a privacy-aware activity node

Figure 3 shows an exemplarily result. In terms of privacy each activity could be represented with (at least) four independent functions:

- Function on data
- Function on policy
- Function on Logging
- Function on anonymity level

It is important to understand that an activity may be described with all three functions, it is not that the three functions are an either or.

The function on data, which we call f(x,y,z) in the example above, is just the operation which the activity performs. It receives input data and generates output data, the output data will then eventually consumed by other activities as input data. It could be an anonymisation function like filtering out only the city names from an address dataset, or it could be a mathematical function like calculating the median of a list of salaries.

The function on policy, F(X,Y,Z) in the figure, is the operation that an activities applies on a sticky policy travelling with the input data. It generates a new policy P that could be attached to the output of the data function f(x,y,z). The policy function could be simply a concatenation or picking the minimum for each individual setting.

The function on privacy Logging L(x,y,z) describes all the processes that have been carried out on the data from f(x,y,z), aggregates these with those from activities prior than Activity A, and delivers them to the next activity. Concatenation and encryptions mechanisms can be applied for non-repudiation. It is important to note, that this is solely the logging necessary for accountability in privacy compliance. The activity might keep additional logs for security purposes (which might be referenced in the privacy logging, if they include relevant data). This logging might again include personal data concerning those employees processing the data, which means, that a protection scheme has to be applied.

Finally, an activity could have function on anonymity, which we denote as A(x,y,z) in Figure 3. It gives an idea about the the level of anonymity which this specific activity creates. If an activity singles out the city from a PII dataset it provides a level of anonymity of 10^{-3}, assuming that a city usually has at least 1000 inhabitants. Knowing the anonymization functions of all activities allows to measure anonymity of a given data set even at design time.

3.2.4.3 View on Security patterns

The pattern view shall inspect a workflow with using a special knowledge. This knowledge is a dictionary of well known mistakes that regularly occur in software systems. This dictionary is often referred to as patterns. In our case it is security and privacy patterns that we are concerned about. Patterns may regard both structure and activities together, but since they are so special it is durable to see them as an independent view of the analysis. Typical privacy patterns could be:

- Sending PII to a third party
- Receiving PII
- Processing PII
- Storing PII externally
- Summarize set of PII

Patterns can be on level of single activity or on level of groups of activities. An example for a pattern on activity level is “sending PII to external service”. An example for a group-level pattern is “reading, processing and storing PII” since it involved multiple activities. Patterns which are detected at design time could yield to suggestion for the designer to avoid this setting or at least to raise her awareness.

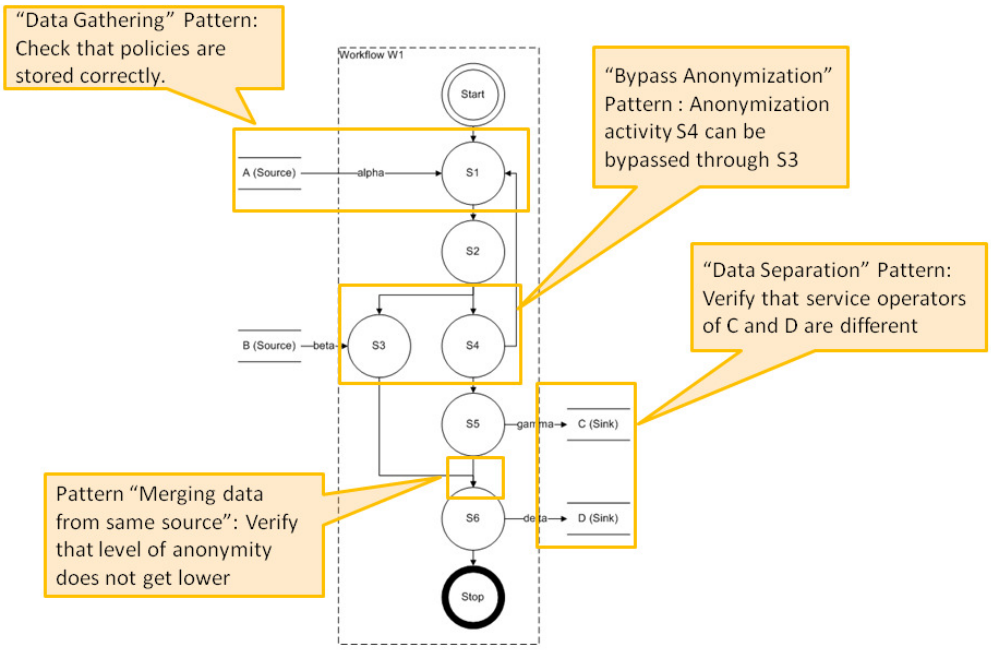

Figure 4: Detecting privacy patterns on a sample workflow

Figure 4 shows possible violations of an example workflow at design time. We point out some places that have implications on the privacy.

One way to deal with detected patterns could be to add “watchers” to activities. A watcher is some extra functionality, which is represented by an extra piece of code or even an extra activity. A watcher could then be used at run-time to

- observe communication
- Perform run-time checks, act as AC on DHP level
- Observe that minimum privacy requirements are met
- Generate meaningful log data

Figure 5 shows some examples for extra functionality to deal with the detected patterns from Figure 4.

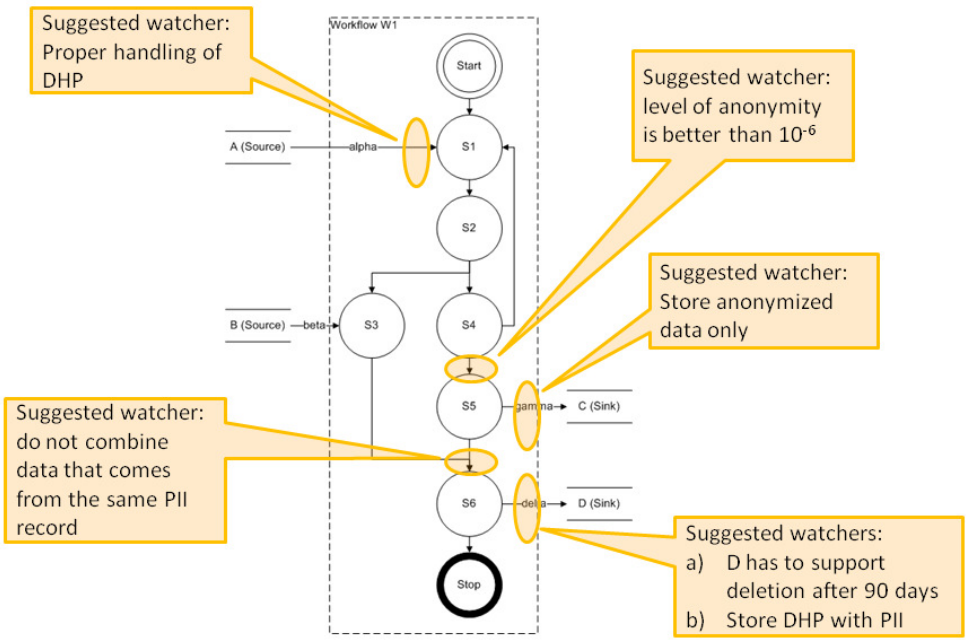

Figure 5: System suggests watchers in a sample workflow

3.2.4.4 Holistic view in detail

Table 2 describes the benefits that can be taken from each combination of approach and view.

View	Static view	Dynamic view
Pattern	Detect suspicious processing patterns; determine interesting points for watchers; suggest type of watcher	Detect dynamic security patterns, such as a potential privacy breach by invoking the same third-party operator at two places that should be kept separated. Put watchers in place, audit execution based on watchers
Structure	Detect problems such as merging data from same source; find processing paths; allow hierarchic view on privacy	Observe what path PII actually takes in workflow; log which PII was processed by which service provider
Activity	Calculate minimum level of privacy; make estimations about overall policy transformations; suggest rearranging of activities	Calculate actual level of anonymity for individual PII; perform policy transformations

Table 2: Combination of analysis approaches and views

While Table 2 describes the benefits individually, it sounds very promising to use different results together. For instance, one could use the graph and the flow approach to calculate the anonymity for various routes that PII can potentially take through the workflow. This could then be enhanced with the pattern approach and its watchers to detect if data takes a route that provides less anonymity.

3.3 Key topics for research

Privacy-aware modeling of the composite service, e.g. data-handling annotations of the business model, will facilitate overall vision of data handling in the composition. We aim at addressing both dynamic and static analysis in our research:

- Providing a high-level picture of data-handling process that would simplify compliance check.
- Facilitating adoption of new regulations or auditors expectations, e.g. specifying more precise audit logs.
- Static analysis of privacy in composition, e.g. analysis of data-flow in a workflow to see whether static constraints are violated? Constraint-aware modeling tools, support for external audit. Need DHP metadata.
- Combining privacy constraints (and resolving conflicts), e.g. policy composition in eCV scenario .
- Run-time enforcement of privacy constraints, e.g. make sure that a workflow does not violate external privacy commitments (à la EPAL).

Chapter 4

Requirements for SOAs

The depicted requirements are a progression of previous research done in [21]. In this research, a use-case gap-analysis was conducted and matched against an analysis of the technical and legal framework. The current research advances the results in several aspects.

The structure of the requirements relates to the different functions of activities of services as depicted in the technical framework,(see above, Chapter 3) so first policy requirements are described, followed by transparency requirements. The latter are subdivided into two parts: logging and access to the (core) information. Thus a threefolded structure comes into play, roughly mapping to the three states, before (policy), during (logging), and after processing (access to primary information). Primary information in this context denotes those sets of data, that are object of the processing, whereas logs, document such processing.

For each of these structural elements, the same types of requirements reoccur: the need for formalisation, the non-repudiation, and accessibility, and the answer to the questions: Who processed the data? What data is processed? How is the data processed? and to whom is the data transferred to? Table 2 maps the core requirements for each state to the elements mentioned above. For accessing primary information the data subject needs to approach the processor, and therefore must know this processor. Consequently there is no requirement on addressing the question who processes this data.

	Policies	Logging	Primary Information
Formalisation	Req't No. 1	Req't No. 10 & 11	Req't No. 19
Non-repudiation	Req't No. 2	Req't No. 12	Req't No. 20
Accessibility	Req't No. 3 & 4	Req't No. 13	Req't No. 21
Who	Req't No. 5	Req't No. 14	n/a
What	Req't No. 6	Req't No. 15 & 16	Req't No. 22
How	Req't No. 7	Req't No. 17	Req't No. 23
To Whom	Req't No. 8 & 9	Req't No. 18	Req't No. 24

Table 3: Mapping core Requirements to functions

The three functions are supplemented by a short section of cross-domain specific requirements, which are mostly relating to the policy function. Finally a number of requirements for additional mechanisms that fit in neither of the categories are outlined.

The requirements follow a certain usage of the words “must” and “should”. A requirement must be followed, if this is considered a direct effect of legal compliance. The other requirements should be observed to allow for privacy-enhancing functionality of a Service Oriented Architecture. For each of the requirements a short description is given, where applicable a possible solution is envisioned and a reference to the Scenario is made available for exemplification.

4.1 Core Policy Requirements

Requirement No. 1 Policies should be available in an unambiguous formalisation. Thereby, the content of policies should be machine interpretable.

Policies are used by service providers to describe restrictions on the processing of personal data. From a privacy point of view, policies on purpose limitation, non-disclosure and data retention period are of major importance. Since policies should be available for automatic processing and comparison with user preferences, they have to be available in a machine-interpretable form. To avoid misinterpretation of policies and thus reduce legal conflicts, unambiguity in the formalisation is necessary. However, there may be cases, where policies given in natural language cannot be transformed into such strict machine-interpretable form.

eCV-Scenario: Requirement 1 is exemplified at 5.1.2.

Solution: A formal language, possibly supported by ontology to avoid ambiguity, has to be standardised.

Requirement No. 2 It must be ensured that communicated policies cannot be argued by the ensuring entity.

Policies must be binding, i.e. the ensuring entity must not be able to argue their existence and exact content. This requirement can be met by using electronic signatures.

eCV-Scenario: Requirement 2 is exemplified at 5.1.1.

Solution: The ensuring entity digitally signs its policies using cryptographic methods.

Requirement No. 3 Policies must be easily accessible to users. The way of accessing the policies should be determined by a clear specification.

Potential users of a service should be able to see the policies of every service provider without troubles. A standardised means of access could be made available, but should only provide as little communication as possible, to limit the amount of data made available through this communication.

eCV-Scenario: Requirement 3 is exemplified at 5.1.1.

Solution: Policies are part of the description of the interfaces of the services. They can be automatically accessed through applications.

Requirement No. 4 Policies should be presented to users in an easily comprehensible manner.

As policies can be very complex, users that do not have detailed legal knowledge might not be able to understand and assess them. Thus, policies should be described in a manner that is easily comprehensible to the general public.

eCV-Scenario: Requirement 4 is exemplified at 5.1.1.

Solution: Use of a formalism allowing for the presentation of policies in different degrees of abstraction.

This solution is in line with the multi-layered approach followed by the Article 29 Working Party in its opinion on more harmonized information provisions (WP 100). The three layers identified by the working party (short, condensed, full) could be amended by an additional layer introducing a meaningful iconography.

Requirement No. 5 It must be explicitly determined who is responsible for the policy, including a reference to the applicable jurisdiction.

This determination must be visible for users.

eCV-Scenario: Requirement 5 is exemplified at 5.1.3.

Solution: A reference to the assuring entity is given within the policy.

Requirement No. 6 It must be explicitly determined what data are covered by a policy. This determination must be visible for users.

A clear link between data and policy is needed, since different services of one provider may give varying policies, respectively for different parts of one set of data. This is an effect of data separation. It is advisable to communicate policies for each purpose separately.

eCV-Scenario: Requirement 6 is exemplified at 5.1.1.

Solution: Policies are to be integrated into the description of the interfaces of services.

Requirement No. 7 Policies should cover all aspects of data processing with regard to privacy legislation.

Policies can be of arbitrary detailedness. In order to prevent unnecessary complexity, they should not be more detailed than legally / contractually necessary.

eCV-Scenario: Requirement 7 is exemplified at 5.1.1.

Solution: Policies could be generated using an expert system that allows for generating wordings that are compliant with the law.

Requirement No. 8 Recipients or categories of recipients to which the data will be passed on to, must be explicitly determined. This must include a reference to the applicable jurisdiction for the recipient.

eCV-Scenario: Requirement 8 is exemplified at 5.1.3.

Solution: Recipients or categories of recipients are specified within the policy.

Requirement No. 9 It should be explicitly determined under what policies data is passed on to other parties.

If personal data is passed down a service chain, the receiving service provider is legally bound in regard to what it may do with this data. As this may be only a subset of what the originating service may do, this should be reflected in a separate (derived) policy.

eCV-Scenario: Requirement 9 is exemplified at 5.1.3.

Solution: A separate policy is attached to the personal data.

4.2 Privacy Logging Requirements

Requirement No. 10 Log files should be available in an unambiguous formalisation and their content should be machine interpretable.

If logging takes place in a log file jointly used by different organisations that form part of a cross-domain service composition, it has to be agreed upon a common log format. Even if every service provider generates its own log files, the use of a standardised log format facilitates partly automated access to information. In order to ensure the unambiguousness of information to be accessed formalisation should be non-ambiguous as well.

eCV-Scenario: Requirement 10 is exemplified at 5.1.4.

Solution: Machine interpretability is achieved by determination of a formal language whereas unambiguousness is obtained by an agreement about a joint ontology.

Requirement No. 11 It must be possible to check the compliance of processing operations with communicated policies on the basis of log files afterwards.

Using log files allows reconstructing of processing of data by the service. Thus, it is possible to match policies and log files and to identify incidents that are not compliant with the policies.

eCV-Scenario: Requirement 11 is exemplified at 5.1.4.

Solution: Usage of a semantic formalisation as basis for the description of policies and for logging allows for partly automatic review of compliance.

Requirement No. 12 It must be ensured that log files cannot be argued by their originating entity in charge of the processing.

Not only policies (requirement 2), but also logs must be binding. The originator must not be able to argue that it generated the log file in the existing form. This requirement can be met by means of electronic signatures.

eCV-Scenario: Requirement 12 is exemplified at 5.1.4.

Solution: Logs are digitally signed by the originator on the basis of cryptographic methods.

Requirement No. 13 The fact that data are logged must be visible to the user.

When logging the processing of personal data, these logs themselves often are personal data. This information needs to be visible to the user as part of the transparency principle. The user must be informed of the fact that logging is applied, and information on the specific logs may be in scope for subject access requests.

eCV-Scenario: Requirement 13 is exemplified at 5.1.4.

Solution: The log data is attached to the personal data (concept of “sticky logging”).

Requirement No. 14 The originator of a logging entry must be clearly visible. In particular, it must be visible which service provider of a cross-domain service composition is the originator of a certain logging entry.

Logging also serves the purpose of proofing the lawfulness of the data processing, e.g., that data was corrected, or that it was not accessed by a specific employee etc. It must therefore be clear, which entity the log entry originates from. This is especially relevant if several entities write to the same log file.

eCV-Scenario: Requirement 14 is exemplified at 5.1.4.

Solution: For each log entry, the entity from which the entry originates from is described.

Requirement No. 15 A simple methodology must allow access for the user to those logs or parts thereof, to which s/he has a legal right to access, or to which the service provider wants to grant access to.

Legal transparency requirements give the data subject the right of access to any information available on him or her. This includes especially the right to know what data have been processed for what purpose, whether they were changed, and in some cases also how they were processed (especially if automatic decisions were taken). Additionally all entities that have received any information about the data subject must be named. In some cases the service provider might be interested in allowing access beyond what is needed for legal compliance to support the trust relationship with the user, for example access to every single processing.

eCV-Scenario: Requirement 15 is exemplified at 5.1.4.

Solution: The log files serve as a core repository of relevant data, but need to be supported by access mechanisms.

Requirement No. 16 It must be clearly visible to what data a log entry refers.

Logs are one source of information for subject access requests. For this purpose logs must describe actions that were applied to personal data (such as modifications, transferrals, possibly also simple reading access). When a service is invoked a number of processes could be applied to different data, therefore the log must be unambiguous in describing what data it refers to.

eCV-Scenario: Requirement 16 is exemplified at 5.1.4.

Solution: One possible solution is that log data is attached to the personal data ("sticky logging").

Requirement No. 17 Log files should describe all contractual and further legally relevant aspects of data processing. Beyond that, technical aspects should only be described in case they are relevant.

A service could apply several actions with several purposes on a set of data. This may and in some cases must be reflected in the logging. Additionally the corresponding entities or services and the data that were target of the logged action need to be referred to. Obviously logs can get huge and large amounts of data can be produced. Not all actions however, need to be logged, but only those that are relevant with regard to data protection. Relevant anyway, are most of the processes applied on the data, especially changes or corrections, deletion, but also the information from whom the data was received, and to whom it was send to.

eCV: Requirement 17 is exemplified at 5.1.4.

Solution: A filter is applied to the logs. Such a mechanism could be developed through an expert system and supported by experts with the necessary legal and technical background.

Requirement No. 18 Log files must contain explicit information on recipients or categories of recipients data have been passed on to. This includes a reference to the applicable jurisdiction.

This requirement is derived from the legal duty to ensure transparency with regard to recipients or categories of recipients of personal data.

eCV-Scenario: Requirement 18 is exemplified at 5.1.4.

Solution: Recipients or categories of recipients are included in sticky logs.

4.3 Requirements on access to primary information

Requirement No. 19 Access to personal information should be provided in an unambiguous formalisation. The content of the information should be machine interpretable.

If users of a service shall be able to analyse accessed information in a partly automated manner, a machine interpretable formalisation is indispensable. In order to on the one hand avoid misinterpretation of accessed information and on the other hand prevent possible legal disputes about differently interpreted information, unambiguousness of formalisation is required.

eCV-Scenario: Requirement 19 is exemplified at 5.1.1.

Solution: Information accessed by users is generated on the basis of logs that are in line with requirement 10.

Requirement No. 20 It must be ensured that access to information that has been granted cannot be argued by the granting entity.

Equal to the logs, the answer to subject access requests must be binding. The sender of the information must not be able to argue the information it has delivered.

eCV-Scenario: Requirement 20 is exemplified at 5.1.1.

Solution: Electronic signatures could be implemented to support such mechanisms.

Requirement No. 21 A simple methodology with regard to request and granting of access to information should be provided to users.

Users of a service should be enabled to easily get access to the information they provided. For this a standardised procedure should be used, so the efforts for such accesses are kept low on both sides. Through standardised clauses an automation of the process - at least partially - could be feasible.

eCV-Scenario: Requirement 21 is exemplified at 5.1.1.

Solution: A standardised interface for subject access requests is supported by every service. This interface is mentioned in the description of the service.

Requirement No. 22 Users accessing information must be enabled to easily recognize what data covered by what policy have been disclosed to what third parties.

If personal data of users are processed when a service is invoked, they have the right to obtain information from the service provider about categories of processed data, purposes of the processing, and recipients or categories of recipients.

eCV-Scenario: Requirement 22 is exemplified at 5.1.3.

Solution: One solution is to make use of sticky logging as described in requirement 16 and to enable users to access the logging data attached to personal data.

Requirement No. 23 Accessed information should cover only contractual or further legally relevant aspects of data processing.

Service providers are legally obliged to grant users access to specific information (see requirement 17). In principle, the accessible information provided should be limited to this specific information in order to avoid too much complexity.

eCV-Scenario: Requirement 23 is exemplified at 5.1.3

Solution: As repository for accessible data, sticky logs are used. To avoid too much complexity, a filter is applied to the logs that is developed through an expert system and supported by experts with the necessary legal and technical background (see requirement 17).

Requirement No. 24 Users must be enabled to access explicit information on recipients or categories of recipients data have been passed on to. This includes a reference to the applicable jurisdiction.

eCV-Scenario: Requirement 24 is exemplified at 5.1.3

Solution: As repository for accessible data, sticky logs are used. Recipients or categories of recipients are included in these logs (see requirement 18).

4.4 Cross-Domain-specific Requirements

Requirement No. 25 It must be possible to maintain communicated policies even if the Service Oriented Architecture is dynamically adapted (refers to the constellation of a SOA being established by several entities).

It may happen that a member of a Service Architecture leaves the organisation and is replaced by another entity. Dynamic changes of this kind should be possible without resulting in the need to negotiate policies once again with customers or even in the necessity to terminate contracts with customers. This requirement does not apply to the virtual organisation: The formalisation of policies e.g. may not restrict replacement of enterprises and their services during their run-time.

eCV-Scenario: Requirement 25 is not exemplified in the scenario.

Solution: With the aid of semantic descriptions it is checked - as far as possible -, whether planned changes of the virtual organisation are deemed to be possible if considering policies that have been communicated. Policies generated by means of an expert system as introduced in requirement 6 facilitate such changes of the virtual organisation because they do not unnecessarily restrict these facilities for alteration.

Requirement No. 26 If it is not possible to maintain (all) communicated policies in case of an adaptation of the virtual organisation, it must be possible to adapt the communicated policies (builds on requirement 25) through renegotiation, if this fails the service must be stopped.

This requirement complements the previous one: As already mentioned, it sometimes might not be feasible to retain policies when undertaking - possibly inevitable - alterations of the virtual organisation. In such cases, mechanisms have to be in place allowing for adaptation of already communicated policies to the new conditions in mutual agreement. Alternative, it must be possible for customers to withdraw from a contract.

eCV-Scenario: Requirement 26 is not exemplified in the scenario.

Solution: Negotiation of new policies compliant with the law is technically enforced before data are processed in a manner that infringes old policies. At this, semantic descriptions of policies allow for identification of necessary changes and thus offer a basis for renegotiation.

Requirement No. 27 A service provider whose service is a downstream part (those that process data later) of the overall workflow must adhere to policies given by service providers whose services are upstream parts (those that process data first) of the workflow.

As the service provider who is in contact with the customer makes binding policies for the whole workflow, service providers whose services are downstream parts of the overall workflow have to adhere to these policies.

eCV-Scenario: Requirement 27 is exemplified at 5.1.4.

Solution: In order to achieve that common policies do not have to be negotiated in advance, a mechanism is applied that generates new preferences from existing preferences and policies: At the first service of a workflow customer preferences and policies of the service are matched. The result of the matching process then is matched as set of preferences with the policies of the second service. If preferences and policies are specified on the basis of the same semantic formalism, new preferences can be derived partly automated from them by means of a reasoner.

Requirement No. 28 Multi-level-matching within a Service Oriented Architecture must be supported.

A multi-level-matching always takes place, when a Service A, which is approached by a user, launches another Service B. In this case Service A has to integrate the policies of Service B.

eCV-Scenario: Requirement 28 is exemplified at 5.1.4.

Solution: Multi-level-matching of policies is enabled by means of formal methods as used for software verification.

Requirement No. 29 The ability of the data subject to have access to information must be ensured for the future.

If subject access requests are answered on the basis of logs this can invoke serious difficulties, if a Service Composition or a Virtual organisation is later decoupled. It could be difficult to identify all parties that participated in the specific service. Therefore mechanisms need to be implemented, that allow subject access requests for a longer period of time than the actual service composition is available.

eCV-Scenario: Requirement 29 is exemplified at 5.1.3.

Solution: The logging is attached to the personal data themselves, as metadata. In a service flow, the final step is to deliver the whole data including this metadata back to the original service.

Requirement No. 30 A ex post notice must be enabled by appropriate mechanisms.

If policies change, an ex post information of the user becomes necessary (see reqt's 25&26). Therefore mechanisms need to be included, that allow for notice in multi-level workflows, even if the user is not known to all services. Equally it must be possible for the user to accept the changes towards all included services.

eCV-Scenario: Requirement 30 is exemplified at 5.1.2.

Solution: Standardised interfaces, allowing information against the stream of the workflow.

4.5 Requirements for additional mechanisms

Requirement No. 31 It must be ensured that correction and erasure of user data are feasible.

Data protection legislation gives each data subject the right to rectification and erasure of his/her data to be used towards the controller of the processing. In order to be able to fulfil its obligations, the service provider must be capable of specifically manipulating customer data as well as log entries and backup data about single persons.

eCV-Scenario: Requirement 31 is exemplified at 5.1.2.

Solution: Customer data and logs are stored in a database or a data format that allows tender-granular manipulations.

Requirement No. 32 It must be ensured that blocking of user data is feasible.

If data can or may not be erased there must be a mechanism in place that restricts their further use to the necessary minimum for the given situation. Partial blocking of subsets of a larger data set must be feasible.

eCV-Scenario: Requirement 32 is exemplified at 5.1.2.

Solution: In order to implement the blocking, data are encrypted and stored at the location where they already were stored unencryptedly. Subsequently, the key may only be used for enforcement of legal obligations as access to information or for audits.

Requirement No. 33 It should be made easy for users to exercise their rights of correction, erasure and blocking.

As correction, erasure and blocking are instances that are initiated by the user, technical feasibility as such is not sufficient as requirement. Rather, it shall be smoothly possible for the user to exercise his rights.

eCV-Scenario: Requirement 33 is exemplified at 5.1.2.

Solution: The interface described in the solution offered to requirement 28 is analogously provided for correction, erasure and blocking.

Requirement No. 34 It should be possible to guarantee compliance with communicated policies.

For this, a mechanism is needed that technically prevents the service provider from infringing its policies - i.e., that means that a part of the provider's infrastructure must be exempted from the provider's direct control.

eCV-Scenario: Requirement 34 is exemplified at at 5.1.2.

Solution: The use of DRM can guarantee control of data usage and compliance with communicated policies ("DRM4privacy").

Requirement No. 35 There should be a possibility to support trust between user and service provider.

There is a need for an infrastructure that enables users to come to trust an up to now unknown provider. This can be built through reputation, amongst other mechanisms.

eCV-Scenario: Requirement 35 is exemplified at 5.1.1.

Solution: Certification of the service provider by an independent third party.

Requirement No. 36 The user shall have the possibility to express her preferences in an easy manner.

As users quite often are technical and legal amateurs, tools enabling them to express their preferences in a formalized manner (as defined by requirement 1) should be made available to them. These tools should be easy to use. The preferences can be the basis of a partly automated negotiation of new policies.

eCV-Scenario: Requirement 36 is exemplified at 5.1.2.

Solution: Provision of a well defined ontology that is limited to concepts necessary for the explanation of users' preferences. This ontology serves as basis of an expert system supporting the user at drawing up her preferences.

Requirement No. 37 User *and* service provider should be able to match preferences and related policies.

In principle a match of preferences and policies could be processed either on the service or on the user side. To allow for different market requirements both alternatives, such a matching should be possible on both sides. This is also a matter of trust. Does the user or the service provider trust the entity doing the matching? If both parties are enabled to do the matching themselves, a manipulation by one party would become obvious to the other.

eCV-Scenario: Requirement 37 is exemplified at 5.1.2.

Solution: Both entities have a software for the matching or can do it visually.

Requirement No. 38 Matching of preferences and policies must be comprehensible.

Matching must take place in such a way that both user and provider can comprehend the result, and that the entity processing the matching can reason, that it was done correctly. In those cases, where preferences and policies do not macht, negotiation mechanisms could be employed which adopt the policies according to the preferences.

eCV-Scenario: Requirement 38 is exemplified at 5.1.2.

Solution: The result of match and mismatch of preferences and policies must be presented to the user.

Requirement No. 39 A mechanisms to express the anonymity set with regard to a specific data type should be supported

Every operation should make a statement about the influence that its functionality has on the anonymity of a given set of data records. This allows estimating the overall level of anonymity that a workflow provides.

eCV-Scenario: Requirement 39 is not exemplified in the scenario.

Solution: A simple anonymity function could state how on many people a given set of attributes applies. A full disclosed PII data record has an anonymity level of 1 since it applies only to one individual. If an activity filters only the city name from an address record the anonymity level would be 10^4 assuming that a city has a minimum of ten thousand inhabitants; the filtered data record could apply to each of them. If in addition to the city the gender was given, the anonymity level would lower to $0.5 \cdot 10^4$.

Chapter 5

eCV Scenario

This text describes a scenario that features both privacy in service composition and policy aggregation. Moreover it offers interesting economical and legal questions. It is intended for further discussion on scenarios and requirements. Although WP6.3 could be one of the key enabler of this scenario, it is not planned to be the key result of WP6.3.
Beyond exemplifying the requirements in this Heartbeat, the chapter addresses the following key research aspects:

- Policy composition
- Privacy-friendly service composition
- Complex matching of policy and preferences
- Sticky policies on claims and documents
- Scenario could be easily extended with anonymous credentials
- Scenario offers variations including the involvement of SmartCards
- Variations offer complex cases for economic and legal research
- Opportunity to show-case economic feasibility of privacy in SOA

5.1 Core Scenario: A Social Community Portal

Inga Vainstein[22] is 46 years old and is currently working as journalist. As a part of her job she is travelling to various countries. She is member of a professionals' online community, which helps her to stay in touch with people she works with. The social platform Inga uses looks like a mixture between LinkedIn and Monster. She is very anxious to keep her public profile clean in order to have a good reputation and to appear trustworthy. Inga is proud of all her skills and recommendations from other members of the community. She attends trainings on a regular basis. For each completed course she gets a certification. Moreover she collects testimonials from former employers. Last year she won an award for her outstanding press story on identity theft.

Besides from staying in touch with people, one of her main reasons to use this platform is to get job offers and apply for new positions in a convenient and easy way. The platform offers a feature called "one-click job application" that allows her to apply for new jobs on an ad-hoc basis.

5.1.1 Profile backed by Claims

The social platform fully supports statements issued by third parties such as universities, former employers, trainings centres, or business partners. Of course Inga cares for her privacy, but she knows that the online platform will take care of that because the operator of the platform is responsible for its lawfulness, as he determines the purposes and means of the processing of data collected from Inga and thus is the controller of the processing. Beyond trust solely based on legal obligations, the trust may further be supported by a privacy seal. This would be in line with requirement No. 35: "There should be a possibility to support trust between user and service provider."

This is supported by requirement No. 2: "It must be ensured that communicated policies cannot be argued by the ensuring entity."

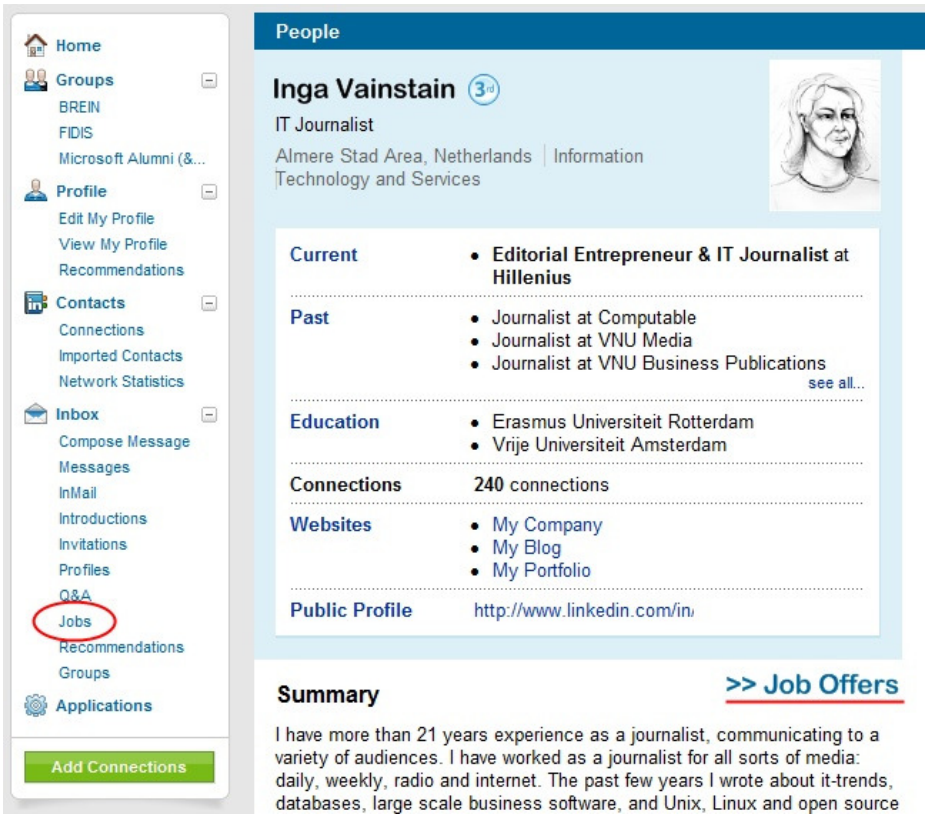

Figure 6: Professional profile of Inga Vainstein in a social community platform. All statements of career and education is backed up with claims or digital documents. The portal looks like a mixture of LinkedIn and Monster.

Figure 6 shows a screenshot (mockup) from the social community platform. The webpage shows Inga's profile. She made statements about her education, past and current employer. Since Inga voluntarily disclosed this data, the concept of specific, unambiguous, freely given and informed choice has to be adhered to.

This is supported by requirement No. 3 and No. 4: "Policies must be easily accessible to users. The way of accessing the policies should be determined by a clear specification." And "policies should be presented to users in an easily comprehensible manner."

Another important aspect is highlighted by requirement No. 7: "Policies should cover all aspects of data processing with regard to privacy legislation."

Most of her education and career statements (university degrees, former employers, etc.) are asserted either with digital claims or scanned documents. Community members can recommend other members, again these results in claims.

Thus her CV can be seen as a set of statements which are (all/partly) backed up by claims, either in the form of electronic documents or in the form of security tokens. Both the documents and tokens are typically signed so that the issuer can be verified by a third-party and both might lead to an improvement of quality of data about a person.

Claims and documents are stored by Inga once she got them from the issuer in whatever way (random assumptions):

- *Erasmus University Rotterdam* offers a web-based service for digitally signed claims.
- *Vrije Universiteit Amsterdam* offers a protected web storage where a former student can download a digitally signed PDF of her the university diploma
- *VNU Business Publications and VNU Media* sends digital claims by e-mail to Inga upon request.
- *Computable* offers a simple FTP server where it stored a letter of reference. The access is password protected. This measure serves the purpose of technically safeguarding the letter of reference that constitutes personal data about Inga.

The examples above show the usefulness of the requirements supporting subject access requests to primary information. All the documents above are at the same time personal data that are subjected to requirements No. 19, 20 and 21:

- "Access to personal information should be provided in an unambiguous formalisation. The content of the information should be machine interpretable."
- "It must be ensured that access to information that has been granted cannot be argued by the granting entity."
- "A simple methodology with regard to request and granting of access to information should be provided to users."

So the implementation of the above mentioned requirements may not only serve as a mechanism to support Inga's legal subject access request, but may also serve as a business facilitator for an additional service such as the eCV-portal in our case. If her data, the documents are stored machine readable, they can easily be integrated and interpreted by the portal. The non-repudiation requirement (No. 20) is in this case implemented via claims.

Inga's storage device could be a simple hard disk or a smart card that eventually allows her to use the claims in other places. The claims themselves are not accessible by other community members. They are only used if Inga wants to apply for a job or actively hands them out to another community member. This again refers to the legal concept of consent and choice: Claims are only disclosed under certain circumstances determined by Inga.

The validation of her profile is possible for all statements made in the social network platform, such as date of birth, nationality, skills, education and community recommendations. Issuer of the claim/document is always an organisation that is in the position to assert Inga's statements, e.g. employers, universities, training centres, government, and other community members. The following table gives examples of such claims.

Example Claims / Documents	Issuer
Day of Birth, Gender, etc.	Government
Work permission	Local administration
Profession	Chamber of Crafts
Medical examination / allowance	Independent doctor
Training certificate	Trainings center, school
Diploma, PhD	University
Flying hours of pilots	Employer
Recommendations	Community members
Salary	Tax authorities
Business contacts	Customers

Table 4: Example claims and potential issuer (in case a claim reveals information on medical examination / allowance, it contains sensitive medical data in the sense of the Data Protection Directive)

Inga is very concerned about her privacy. She does not want that claims / documents she handed out to a third party shall be used in different context. Therefore the claims and documents travel with a sticky policy. Inga has policy preferences for claims which she hands out to other people.

This is supported by requirement No. 6: “It must be explicitly determined what data are covered by a policy. This determination must be visible for the users.”

5.1.2 Combining Claim Policies

Protection of information is an important issue for the issuers as well. When producing claims they may have an specific intention for claims / documents. An example can be derived from the legal framework in Germany: In case of application with a governmental agency, a special type of police clearance is sometimes needed. Those certificates may not be used in the private sector. The use of the certificate always requires the consent of the employee.

The agency in charge of issuing the certificates is liable in case of illegal non-disclosure of those certificates. Therefore, the agency will only be willing to hand out those certificates under rare circumstances. This is supported by the mechanisms stipulated in requirement No. 34: “It should be possible to guarantee compliance with communicated policies.”

When Inga combines various claims and documents she has to respect the policies from each individual issuer plus her own preferences. Luckily the social network website features a policy engine. This engine helps her to create a common policy that regards all aspects resp. It tells her if the policies are conflicting so that she has to solve the matter “manually”, e.g. by dropping one claim, talking to the issuer, or changing her preferences.

These mechanisms are in line with requirements No. 36 – 38:

- "The user shall have the possibility to express her preferences in an easy manner."
- "User and service provider should be able to match preferences and related policies."
- "Matching of preferences and policies must be comprehensible."

Since her CV can be seen as a set of claims, it comes with a single automatically generated policy. The common policy is generated

The reasons for having a single policy are:

- Common policy "hides" privacy agreements between individual issuers and Inga
- Inga wants to be more/less restrictive on case by case basis
- Individual claims in a combined CV shall not be treated differently

This mechanism is supported by Requirement No. 1: "Policies should be available in an unambiguous formalisation. Thereby, the content of the policies should be machine interpretable."

If a mismatch of policies occurs, a mechanism is needed to inform the user. Thus, requirement No. 30 needs to be supported: "An ex post notice must be enabled by appropriate mechanisms."

In some cases Inga may detect that some information stored by the OneClickCareer Portal is incorrect or she may decide that she does not want to further use the portal or certain parts of it. In those instances, requirements No. 31 – 33 have to be supported: "It must be ensured that correction and erasure of user data are feasible." And "It must be ensured that blocking of user data is feasible." And "It should be made easy for users to exercise their rights of correction, erasure and blocking."

5.1.3 One-Click Job Search

The community portal allows a one-click job search based on Inga's electronic claims-based CV. She simply clicks on the button "Search Jobs >>" **Fehler! Verweisquelle konnte nicht gefunden werden.**and the search engine will find suitable open positions. The search automatically regards her privacy settings and thus adheres to the legal concept of purpose limitation.

Search happens in two steps:

- Step 1: Search based on Inga's CV delivers all fitting job openings

  Inga's skill set as asserted by her electronic CV will be compared with all current job openings. We assume that the portal features a search engine that does this comparison. All open jobs will be announced to the platform by the employers. They state what experience and education they expect from an applicant. Inga can influence the search parameters by stating that she looks for a job in a specific country or in a particular position.

- Step 2: Narrow results from step 1 based on privacy policy

  At the moment of the one-click job search, the policy engine in the community portal creates a common policy for Inga's CV. The policy is combined from the individual policies of the involved claims plus Inga's user preference. Moreover Inga might want to scope the common license down for this specific job application. The common policy is compared against the policy of the job service where Inga's CV is submitted to. Depending on the national laws of the country the privacy policies of the hiring company might be different.

- Examples

  Applicants in UK need to give race and gender information whereas applicants in the US have to hide this information.

Storage and deletion policy of potential employer e.g. data computation and storage in EU only, delete after 30 days

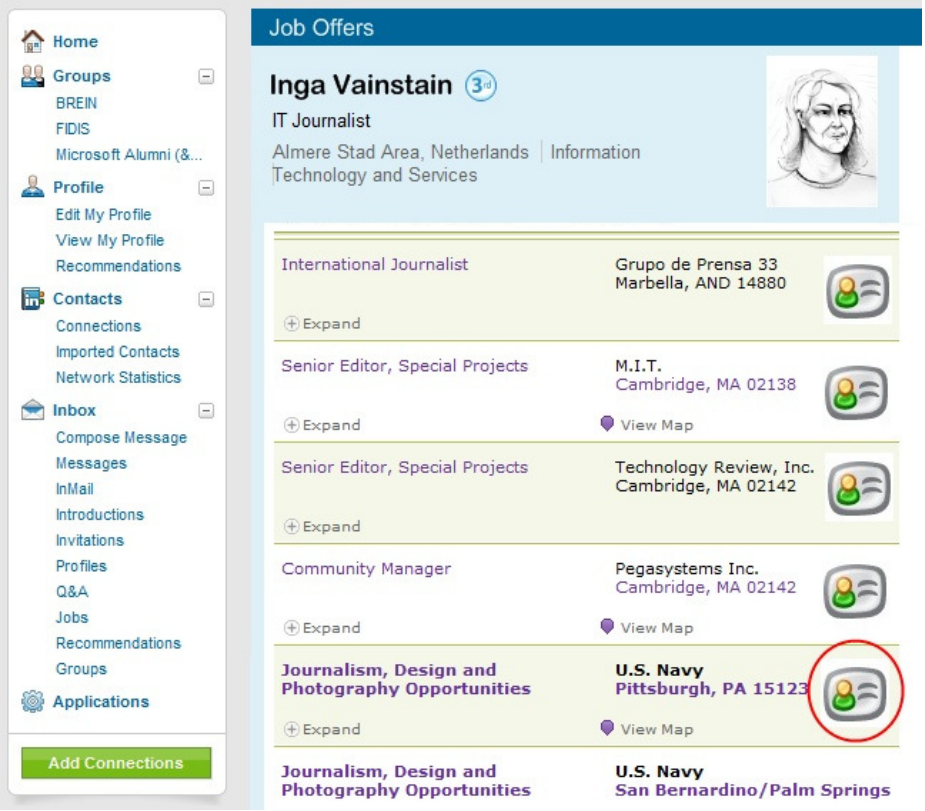

Figure 7: Social community portal offers jobs to Inga, which fit her CV and her privacy constraints.

Requirement No. 5 supports this: “It must be explicitly determined who is responsible for the policy, including a reference to the applicable jurisdiction.”

Finally Inga applies for a job by clicking a button on the website. Again, she has the choice to apply or not apply for the jobs identified by the community portal. Instead of this, a less privacy-friendly implementation could foresee that her CV and claims are automatically transmitted to potential employers.

Especially this latter case needs requirements No. 8 and 9: “Recipients or categories of recipients to which the data will be passed on to, must be explicitly determined. This must include a reference to the applicable jurisdiction for the recipient.” And “It should be explicitly determined under what policies data is passed on to other parties.” In the case of the application, this policy must include a reference to this specific purpose, other uses must clearly be ruled out.

Equally requirement No. 22 to 24 apply, for subject access after disclosure:

- “Users accessing information must be enabled to easily recognize what data covered by what policy have been disclosed to what third parties.”
- “It must be possible to maintain communicated policies even if the Service Oriented Architecture is dynamically adapted (refers to the constellation of a SOA being established by several entities).”
- And “users must be enabled to access explicit information on recipients or categories of recipients data have been passed on to. This includes a reference to the applicable jurisdiction.”

In cross-domain service compositions, one also has to consider the possibility of one service provider going out of business. In this case, requirements No. 29 applies: “The ability of the data subject to have access to information must be ensured for the future.”

Since the social community platform has all her attestation in an electronic form, Inga does not need to fill in large application forms anymore. Instead, the CV is automatically submitted to the hiring company together with her privacy policy. Figure 7 shows the result of the search, just before she clicks on a job offer.

5.1.4 Scenario Architecture

The scenario is illustrated in Figure 8. Inga Vainstain has a profile (CV) at the famous OneClickCareer Portal. The workflow used to upload user information, open job positions and search for them was designed by an employee of OneClickCareer Portal. This is important as static analysis of workflows is one of the research topics in WP6.3. The WF Designer is supported by a developer tool when he combines services. The tool informs him if his workflow does not comply with the privacy policies of one of the used underlying services (which are not shown in this picture).

Inga is a user of the portal. She uploads information and configures her privacy preference in the portal. The upload of claims / documents is done by providing a link to the issuers together with credentials to access them (step 1(In this case there could be the problem of impersonation of Inga. So she should have some delegation claims in order not to allow the Portal’s administrator to apply for random jobs on her behalf.)

Next the portal uses the links and the provided credentials to access the claims (step 2). The portal could verify that the credentials work properly by trying to get the claims from the issuer. At each job search the portal will invoke the issuer’s services again to get a fresh claim (step 2). Each issuer has its own policy, which is probably specific to Inga, i.e. they could have different policies for different students, customers, etc.

The various claims will be aggregated to a CV (step 3). This aggregation involves both the content (creating a CV from various claims) and the policies (creating a common policy from the individual policies and from Inga’s preferences).

Next the CV is taken for the search (step 4). As described above, the portal’s job matching engine will find all jobs that fit to Inga’s profile and search parameters. In Figure 8 this search yields to Job 1, Job 2, Job 3, and Job 4. We assume that the jobs are offered through Job services which have their own policies.

The search is then narrowed by comparing the common policy of the CV and each individual policy of the job services (step 5). In Figure 8 this matching yields to Job 2 and Job 3 out of the former four jobs.

This is supported by requirements No. 27 and 28: “A service provider whose service is a downstream part (those that process data later) of the overall workflow must adhere to policies given by service providers whose services are upstream parts (those that process data first) of the workflow.” And “Multi-level-matching within a Service-oriented architecture must be supported.”

Inga can now easily apply by clicking a button on the search results page (compare Figure 7). The CV will be sent to the Job service of the potential employer and processed further. This processing is in the hands of the very employer where Inga applied.
When passing data on to the employer the relevance of logging becomes visible. In case application data is leaked, this could result financial damages for Inga, which she would claim against the involved parties. Thus, the parties will have a strong interest in proving, that they were

not responsible for this leakage. Therefore requirement No. 12, 14, 16, 17, and 18 must be supported:

- "It must be ensured that log files cannot be argued by their originating entity in charge of the processing."
- "The originator of a logging entry must be clearly visible. In particular, it must be visible which service provider of a cross-domain service composition is the originator of a certain logging entry."
- "It must be clearly visible to what data a log entry refers."
- "Log files should describe all contractual and further legally relevant aspects of data processing. Beyond that, technical aspects should only be described in case they are relevant."
- "Log files must contain explicit information on recipients or categories of recipients data have been passed on to. This includes a reference to the applicable jurisdiction."

To filter out the relevant logs for such a case support of requirement No. 10 and 11 is crucial: "Log files should be available in an unambiguous formalisation and their content should be machine interpretable." And "it must be possible to check the compliance of processing operations with communicated policies on the basis of log files afterwards."

At the same time the legal principle of transparency mandates that this logging is visible to Inga (Requirement No. 13 and 15): "The fact that data are logged must be visible to the user." And "a simple methodology must allow access for the user to those logs or parts thereof, to which s/he has a legal right to access, or to which the service provider wants to grant access to."

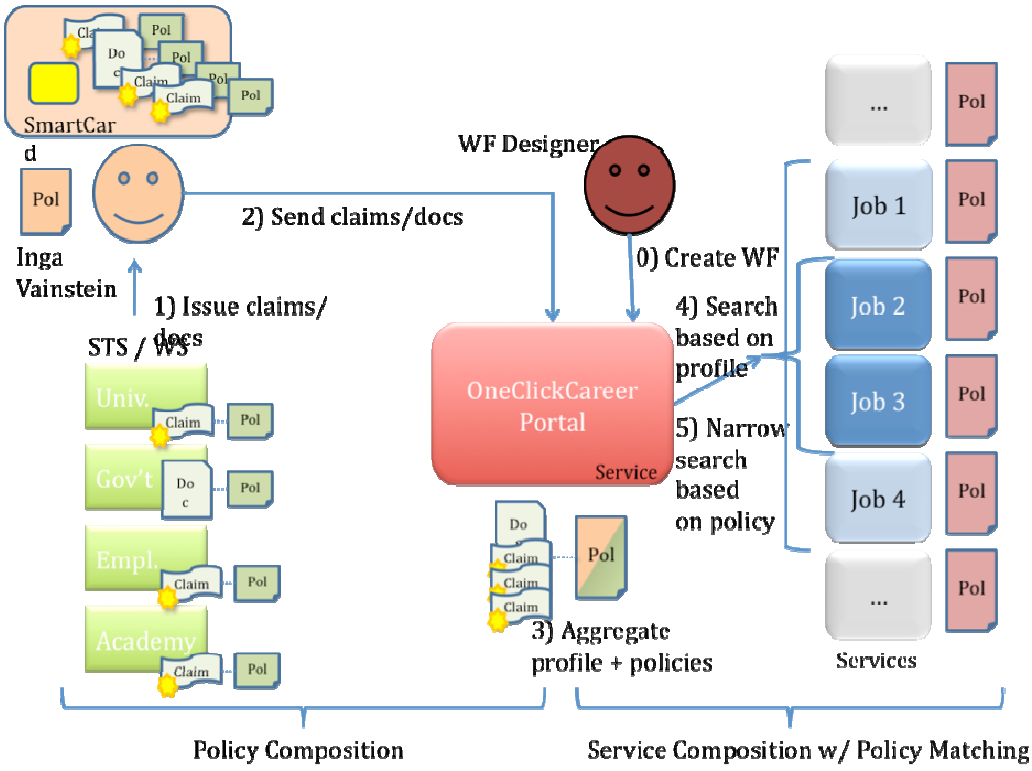

Figure 8: Illustration of the basic scenario.

The basic scenario can be easily varied to highlight various trust relationships. An easy way to change the security assumptions is to define where the claims/document are actually stored after they were issued. The next sections will highlight different variations from the example. These variations have technical, legal and economic implications. There could be even a mixture of the four variations we pointed out.

5.2 Scenario Variations based on Technical Aspects

5.2.1 Scenario Variation A: Claims stored by Issuers

Claims are stored by the issuer. When Inga configures her profile she provides only a link to the issuer's storage/service and the access credential to the social community portal. Whenever Inga does a job search, the portal will get fresh tokens from the issuers. This has a couple of the major drawbacks. First, the issuers would deduce from the invocations when and how often Inga is searching for a job. Second, Inga has to provide authorization credentials for each issuer, which is something that has strong implications on the architecture of issuer's service, i.e. we would need a delegation mechanism so that the career portal cannot impersonate Inga. Third, once the portal has retrieved the tokens it could cache them which instantly leads us to scenario B.

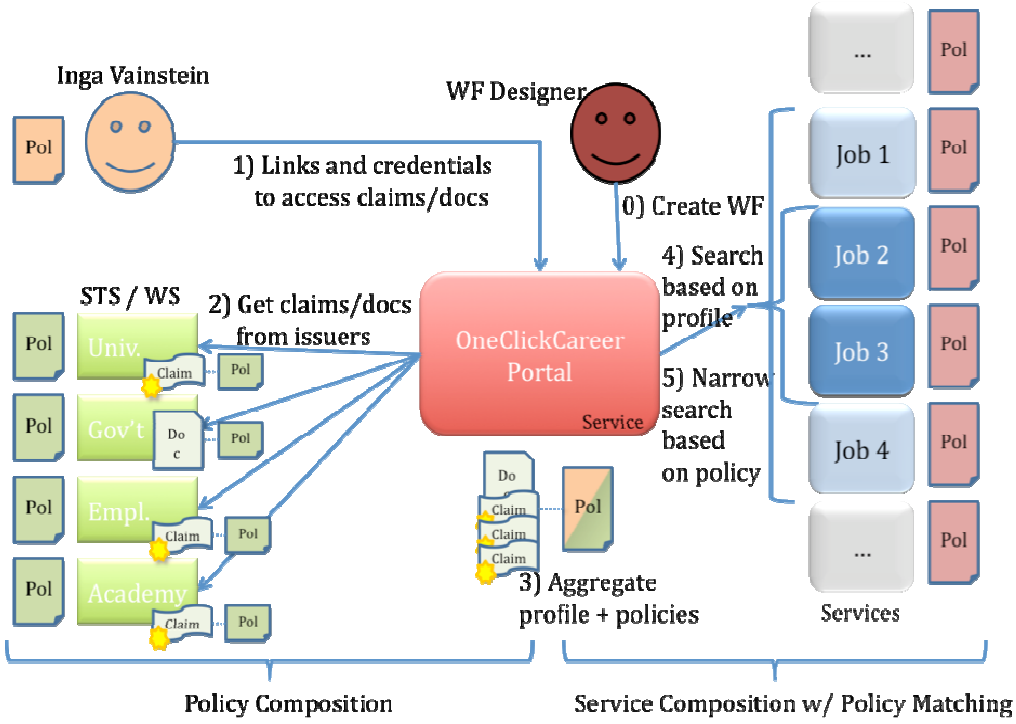

Figure 9: Illustration of the scenario variation A.

Inga did not download the documents, but configured the social network website with the

- Links to the claim/document: URL
- Form of access: WWW, FTP, WS-Trust (where we assume WS-Trust as common protocol for issuing the claims via web service)
- Credential to access: Password (or another claim issued by Inga)

for each of her claims. The configuration is assumed to be very easy from the user experience perspective. Let's assume that Inga gets an e-mail with a link that she simply copies to the respective text field on the social network website.

5.2.2 Scenario Variation B: Portal stores Claims

Inga provides the Claims to the Portal which in turn stores them. The claims are only accessible to Inga and not visible to other community members per se. Inga may want to use the stored claims when she is updating her public profile; she links a statement to a stored claim and the platform visualizes the information e.g. her university degree about as "verified information".

5.2.3 Scenario Variation C: Pure In-house solution

The assumption is that the One-Click Career Portal is used as an intra-organisational infrastructure in a large company intended for career moves within the company. Inga's claims are stored by Inga's current employer, because the employer should know Inga's CV anyway and might have large and well-maintained HR facilities that she is allowed to use as employee.

5.2.4 Scenario Variation D: Current Employer stores Claims

Inga's claims are stored by Inga's current employer; because the employer should know Inga's CV anyway and might have large and well-maintained HR facilities that she is allowed to use as employee. Inga uses the HR facilities of her current employer to feed an external Career Portal. The portal looks for Jobs outside Inga's current organisation. In this case, privacy should be carefully taken into account: the access pattern to the HR service must not be traceable by the current employer.

Claims could be also stored at social community portal. It could be a business case for the owner of the portal to provide well-maintained data store for this kind of information.

5.2.5 Scenario Variation E: Mobile Code executed at Inga's side

Claims are stored at the issuer/current employer/user, but they are never in the portal's hands. The composition intelligence is captured by a piece of mobile code that is downloaded and executed locally by the user. The output is the CV along with all its combined policies and credentials. This is then sent back to the portal, which can carry on with its task (job lookup, actual job application…)

5.3 Scenario Variations based on Economic Aspects

5.3.1 Pure Inhouse Application Scenario

In the Pure Inhouse Application Scenario the One-Click Career Portal (eCV Manager) is used solely as an intra-organisational infrastructure in a large multinational company intended for national and international career moves within this company (see figure 10).

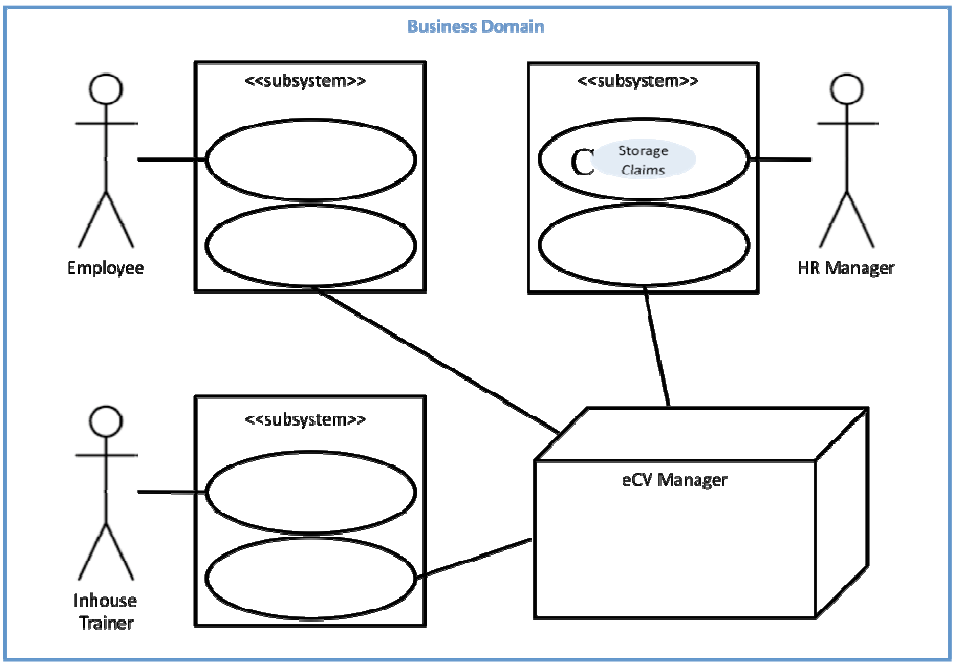

Figure 10: Illustration of the Pure Inhouse Solution

The eCV Manager in the Pure Inhouse Solution consists of three primary subsystems:

The first subsystem enables the Employee to:

- request job offers and claims for the CV-Portfolio from the HR Manager
- provide job requests and CV-Information to the HR Manager
- request training offers and training claims for the CV-Portfolio from the Inhouse Trainer
- provide training requests and CV-Information to the Inhouse Trainer

The second subsystem enables the HR Manager to:

- request job requests and CV-Information from the Employee
- provide job offers and claims for the CV-Portfolio to the Employee

The third subsystem enables the Inhouse Trainer to:

- request training requests and CV-Information from the Employee
- provide training offers and claims for the CV-Portfolio to the Employee

5.3.2 Pure Platform Application Scenario

In the Pure Platform Application Scenario the One-Click Career Portal (eCV Portal) is used solely as a multinational and -organisational platform for a variety of legal independent organisations and institutions intended for national and international career moves in different companies (see figure 11).

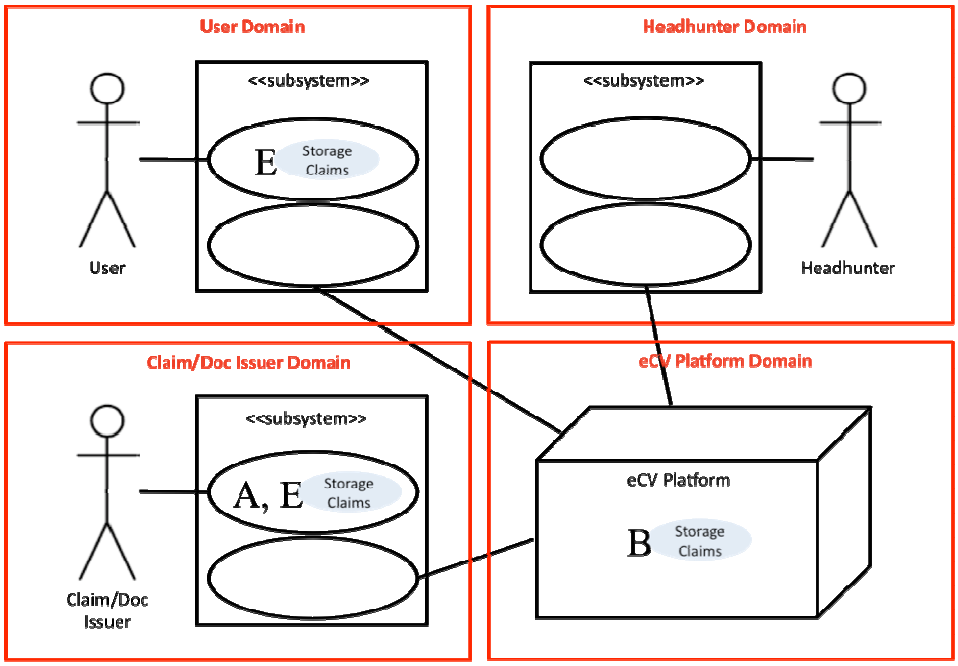

Figure 11: Illustration of the Pure Platform Solution

The eCV Portal in the Pure Platform Solution consists of three primary subsystems:

The first subsystem enables the User to:

- request job offers from the Headhunter
- provide job requests and CV-Information to the Headhunter
- request claims for the CV-Portfolio from the Claim Issuer
- provide CV-Information to the Claim Issuer

The second subsystem enables the Headhunter to:

- request job requests and CV-Information from the User
- provide job offers to the User

The third subsystem enables the Claim Issuer to:

- request CV information from the User
- provide claims for the CV-Portfolio to the User

5.3.3 Career Homepage Application Scenario

The Career Hompage Application Scenario is an extension of the Pure Inhouse Application Scenario. The One-Click Career Portal (eCV Manager) is used as an intra-organisational infrastructure in a large multinational company intended for national and international career moves for company internal employees and company external users (see figure 12).

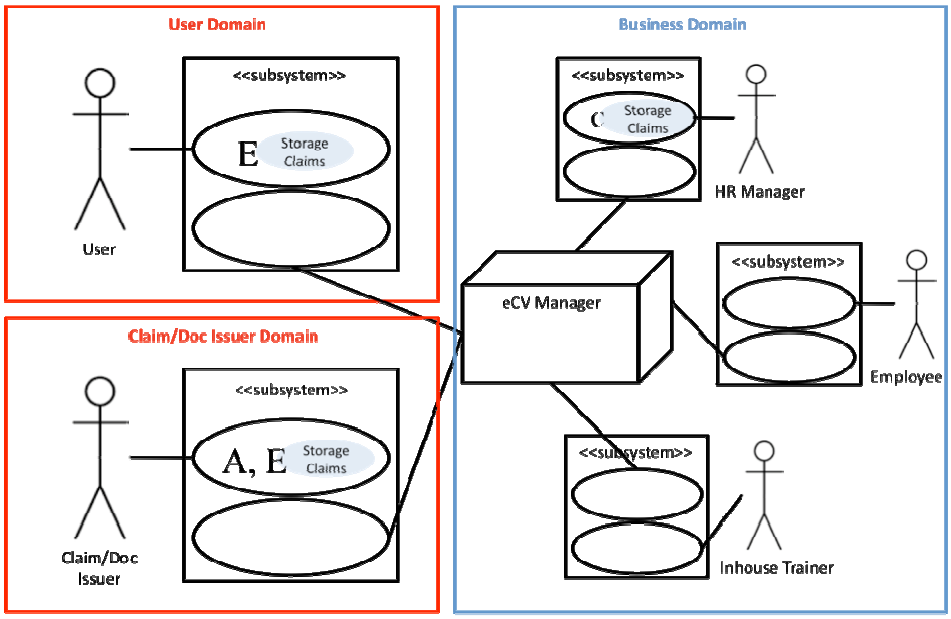

Figure 12: Illustration of the Career Homepage Solution

The eCV Manager in the Career Hompage Solution consists of two company external subsystems, extending the Pure Inhouse Solution.

The first subsystem enables the company external User to:

- request job offers from the company internal HR Manager
- provide job requests and CV-Information to the company internal HR Manager
- request Claims for the CV-Portfolio from the company external Claim Issuer
- provide CV-Information to the company internal Claim Issuer

The second subsystem enables the company external Claim Issuer to:

- request CV-Information from the company external User
- provide Claims for the CV-Portfolio to the company external User

Because of these two subsystems:

The company internal Employee is additional enabled to:

- request Claims for the CV-Portfolio from the company external Claim Issuer
- provide CV-Information to the company external Claim Issuer

The company internal HR Manager is additional enabled to:

- request job requests and CV-Information from the company external User
- provide job offers to the company external User

5.3.4 Placement Application Scenario

In the Placement Application Scenario a legal independent provider offers the One-Click Career Portal (eCV Portal) as an external organisational infrastructure to a large multinational company, intended to enable this company to replace employees from or into a variety of other legal independent organisations and for national and international career moves within this company (see figure 13).

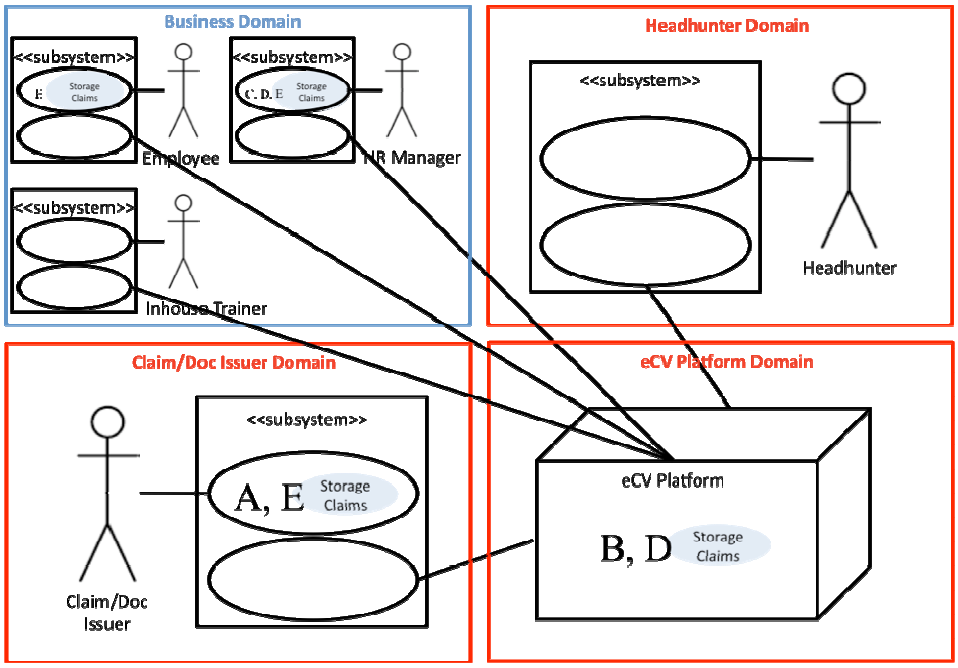

Figure 13: Illustration of the Placement Solution

The eCV Portal of the Placement Solution enables the company internal actors of the Business Domain to use the same functions as by the eCV Manager in the Pure Inhouse Solution. The only difference is that these functions are not longer offered by an intra-organisational infrastructure, but by a company external eCV Platform provider.

Additional to these functions:

The company internal HR Manager is enabled to:

- request job offers from the company external Headhunter
- provide job requests and CV-Information to the company external Headhunter
- request claims for the CV-Portfolio from the company external Claim Issuer
- provide CV-Information to the company external Claim Issuer

The company external Headhunter is enabled to:

- request job requests and CV-Information from the company internal HR Manager
- provide job offers to the company internal HR Manager

The company external Claim Issuer is enabled to:

- request CV information from the company internal HR Manager
- provide claims for the CV-Portfolio to the company internal HR Manager

Chapter 6

Conclusion and Outlook

In this deliverable we have presented a comprehensive list of requirements for a privacy-enhancing Service-oriented architecture. Service-oriented architectures pose new challenges, but also an enormous potential to enhancing the privacy of the individual.

With the requirements presented herein specific difficulties of cross-domain service orchestration or composition can be tackled sufficiently for a legally compliant solution. The law is conservative, and often can not easily be applied innovative technological developments. Therefore constructing an SOA that includes services from jurisdictions outside of the EU and some selected countries that match the level of data protection legislation will remain limited. This problem cannot be solved through technology, but needs a broader agreement on common understanding of the right to privacy and data protection.

Several new concepts and functions have been introduced in this deliverable, specifically: a policy function, a privacy logging function, and a anonymity function. If these were implemented a new level of privacy protection and accountability in services could be achieved.

This set of requirements can only deliver a starting point for such an effort, that has to be and will be continued from a research perspective within the further work of PrimeLife and beyond. A number of specific research questions are already foreseeable, when applying this set of requirements:

- How can policies express the legal requirements of purpose specification and use limitation? How can one determine, whether another service lies within the purposes of the prior one? A comprehensive ontology will be needed for such an effort, defining a taxonomy of purposes different services can be used for.
- This deliverable introduces the notion of sticky logging as one possible solution of offering transparency to the data subject on what processes have been applied on his data, and who has received the data. The mechanisms for implementing such a sticky logging, however, are beyond the scope of the deliverable.
- This applies even more so, for the anonymity functionality, which in this deliverable could only be briefly laid out. A mechanism allowing to calculate the linkability and thus the possibility of identifiability of a given set of data, would be very helpful for service composition, as this information determines whether data protection law is applicable – and beyond a legal perspective also gives measurement in regard to the confidentiality of a given set of data.

For real value in cross-domain mash-ups, however, continuous effort is needed in agreeing on languages for making services supporting these functions interoperable. Standardisationbodies will serve as a point of outreach, and partners will showcase prototype implementations of various details in near future, to getting closer to this goal.

References

[1] Directive 95/46/EC on the protection of individuals with regard to the processing of personal data and on the free movement of such data (General *Data Protection Directive*)

[2] Directive 2002/58/EC on privacy and electronic communications (*E-Privacy Directive*).

[3] Article 29 Working Party (Publisher), Opinion 4/2007 on the concept of personal data WP 136 is available at http://ec.europa.eu/justice_home/fsj/privacy/docs/wpdocs/2007/wp136_en.pdf.

[4] ISO/IEC JTC 1/SC 27/WG 5, Privacy Framework, 5th WD ISO 29100 (not published).

[5] Article 29 Working Party (Publisher), Opinion on More Harmonised Information Provisions (WP 100) available at http://ec.europa.eu/justice_home/fsj/privacy/docs/wpdocs/2004/wp100_en.pdf.

[6] Erl, T.: Service-oriented architectureService-oriented architectureService-oriented architecture, Concepts, Technology and Design. In: Prentice Hall Professional Technical Reference, July 2005. ISBN 0-13-185858-0.

[7] Coulouris, G., Dollimore, J., Kindberg, T., Distributed Systems. Concepts and Design, Addison Wesley, 2005.

[8] Cabrera, L. F., Kurt, C., Web Services Architecture and Its Specifications: Essentials for Understanding WS-*, Microsoft Press, 2005.

[9] Windows Workflow Foundation (2008) http://netfx3.com/content/WFHome.aspx, 28.07.2008.

[10] OASIS Web Services Business Process Execution Language. OASIS Web Services Business Process Execution Language (WSBPEL) TC, 2.0 edition, April 2007.

[11] Openkapow http://openkapow.com/, 28.07.2008.

[12] Microsoft Popfly. http://www.popfly.com/, 28.07.2008.

[13] Oasis, Business Process Enterprise Language Version 2.0, available at: http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html, 27.02.2008.

[14] Microsoft Windows Live Services. http://my.live.com/, 28.07.2008.

[15] iGoogle. http://www.google.com/ig, 28.07.2008.

[16] My Yahoo!. http://in.my.yahoo.com/, 28.07.2008

[17] Yahoo Pipes (2008) http://pipes.yahoo.com/pipes/, 28.07.2008.

[18] Dapper (2008) http://www.dapper.net/, 28.07.2008.

[19] Serena Software (2008) http://www.serena.com/mashups/index.html , 28.07.2008.

[20] PrimeLife: Infrastructure for Trusted Content. Spitz, St., Hinz, W., Bergfeld, M. (Eds.). Public Deliverable D6.2.1 by PrimeLife Project Consortium. 2008.

[21] Bizer, J., Grimm, R., Staab, S., Meissner, S., Pähler, D., Ringelstein, C., Rost, M., Schallaböck, J., Schwagereit, F., Chancen und Risiken von Service-orientierten Architekturen in Virtuellen Organisationen, 2007, available at https://www.datenschutzzentrum.de/soa/SOAinVO-Analyse.pdf

[22] The persona of Inga is taken from the PrimeLife list of Personas. Köffel/Wästlund/Wolkerstorfer: The PrimeLife Personas, internal report, 2008, available from https://trac.ercim.org/primelife/wiki/UIRepresentation .