

Opportunities and Risks for Privacy in Service-oriented Architectures

Christoph Ringelstein, Felix Schwagereit, Daniel Pähler
University of Koblenz-Landau
56070 Koblenz, Germany
{cringel,schwagereit,tulkas}@uni-koblenz.de

June 2007

Abstract

This paper identifies privacy issues that arise from using service-oriented architectures, which ease the ad-hoc creation of complex workflows. These workflows may also integrate external services. Thus, virtual organizations emerge. Simultaneously with the complexity, the existing privacy issues increase. On the other hand, the flexibility and dynamics gained by using service-oriented architectures give the enterprises new opportunities for controllability, allowing them to reduce the risks and be in compliance with data protection principles as demanded by law and trade agreements. This paper analyzes these risks and opportunities and sketches a solution space for controllability in service-oriented architectures.

1 Introduction

Today, data privacy is an important aspect for modern enterprises because the law demands enterprises to comply with data privacy principles. In the European Union, this is manifested in 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”, which is already enacted in all member states. This directive demands the utilization of various methods to increase the customers’ possibilities to control the processing of their personal data (e.g. notification and information by request, explained below). In addition to the compliance with applicable law, another reason for enterprises to consider privacy lies in the benefit of more customer trust that can be gained by offering plausible privacy policies.

An enterprise which intends to comply with data privacy principles requires a detailed overview of all data processing workflows. The use of the paradigm of service-oriented architectures facilitates the generation of such an overview since it is a paradigm “for organizing and utilizing distributed capabilities that may be under the control of different ownership domains” [17]. To achieve this, the enterprises organize single capabilities as single services and offer them for internal and partially for external use.

Thus a service-oriented architecture eases the access to services that are provided by other enterprises and the integration of these services into internal workflows. In this way virtual organizations emerge. The term virtual organization denotes an organization that dynamically combines services of various enterprises by means of information technology [5, 21]. Virtual organizations allow enterprises to react flexibly to market needs. But this gain of flexibility also holds risks for the protection of data privacy.

The aim of this paper is to identify opportunities and risks for privacy in service-oriented architectures. For this purpose we introduce a real world use case scenario in section 2. In section 3, we identify newly emergent and increasing problems for privacy in service-oriented architectures and describe how data processing can be controlled with respect to data privacy principles. In section 4, we analyze current techniques and research approaches that yield controllability of data privacy aspects in data processing. Finally, in section 5, we present a solution space of how this controllability could be reached. In this context we analyze how existing techniques and research approaches enable or support the controllability and identify open issues.

2 Use Case – Amazon Mechanical Turk

2.1 Participating Enterprises

Apart from its mail order business, Amazon also offers several Web services, which reach from a freely available Web service that offers access to the product catalogue to several services for business customers which are completely independent from the mail order business. Among these, “Amazon Mechanical Turk” (AMT) can be found¹: Some tasks are rather difficult to solve for computers, but can be accomplished by humans with little or no effort, e.g. “Choose which of these photographs depicts best the storefront of company XYZ” or “Transcribe this spoken interview to written text”. The idea of AMT is to make use of this property. So it does not even try to solve these problems with computational means, but hands them over to humans. For these human workers, the incentive in working for AMT lies in a (rather small) reward that they are paid.

Using the AMT service is fairly easy for clients: First, they write a short description of the task in general, and how much they are willing to pay for it. Then, they can send the actual data they want processed by Amazon as a “normal” Web service call. AMT acts as an intermediate between this call and the human workers. A minimum quality of the tasks’ results can be ensured by demanding that the workers successfully complete special qualification tests beforehand.

Shortly after the launch of AMT, a company named “von Kempelen” was set up which offers translation services and is built largely on the AMT: Translations and transcriptions are not done by von Kempelen’s employees, but by AMT workers. By means of the aforementioned tests, the AMT workers who want to work for von Kempelen can achieve different qualification levels, whereas workers of higher levels can check the translation results of workers from lower levels.

In addition to AMT and von Kempelen, this use case contains a software producer that is located in the USA, but sells its products in various countries worldwide; for

¹Cf. <http://aws.amazon.com/mturk> for an explanation of the name.

the purpose of this use case, we call this software producer Zettasoft. Support requests from customers can be written in English, in which case Zettasoft’s staff can answer them directly. Besides, they want to be able to answer non-English support requests. To achieve this, von Kempelen’s services are used. For an overview of all involved parties and their relations, see figure 1.

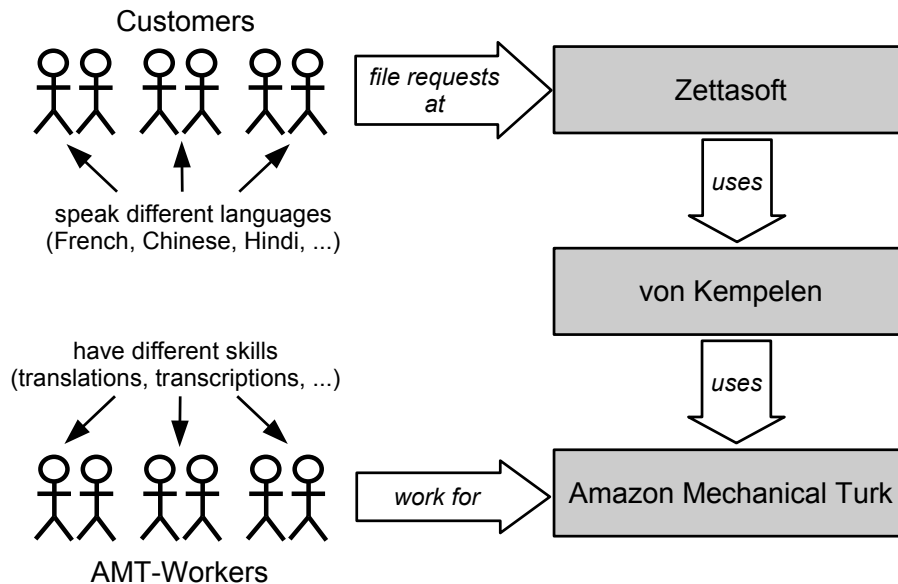


Figure 1: The five parties in the AMT use case

2.2 Workflow

In this section, we describe this use case’s workflow. For a schematic display of an exemplary workflow execution in which the support request of a Chinese customer is translated, answered, and the answer is translated back again, see figure 2.

When one of Zettasoft’s customers wants to request support, they are given the opportunity to do so not only in English, but in a variety of other languages. If they make use of this offer, the message is not read by Zettasoft’s own personnel after receiving it, but it is immediately sent to von Kempelen, with the instruction to translate it to English (the source language was also specified by the customer). Upon receiving this data, von Kempelen creates an AMT task for this translation. As soon as a sufficiently qualified worker accepts the task, the actual data is transmitted, so the worker can translate it. Afterwards, the text (which is now in English) is transmitted back to von Kempelen via AMT.

Since von Kempelen does not want to rely on the results of only one worker, it places a new task at AMT, “Check this translation and correct it, if necessary”. Again, AMT gives some human worker the data which they need (the source message and the translated text), and is at some time later given back a probably improved version of the translation to von Kempelen. This last step is done several times by von Kempelen,

until the result is a final translation the correctness of which is agreed upon by several workers.

von Kempelen hands this translation over to Zettasoft whose employees can now write a reply to the support request. Since this reply is in English, it has to be translated back to the original language the customer used, so it passes through the same process as the customer's message before, only with the languages swapped.

Finally, Zettasoft is able to send the customer a reply to his message in his preferred language.

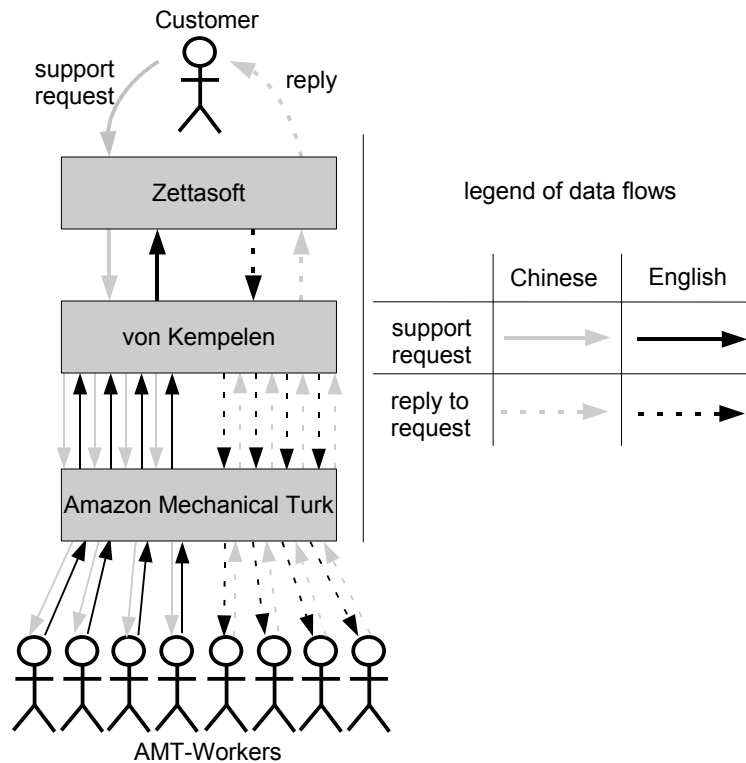


Figure 2: Data flows in an exemplary workflow execution

3 Problem Area

3.1 Privacy Issues

Data privacy is not only an issue for single enterprises which process data about natural persons. It becomes even more problematic within virtual organizations. That is due to an increasing complexity and the emergence of new problems when more actors get involved. Thus the questions arise, “which issues increase and which new issues emerge”? In the following, we take a look at some of the most prominent issues.

Increasing complexity of data flow

To comply with data protection principles and with privacy law, enterprises need an overview of their internal data flow. However, single enterprises often lack such a detailed overview because their workflows may have arbitrary levels of complexity and so have the data flows. In virtual organizations this problem increases, because the participating enterprises, in our use case Zettasoft, von Kempelen and Amazon, combine their single workflows into large, integrated workflows. In addition, this integrated workflow is less transparent because the participating enterprises do not allow each other to monitor their own workflows.

Emergent issue of multiple responsible parties

Because of the fact that a virtual organization consists of multiple enterprises, it has to be clarified which enterprise is responsible for which action performed on personal data [26]. However, only the enterprises in direct contact with the virtual organization's customers have contracts with them. Thus the contracts must include the rights and duties of the enterprises without direct customer contact. Through the involvement of multiple enterprises, in our use case these are Zettasoft, von Kempelen, and Amazon, it is not clear for the customer which enterprise is responsible for which part of a given notice or information by request. However given notices and requested information have to contain also the notices and requested information of the other partners.

In addition, to avoid restrictions on the flexibility of the virtual organizations, it must be possible to exchange enterprises within the virtual organization while data is being processed according to the existing contracts. Therefore contracts must be flexible enough to allow supplementary changes or the renegotiation of contracts has to be supported.

For instance in the AMT use case, if after notice has been given a partner, e.g. von Kempelen, is exchanged, the customer needs to be informed that now another service provider is used.

Emergent issue of internationality

The flexibility gained by the deployment of a service-oriented architecture eases the ad-hoc integration of services of enterprises which are located in foreign countries. This ability to create ad-hoc international cooperation requires special methods to assure the compliance with national laws and national data protection principles apart from the fact that all partners of a virtual organization have to comply with the contracts the virtual organization agrees on with its customers.

This issue cannot be solved by means of service-oriented architectures or other technical methods because privacy law demands that no data is transferred into countries without adequate (as defined in Directive 95/46/EC) data privacy laws. However, this issue is partially solved by international agreements between the EU and countries with, from an EU point of view, insufficient data privacy laws. For instance, the USA and the EU have developed the safe harbor framework [24] which allows US-enterprises to be in compliance with the European data protection principles after certifying their data privacy policies and practices.

In our AMT use case, for instance, Zettasoft may be an European enterprise as well as von Kempelen. If now AMT is a service offered by an American enterprise that does not comply to the safe harbor agreement von Kempelen has to assure that AMT is not used when privacy-related data are involved. This can lead to unforeseen issue especially if the content of the request, which should be translated, contains privacy-related data that is not recognizable by Zettasoft or von Kempelen, because of the foreign language.

Emergent issue of combined data collection

Standardized communication interfaces and data formats, which are fundamental for service-oriented architectures, can make data which was before separated in different enterprises more easily accessible. In this way, it might be possible to combine otherwise separate data sets and deduce new (and more personal) information from them.

For instance, in the AMT use case, Zettasoft is able to map language capabilities, certain software preferences and related problems to their customers' names. If we further consider a job broker built on AMT, where job searching requests are read and mapped to (local) job offers by human workers, new perspectives come into view: An interested third party could obtain information from Zettasoft and the job broker and automatically combine them to profiles that show who with which language and software capabilities is looking for which job.

3.2 Controllability

We identified the controllability of data processing as a key concept for virtual organizations which has to be considered when seeking solutions for the above mentioned problems. Controllability can be defined as the situation in which rights protecting interests or assets of "entities" are not affected in an unacceptable way through the existence or usage of information systems [6]. These entities can be the customers, which means that their data and, indirectly, their privacy rights are protected in a controllable system. But these entities can also be the enterprises of the virtual organization, since law-compliance is an important issue for them.

How can controllability help virtual organizations tackle the privacy induced problems? At first, we have to pay attention to the fact that privacy has a focus on the rights of natural persons concerning their personal data, grounded by the "Right of Informational Self-Determination"². In this context "notice" and "information by request" are rights, which are granted to customers, and duties, which are imposed on enterprises by law. These rights and duties lead enterprises to the need of implementing appropriate facilities and interfaces for giving notice and information by request by means of their service-oriented architecture. But in order to implement these information services to its users, an enterprise needs the ability to control its own data processing. Basic mechanisms which are able to make contributions to the solution for this need are "logging" and "policies".

²In Germany this right is based on a decision of the Federal Constitutional Court from 15-12-1983; Az.: 1 BvR 209/83; NJW 84, 419.

In the next paragraphs we give a short outline of the identified controllability related mechanisms. Additionally, we show which of the above mentioned problems they tackle.

Notice

In giving a notice the enterprise has to inform their customer about (i) the enterprise's identity, (ii) purpose of the collection, processing, or usage of personal data and (iii) possible third parties that might be granted access the customer's personal data in future.³ Since giving a notice to customers is a (usually legal) obligation for a single enterprise, virtual organizations as well have the duty of giving a notice to their customer concerning the processing of their personal data.

For the example use case giving notice means that the customer has to be told explicitly what routes to other enterprises (von Kempelen) their personal data can take, once given to Zettasoft, how long it is stored within the databases, and what persons have access to it.

Generally, notice has to be given in advance, i.e. before the data are collected. In the case of a later occurrence of a notable event, it is possible⁴ to give notice subsequently. Additionally, it should be emphasized that a notice has to be detailed and easily understandable for the customer in order to comply with the common legal requirements.

Notice has two dimensions: the judicial dimension, which is why notice has to be considered by enterprises, and the technical dimension, which focuses on the implementation of law-compliant systems.

In a virtual organization, the task of giving notice is strongly related to the above mentioned "issue of multiple responsible parties". Especially in a dynamic environment with many potential business partners and data flows, it is difficult to determine which notice has to be made to the customer. Therefore an adequate implementation of notice for all enterprises seems to be promising for clarification of privacy responsibilities within the virtual organization.

Information by Request

Obtaining information by request about their personal data is the right of natural persons to get information about (i) existence, content, and origin, (ii) recipients and (iii) the purpose of storing their personal data⁵. In contrast to notice, information by request has to be claimed actively by the customers. So the instrument of information by request is a vital component of controllability because in order to influence the data processing within the virtual organization, the customers need information to decide if their influence is necessary.

Examples for personal data the customer within the given use case might be interested in are addresses and patterns of purchase behavior which might be stored in Zettasoft's or von Kempelen's databases.

The information a person is able to get by request is often prerequisite for the exercise of further rights [26]. So as soon as the users know which of their personal

³Following the definition of notice in § 4 section 3 German Federal Data Protection Act (BDSG).

⁴E. g. in German law this is stated by § 33 section 1 Federal Data Protection Act (BDSG).

⁵Following the definition of information by request in § 34 German Federal Data Protection Act (BDSG).

data an enterprise has, they can demand erasure, locking or correction of that data from the data processing authority. Again, this legal dimension implies the technical dimension of providing means to comply with the law.

Very similar to the problems related to notice, a legally compliant implementation of information by request is a challenge which grows above average with the number of participants of the workflow. That is due to the growing number of data flows each enterprise has to consider when processing an information by request. For this reason a satisfying solution for the processing of information by request is suitable for tackling the general “issue of data flow complexity”.

Policies

A policy is a set of rules for the possible activities of the enterprise. The counterpart of policies are the preferences of the customers, which explicate the activities the potential customer allows the enterprise to do for fulfilling a contract. Policies can be stated for all actions generically, but also for each individual business transaction separately.

In the given use case there may exist a general privacy policy from Zettasoft stated to their customers including the assurance not to sell their addresses to other enterprises for advertisement.

From a technical point of view, a policy has similarity with constraints used in software engineering. In this discipline, constraints are defined as limitations of possible states, behaviors or the nature of elements [22]. Transferred to the privacy context, this means an enterprise’s binding explication of the ways it will handle personal data given to it during a business transaction. So policies in virtual organizations can be seen as an a-priori controllability measure that can help to ensure the compliance of an enterprise with the legal requirements. However, by stating a policy it ought to be assured that only necessary information is provided to the customer so that corporate secrets are protected.

Formalizing and standardizing policies within the virtual organization can make it easier for enterprises to interpret the policies of other enterprises. Therefore, an enterprise can achieve a more automatized handling and generating of the policies it has to state to its customers. So an increasing level of sophistication of policies can help address the above mentioned “issue of multiple responsible parties”. Furthermore, adequate policies might be a way to ensure that business partners from countries with no privacy law comply with the privacy demands of the customer. This can be achieved by including minimal policies given to the customer to the contracts with enterprises along the work flow. Hence policies are a suitable mechanism for reducing the “issue of internationality”.

Logging

The purpose of logging, in the sense of creating and filling log files, is to enable the reconstruction of behavior afterwards (cf. [14] for logging in the database context). So logging is used in many application areas like finding malfunctions in complex information systems or ensuring consistency of databases; as a counterpart to policies, logging serves as an a-posteriori controllability measure. To allow the reconstruction of a certain behavior, logging has to be the process of recording activities of special entities and the time or context they occur.

In our AMT use case for example, AMT logs the translation tasks given to and the translations retrieved from the AMT workers. This is mainly done in order to calculate the wages AMT has to pay to their workers.

Logging is privacy related from at least two points of view. Firstly, the log as an objective record can allow afterward reconstruction of data flows, even from unforeseen perspectives. This helps in generating information by request, but also in tracing violations of privacy law or contracts. For this reason an enterprise is legally obliged to log processes like system logins or access to sensitive data [16]. Secondly, the log files themselves may contain personal data and can so be subject of access restriction.

Because of the property of enabling the reconstruction of behavior, logging can help in analyzing and therefore understanding data flows within the virtual organization. Thus, logging is an adequate instrument to tackle the issue of “data flow complexity”. Furthermore logging can help to avoid occurrences of “combined data collection”, because by the possibility of securing evidences through logging the enterprises are sensitized of this privacy issue.

4 Current state analysis

4.1 Notice and Policies

In the shown use case (cf. section 2) and in reality, the consumer accesses a service offered in the Web by means of a Web browser and not via a Web service interface. The notice for the support service has to be given by Zettasoft to their customer before the request is made. For fulfilling their duty of giving notice, Zettasoft states a privacy policy. For expressing this privacy policy, they have the options of giving the end user a text in natural language or making policy statements in a formal, standardized format. Since Zettasoft is a part of the virtual organization (consisting of Zettasoft, von Kempelen, AMT, and the AMT workers), it has to be assured that the given notice matches reality. In particular, the privacy policies of Zettasoft, von Kempelen, and AMT must match. Moreover, the combination of their policies must be transparent to the end-users, and they must not open “covert channels” for misuse. The combination of privacy policies is, apart from very simple examples, an open field of research.

Text form is the way used by the majority of enterprises to give privacy notices to their customers [12]. The text of the privacy policy is published on the Web page and referred to during the order process. So it becomes an integral part of the contract between enterprise and customer. Typically, a notice in text form comprises approximately 2000 words [18] and is hard to understand for a person without special legal knowledge. This can be explained by the fact that these privacy policies are written and constantly improved by judicially educated personal in order to comply with the latest privacy regulation and to protect the enterprise legally.

The P3P (Platform for Privacy Preferences) standard [27] from the W3C (World Wide Web Consortium) in its current draft version 1.2 is the most prominent effort to standardize privacy expressions for customers formally. P3P is based on XML and contains a set of “uses, recipients, data categories, and other privacy disclosures” allowing the enterprise to encode a privacy notice. To “read” the notice, the customer needs a software tool which can evaluate the retrieved policy against the customer’s own privacy preferences. Typically, the Web browser has this functionality built in or

can be upgraded with a plug-in, but the current P3P implementations in Web browsers are still considered insufficient by the P3P Working Group ⁶. Nevertheless it can be observed that an increasing amount of business Web sites offers P3P-policies [9].

In the example use case, privacy policies exist for all business relations between the connected enterprises of the virtual organization. In particular, the privacy policy of AMT is stated in natural language and included in the “Web Service Licensing Agreement” which is located at the Amazon Web site ⁷. Even though policies between the enterprises should correspond to the notice given to the customer, the privacy policies are created manually and independent from each other. Besides policies which are specified as natural text and therefore independent from any technical implementations, there exist standardization approaches and research projects which aim to formalize privacy policies. Below, we give an overview of the more prominent candidates, which can be used within a virtual organization’s service-oriented architecture. These standards or standardization approaches have in common that they are based on top of XML.

- The Enterprise Privacy Authorization Language (EPAL) in its current version 1.2 [23] was submitted to the W3C by IBM in 2003, but it never became an official standard. The goal of EPAL is to provide a formal language for enterprise privacy policies. These policies consist of rules which allow or deny actions on defined data categories. While P3P describes policies for publications, EPAL describes rules and conditions which govern privacy related processes within organizations.
- The Web Services Policy Language (WSPL) [20] is based on XACML (eXtensible Access Control Markup Language) [19], which allows the definition of access control, authorisation, and privacy policies. WSPL as a subset of XACML is intended for giving a formalism to state privacy policies for Web services. Similar to EPAL, the policies stated in WSPL consist of rules. But in comparison to EPAL, WSPL provides more functionality [1]. The current status of WSPL is that of a working draft mainly propagated by SUN.
- WS-Policy [3] is a framework allowing to formulate policies of a Web service. It can be extended with assertions indicating domain specific semantics. One proposal for a privacy relevant extension of WS-Policy is WS-PolicyConstraints [2].

All of these standardization approaches have been in development for years. Nevertheless, current workshops of researchers and practitioners ⁸ show that none of these standardization approaches has got the maturity or wide acceptance needed to ensure a guaranteed practical interoperability of information systems within a virtual organization.

Beside the above mentioned standards which are advanced by the industry, there exist research projects aiming to develop ways of formulating privacy policies. Most of

⁶<http://www.w3.org/P3P/>

⁷<http://www.amazon.com/AWS-License-home-page-Money/b?ie=UTF8&node=3440661>

⁸e.g. the W3C Workshop on Languages for Privacy Policy Negotiation and Semantics-Driven Enforcement <http://www.w3.org/2006/07/privacy-ws/> or the WS-Policy interoperability workshop in Wall-dorf <https://weblogs.sdn.sap.com/pub/wlg/3616>.

these approaches use semantic formalisms like the Web Ontology Language (OWL). Based on these formalisms there are ontologies defined which allow automated reasoning about the implications of a given policy for a given privacy relevant action. Examples for this research are Rei [13] or KAOs [25].

4.2 Information by Request and Logging

As was already mentioned, each person has the right to view all data related to them which an enterprise holds, but this requires their own initiative. Traditionally, this required the person to find out the enterprise's address, ideally even the name of the enterprise's responsible data protection officer. Next, they would have to express their desire to view "their" data in a letter, referring to concrete laws and paragraphs to point out their rights in case the enterprise was still ignorant in this respect. As it turns out, the Internet and related technologies, while becoming an increasingly large part of everyday life, have done little to improve the situation. Customers can now use other channels to contact the enterprise, but there is still no easy and standardized way for them to express their wishes.

As Weichert points out [26], new problems have arisen in today's distributed systems: if a person's data is spread between several parties in a distributed system (such as a virtual organization), it becomes difficult for this person to keep an overview of which data is used where for what purpose. Since each party is only responsible for giving the customer information about the data that they themselves have and the direct partners they gave it to, the task of finding and contacting each of these parties can be extremely tedious and time-consuming.

The fact that the involved enterprises in a virtual organization are not legally required to keep an overview of the organization's structure shifts the information problem to the customers. For the use case from section 2 this means that the customer would have to ask all three involved enterprises (Zettasoft, von Kempelen and AMT) separately, and would then have to combine the answers in their context. As an added problem, there would be an "informational gap" since the customers' messages were handed over to individuals, who do not keep logs of their actions.

In section 3.2 it was already pointed out that log messages can be an adequate means to generate detailed replies to users' requests for their personal information. Most software systems have logging capabilities in one form or another, reaching from short, system-related messages used e.g. by UNIX systems to arbitrarily long application-specific messages in modern software. While automatic log file processing as well as central logging in distributed systems have already been possible for a while⁹, some crucial problems remain:

- Each enterprise logs their own data. In a virtual organization, this means that logs concerning a single customer's data are scattered throughout the organization.
- High-level information can get lost in log files. While the system logs might be perfectly appropriate for administrative reasons within the enterprise, they can lack the necessary context information that would allow for a reconstruction of

⁹This is due to standardized log file formats such as the W3C's "Extended Log File Format" [11] and logging software such as "syslog", respectively.

the process a set of user data has undergone. High-level logging, on the other hand, still lacks well-supported standard formats [28].

- Security is still an issue. This includes e.g. the obvious fact that a system's logging instance has to be protected from manipulation, be it accidental or malevolent, the requirement that the source of a log message and the logging instance verify each other's identities, and many more. Since there is still no single, unified logging mechanism for all needed purposes, combinations of several methods are common practice [28], which threatens the system's controllability in a sense that administrators might not be able to protect it as thorough as necessary.

5 Solution Space

The question is how far the current technologies and research approaches are able to solve the abovementioned privacy issues. For this purpose, we present an approach which solves these privacy issues and allows a higher degree of automation in order to reduce management efforts. Furthermore, we define the requirements for this approach.

As shown above, various approaches exist to tackle different aspects, like notice or logging. However, all these approaches only provide solutions for some of those aspects. Due to dependencies between policies, preferences, and notice, between notice and information by request, as well as between information by request and logging we suggest one integrated approach for a machine-interpretable formalization that uses a well-defined shared ontology. A machine-interpretable formalism enables the semi-automated processing of most tasks and the complete automation of some simple tasks, which leads to a reduction of complexity. The ontology, which should be used by all involved parties, facilitates a common understanding of terms and expressions related to privacy and the service's domain. Thus, translation efforts are reduced. In the following, we give a short overview of which privacy related tasks can be automated or at least semi-automated and to what extent this is possible with existing technologies.

Policy/Preference Comparison

The customers' preferences describe what the customers allow to be done with their data. Specifying the policies and the preferences by means of the above postulated machine-interpretable formalism enables the semi-automated comparison of both. Thus, in combination with an ontology, human intervention is only needed if a conflict occurs during comparison that cannot be solved automatically. In our AMT use case, the customer may have the preference that his personal data, like address etc, is not transferred by Zettasoft to other parties. An automated comparison allows the customer to not check the policy by himself.

Presently, there exist different approaches that fulfill this requirement, e.g. WSPL and Rei, but these approaches are neither widely used to describe real world Web services, nor are they designed to express logs or information by request. Thus, we identify the extension of policy description languages to describe logs and information by request as an open issue.

Policy Combination

Policies are needed to give notice about what action the virtual organization will do with submitted personal data for which purpose. To give notice, a virtual organization has to combine the policies of all involved enterprises. This combination can be achieved by means of the above postulated formalism and ontology. In addition, the application of these methods allow the semi-automation of the combination task. In the AMT use case Zettasoft has to give notice that it passes the content of the request to a translation company. Von Kempelen has to give notice that it passes the fragmented content to AMT. Thus in combination Zettasoft has to give notice that the content of the request is passed to von Kempelen that passes fragments of the content to AMT. The postulated machine-interpretable formalism allows Zettasoft to do this semi-automatically. Only if policies occur that are not combinable by a machine¹⁰, a human employee of Zettasoft has to formalize the combined policy by hand.

Some of the above mentioned approaches, like WSPL, already allow the combination of policies from different parties to one joined policy. If these approaches are extended to allow the specification of logs and information by request, they would also meet the need of combinable policies.

Versatile Logging Standard

Logging is required to collect information about what is done with the customers' personal data. From the log files, the requested information is mined. The automation of data mining and the generation of a response to the request for information can be achieved through a combination of the following requirements: a machine-interpretable logging formalism, a mechanism tailored for logging privacy-relevant actions, and a machine-interpretable formalism for the information given upon request. In addition, if the customer's request is also formalized by means of the postulated machine-interpretable formalism and ontology, the whole request process can at least be semi-automated. In our use case the customer may request information about the processing of its data by Zettasoft. A ontology-based logging mechanism would enable Zettasoft to automatically mine its logs for all related information automatically.

Even if some applications allow the creation of formalized logs that uses ontologies, these ontologies are only defined for the application's domain [28] and are not standardized. In addition, these approaches are not designed to be used to express policies, either. Thus, we identify the definition of a logging standard that allows application-independent logging by means of ontologies for individual domains as an open issue. In addition, this logging mechanism should be compatible with the formalism used to specify policies.

Log Accessibility

To enable the mining of log files, it is vitally important that the enterprise which is responsible for giving information by request can access all relevant log data. This can be facilitated by different methods that provide a "single point of access", i.e. a single entity that keeps and manages all log data. Central logging is one of these methods:

¹⁰Machine-combinable in the context of a formalism means the ability of a reasoner to derive one combined statements from two single statements.

all enterprises of a virtual organization send their log data to one logging server, where they can be accessed afterwards. While technical solutions for this method do exist, as was already mentioned, a grave problem remains: since virtual organizations are built in an ad-hoc fashion and can break up just as quickly, it is difficult to find this central logging server, and considering that all involved enterprises have to agree with this choice, this method might severely limit the organization's flexibility. An alternative method is the usage of "sticky logs": Whenever log information occurs, it is attached to the actually processed data as metadata and "travels" with the data in the further workflow execution. Consequently, all information about the processing of this data by all partners is connected to the data and thus directly accessible at a single place. Furthermore, only logs relevant to the processing of this data are included. A disadvantage of this approach is that the logs are accessible and manipulable wherever the data is, but this can be solved by means of encryption and digital signatures.

Because of the advantages of data logs that are directly attached to the data, we recommend this as a solution. Thus, the development of a logging mechanism that implements sticky logs is identified as an open issue. The required mechanisms for encryption and digital signatures already exist with the W3C's XML-Signature [8] and XML-Encryption [7].

Log Sustainability

To be able to grant information by request, enterprises are legally required to keep their logs for a certain period of time. In virtual organizations that rely on distributed logs (i.e., each enterprise keep its own log files), information could partially get lost when an enterprise leaves the organization, making the reconstruction of certain processes difficult or even impossible.

We propose sticky logs as a means of fulfilling this requirement, too. Since it is the customer who induces the workflow execution in a virtual organization, the enterprise that has direct contact with the customer is also the last enterprise that the sticky logs pass through before the workflow's result is returned to the customer. This enterprise can later generate exhaustive information when requested by its customers, even if the virtual organization does not exist anymore.

Standardizing Information by Request

The information that a virtual organization gives by request is about the privacy-related actions of all enterprises. Thus, as for policies, to give information by request, the virtual organization has to combine the responses of all involved enterprises. This combination can also be semi-automated by means of the above postulated formalism and ontology. In the AMT use case the customer may request information about the processing of its data by Zettasoft. Like for notice, a machine-interpretable formalism allows the automated combination of the information given by different enterprises. This also includes the need of human interaction for not machine-combinable informations.

As mentioned above, to specify responses to requests for information, no formalism based on ontologies for individual domains exists. Thus, we identify the definition of such a formalism that allows the combination of the responses of multiple

enterprises into one response as an open issue. The basic technologies for this formalism already exist and are used by ontology-based formalisms to specify policies, like WSPL and Rei.

Audit

To check the adherence to a given notice, both notice and information by request are required. If the policies used to give notices and the responses used to give information by requests are specified using the same formalism, and if this formalism also uses an ontology that is accepted by both the customer and the virtual organization, the adherence can be checked at least semi-automated. In our AMT use case this formalism allows the user to automate the adherence check as long as no conflict occurs.

Because no approach exists to formalize information by request in the proposed manner and such formalism is required for the semi-automation of the adherence check, we redefine the issue of defining a formalism for specifying information by request in a way that this formalism also supports the semi-automated comparison of policies and responses.

Data Minimization and Separation of Information Duties

Like single enterprises, virtual organizations have to separate information duties for privacy-related data as well as reduce the use of those data to the indispensable minimum. This is of importance for all operations performed on or with the data, like processing, storing, and logging as well as passing the data to other partners within in the virtual organization. Thus, we propose that the data is annotated with metadata describing for which purposes the data may be used and of which type the data is. By means of the abovementioned ontology-based description formalisms it is possible to semi-automatically identify which actions are allowed on which data and for which purpose they may be processed, stored, logged, or passed to another party. In our AMT use case, for example, there is no need that Zettasoft transmits names, addresses, and payment info of its customers to von Kempelen, even if Zettasoft needs names and addresses of its customers for its own business purpose. Vice versa, Zettasoft needs no knowledge about the individual human workers of AMT.

For the formalization of preferences approaches exist, e. g. APPEL [4]. In addition, there are techniques, like RDF [15], that allow the annotation of data with arbitrary metadata. However, we identify the specification of a formalism which is tailored to support data minimization and separation of information duties, as an open issue.

6 Conclusion

Virtual organizations are a form of collaboration between enterprises which will gain more influence in the future. One precondition is the application of the service-oriented architecture paradigm for information systems which are based on well defined standards to facilitate automated communication between participating enterprises. Although dynamic virtual organizations seem to be a proper means to increase

competitive capability, the compliance to privacy standards is nevertheless a requirement which must not be neglected. So the challenge emerges to cope with privacy related problems which arise or increase through the usage of service-oriented architectures. The problems we found comprise the increasing complexity of data flows, the multiple responsible parties, the emergent issue of internationality, and the emergent issue of combined data collection.

We identified controllability of data processing as a key concept which has to be considered in this context. In order to operationalize controllability, the mechanisms notice, information by request, policies and logging exist. After analyzing the current state of these mechanisms, we identified shortcomings in existing standards and their application. Therefore, risks emerging from incompliance of the implementation of the mechanisms with legal obligations may hinder further facilitation of virtual organizations. In order to show options for further development of controllability mechanisms, we span a solution space containing various possible directions.

The proposed mechanisms will be implemented in succeeding work. This will also include additional problem areas such as data minimization and separation of information duties. Pseudonyms and element-based encryption are good candidates for solutions to these problems.

By proactively tackling the mentioned privacy related problems, the enterprises of virtual organizations do not only have the opportunity of gaining more controllability for their information systems in general. Furthermore, they can give more controllability to their customers in order to get competitive advantages.

7 Acknowledgement

We thank Rüdiger Grimm and Steffen Staab from the University of Koblenz for their feedback on the draft version of this paper and Sebastian Meissner and Martin Rost from the Independent Centre for Privacy Protection Schleswig-Holstein for discussing various privacy related issues.

This work has been partially supported by the Federal Ministry for Education and Research in the project SOAinVO - "Technique Analysis and Risk Management for Service-oriented Architectures in Virtual Organisations".

References

- [1] A. H. Anderson, A comparison of two privacy policy languages: EPAL and XACML, SWS '06: Proceedings of the 3rd ACM workshop on Secure web services, Alexandria, Virginia, USA, ACM Press, New York, 2006, pp. 53-60.
- [2] A. H. Anderson, Domain-Independent, Composable Web Services Policy Assertions, Seventh IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'06), 2006, pp. 149-152.
- [3] S. Bajaj et. al., Web Services Policy 1.2 - Framework (WS-Policy), <http://www.w3.org/Submission/2006/SUBM-WS-Policy-20060425/>, retrieved May 2007.

-
- [4] L. Cranor, M. Langheinrich, and M. Marchiori, A P3P Preference Exchange Language 1.0 (APPEL1.0), <http://www.w3.org/TR/2002/WD-P3P-preferences-20020415>, retrieved June 2007
 - [5] W. Davidow, and M. Malone, *The Virtual Corporation*, HarperCollins, New York, 1992
 - [6] R. Dierstein, Sicherheit in der Informationstechnik - der Begriff IT-Sicherheit, In: *Informatik-Spektrum*, 2004 (4), pp. 343-353.
 - [7] D. Eastlake, J. Reagle (eds.), XML Encryption Syntax and Processing, <http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/>, retrieved May 2007
 - [8] D. Eastlake, J. Reagle, D.Solo (eds.), XML-Signature Syntax and Processing, <http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/s>, retrieved May 2007
 - [9] S. Egelman, L. Faith Cranor, A. Chowdhury, An analysis of P3P-enabled web sites among top-20 search results, ICEC '06: Proceedings of the 8th international conference on Electronic commerce, pp. 197-207.
 - [10] P. B. Gove (ed.), *Webster's third new international dictionary of the English language unabridged*, Merriam-Webster, Springfield, Mass., 1993.
 - [11] P. M. Hallam-Baker, B. Behlendorf, Extended Log File Format, <http://www.w3.org/TR/WD-logfile.html>, retrieved May 2007
 - [12] C. Jensen, C. Potts, Privacy policies as decision-making tools: an evaluation of online privacy notices, Proceedings of the SIGCHI conference on Human factors in computing systems 2005, pp. 471-478.
 - [13] L. Kagal, T. Finin, A. Joshi, A Policy Language for a Pervasive Computing Environment, In: *POLICY '03: Proceedings of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks*, IEEE Computer Society, Washington DC, 2003, pp. 63-74.
 - [14] A. Kemper, A. Eickler, *Datenbanksysteme: Eine Einführung*, Oldenbourg, 2005.
 - [15] G. Klyne, and J. J. Carroll (eds.), Resource Description Framework (RDF): Concepts and Abstract Syntax <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>, retrieved June 2007
 - [16] N. Leopold, Protokollierung und Mitarbeiterdatenschutz, In: *DuD: Datenschutz und Datensicherheit*, 2006 (5), pp. 274-276.
 - [17] C. M. MacKenzie, K. Laskey, F. McCabe, P. F. Brown, R. Metz, and B. A. Hamilton (eds.), Reference Model for Service Oriented Architecture 1.0, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm, retrieved May 2007
 - [18] G. R. Milne, M. J. Culnan, H. Greene, A Longitudinal Assessment of Online Privacy Notice Readability, In: *Journal of Public Policy & Marketing*, 2006 (25) 2.
 - [19] T. Moses (ed.), eXtensible Access Control Markup Language (XACML) Version 2.0, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf, retrieved May 2007.

- [20] T. Moses (ed.), XACML profile for Web Services, Working draft 04, <http://www.oasis-open.org/committees/download.php/3661/draft-xacml-wspl-04.pdf>, retrieved May 2007.
- [21] R. Nagel, and R. Dove, 21st Century Manufacturing Enterprise Strategy, Lehigh, Pa.: Iacocca Institute of Lehigh University, 1991
- [22] B. Oestereich, Objektorientierte Softwareentwicklung: Analyse und Design mit der UML 2.0, Oldenbourg, München, 2004.
- [23] C. Powers, M. Schunter (eds.), Enterprise Privacy Authorization Language (EPAL 1.2), <http://www.w3.org/Submission/2003/SUBM-EPAL-20031110/>, retrieved May 2007.
- [24] Safe Harbor Privacy Principles, Issued by the U.S. Department of Commerce on July 21, 2000
- [25] A. Uszok et. al., KAoS policy and domain services: toward a description-logic approach to policy representation, deconfliction, and enforcement, In: Proc. of the 4th IEEE Intl Workshop on Policies for Distributed Systems and Networks, Los Alamitos, 2003, pp. 93-96.
- [26] T. Weichert, Auskunftsanspruch in verteilten Systemen, In: DuD: Datenschutz und Datensicherheit, 30 (2006), pp. 694-699.
- [27] R. Wenning, M. Schunter (eds.), The Platform for Privacy Preferences 1.1 (P3P1.1) Specification, <http://www.w3.org/TR/2006/NOTE-P3P11-20061113/>, retrieved May 2007
- [28] S. D. Wolthusen, Revisions sichere Protokollierung in Standardbetriebssystemen, In: Datenschutz und Datensicherheit, 2006 (5), pp. 281-284.