Towards Automated Processing of the Right of Access in Inter-Organizational Web Service Compositions

Ralph Herkenhöner*
Hermann de Meer*
Computer Networks and Communications,
University of Passau, Germany
{rhk|demeer}@fim.uni-passau.de

Meiko Jensen
Horst Görtz Institute for IT Security,
Ruhr University Bochum, Germany
Meiko.Jensen@rub.de

Henrich C. Pöhls
Chair of IT Security,
University of Passau, Germany
hp@sec.uni-passau.de

***Abstract*—Enforcing the right of access to personal data usually is a long-running process between a data subject and an organization that processes personal data. As of today, this task is commonly realized using a manual process based on postal communication or personal attendance and ends up conflicting with trade secret protection.**

In this paper, we present an automated architecture to enable exercising the right of access in the domain of inter-organizational business processes based on Web Services technology. Deriving its requirements from the legal, economical, and technical obligations, we show the architecture's overall approach solving the conflict between trade secret and exercising the right of access.

I. Introduction

According to legal obligations, every company that processes *personal data* must provide the *data subject* with means to query the company about what information is stored, processed, and exchanged with other companies. This *right of access* is usually provided as a manual process, requiring the data subject to place his request in person or via postal mail. Additionally, such a request will only cover the personal data regarding a certain company. In times of inter-organizational business processes and worldwide supply chains, this implies that the data subject has to query every company within such a business process in order to determine the full processing path his personal data took.

In contrast, a majority of inter-organizational business processes of today is realized using technical infrastructures like Web Services. It would be reasonable to provide information on processing of personal data according to the right of access as a dedicated Web Service itself. This way, the providing can be performed automatically, hence saving operational costs and enabling a company's customer to easily monitor the processing of his personal data. To the best of our knowledge, there exists no approach built on Web Services to fulfill the legal requirements originating from data protection law in an automated way. However, a business process involving any data about customers will most likely result in a Web Service processing *personal data*. This is covered by data protection laws. In this paper, we take the European data protection law codified in EU Directive 95/46/EC [1] as an example. In this context, *personal data* "shall mean any information relating to an identified or identifiable natural person (*data subject*)" [1].

In a common business process, the customer is the data subject and gives his personal data to a primary Web Service. The primary Web Service then forwards it—fully or partially—to its successors, and so forth. Each company that is given any part of that data—which is still considered personal—becomes a *controller*. A controller is required by law to grant the data subject the *right of access* among others. The right of access allows the data subject to determine which *categories of data* are being processed and the *purpose of the processing*. According to law, the controller's answer should also include the *sources* and the *recipients* of the personal data if forwarded to successors. We call this process *providing of information*. However, this poses a problem for real-world business processes, since a company might consider business partners or customers a trade secret. To protect such trade secrets, a controller can reply with *categories of recipients* rather than identifying them directly. In complex business processes, such categorized answers lead the data subject's request to a dead end, as preceding and succeeding processing of personal data and involved controllers remains hidden. This is a major shortcoming in the existing practice of information providing.

Based on these observations, we identified the following **problems**: A business process, realized as an inter-organizational service composition, does not automatically allow the data subject to exercise his right of access while preserving trade secrets of the participants. In current practice, requests of the data subject are handled by manual investigation and sending paper letters. This implies four major challenges to address:

- **Cost-efficient scalability**: A company taking part in a service composition has no efficient automated way to answer the data subject's requests. If business transactions

*The research of R. H. and H. d. M. has been partially supported by the project inSel (Funding-Id: 01|S08016B) of the Federal Ministry of Education and Research (BMBF) of the Federal Republic of Germany, and by the EuroNF network of excellence (IST 216366) of the European Union.

increase, Web Services are able to scale, but manual processes involving human agents answering the requests do not. Companies have to find means to comply with legal requirements with minimal operational costs.
- **Authentication of requests**: An answer must only be given to a request from an authorized data subject.
- **Protecting trade secrets while answering requests**: The answer to a request on processing of personal data may violate trade secrets. Giving this information would allow to reconstruct all companies involved in a business process along with their position within it.
- **Confidential information providing**: The answer has to be delivered exclusively to the data subject, to prevent illegal disclosure of personal data.

In order to provide a **solution** for these issues we propose an extension of existing *service-oriented architectures* (SOA)—based on Web Services technology—that implements the right of access given by the European data protection law. We show that our approach solves all of the presented challenges in the best possible way.

In order to understand the problems behind inter-organizational business processes and the right of access, we refine the legal, business, and technical requirements and constraints in Section II. In Section III, we present the overall approach, showing its interactions, message formats, and security-relevant details. We discuss how our solution technically fulfills legal and business obligations in Section IV. Finally, we conclude in Section V.

II. Requirement Analysis

To develop a full automation of providing information according to the right of access, we investigate all of its requirements and influencing factors in the following section.

A. Legal Obligations

In the European Union, the right of access is legally based on the Directive 95/46/EC [1]. Its intention is to guarantee any person the right of access to obtain information about the processing of his personal data. This right has to be exercised that neither it "adversely affect[s] trade secrets" nor protection of trade secrets results "in refusing all information" ([1], Recital 41).

In any case of collection of data from the data subject, the controller has to provide the data subject with at least the identity of the controller and of his representative, the purposes of the processing, and the recipients or categories of recipients of the data ([1], Art. 10). This is usually stated in a company's privacy policy. The data subject can get further information by exercising his right of access. Then the controller—in accordance with Art. 12 [1]—has to confirm whether or not data of the data subject is being processed, and if yes, he has to provide information as to

- purposes of the processing,
- categories of data concerned,
- recipients or categories of recipients,
- sources of the data,
- data undergoing processing (e.g. adress), and
- logic involved (e.g. scoring) in any automatic processing.

To protect trade secrets or other vital interests of the controller or its business partners, the controller may refuse to provide the information on data undergoing processing and logic involved in any automatic processing. Alternatively, he is allowed to reduce the provided information to an abstract level. For the same reason, he is also allowed to anonymize the recipients and state them by category.

B. Business Obligations

For a businesses entity, called *company*, major business requirements are cost effectiveness and protection of its trade secrets (see e.g. [2]). Of course, the company also has to operate within legal boundaries.

While legal compliance can be reached by having manual processes to comply with Directive 95/46/EC, such an architecture does not scale and will not be cost-efficient if the number of customers exercising their right of access increases. In particular, the data subject could normally not be billed for this service. To keep the operational costs low, most of the requests should be answered automatically.

To stay in business, a company has to protect its trade secrets, i.e. internal workings, customers, and subcontractors. If a group of companies is working together to provide a certain functionality, they don't have to know about each other being involved. Thus, the *overall view* of the complete business process is not necessarily known to all of the process' participants. While each company strives to protect their trade secrets, a more generalized *abstract view* might be tolerable, i.e. by hiding the customers' and subcontractors' identities.

C. Technical Foundations

A common architecture for realizing inter-organizational business processes is based on the *paradigm of service-orientation* (Service-Oriented Architectures, SOA [3]). This paradigm defines that every *business unit* (e.g. a company) within a business process provides its functionality via an explicit *service interface* that contains all data necessary to use its functionality. Hence, every service within a business process has its *service provider* and one or more *service users* that integrate its functionality within their own applications, which again are provided as services. However, a service provider is not required (and as stated above not supposed) to know details about the functionality of its service users (*predecessors*). Vice versa, a service user may not know about the internal operations of a service, e.g. whether it again relies on further services (*successors*) or provides its functionality all by itself. The most widespread technical realization for SOAs are Web Services. Their specifications [4] cover all aspects of providing and using a service via communication networks, such as service descriptions (WSDL, WS-Policy), message formats (SOAP, REST), or non-functional properties (WS-Security, WS-Trust). For *service composition*, the WS-BPEL specification [5] can be used for a full description of Web-Services-based business processes. It covers all aspects

of service usage, service providing, and simple workflow functionality.

Within BPEL-described business processes, a fundamental concept that will be necessary to understand the remainder of this paper is the mechanism of *correlation sets*. These sets are defined in the BPEL language and can be used to uniquely identify BPEL process instances during their execution and after termination. The underlying observation is that for every two business partners exchanging business process data there usually exists a mechanism for business process instance identification. Common approaches for this identification are e.g. assigning unique ID values or using a unique subset of the data itself (e.g. a customer's first name, last name and date of birth). Hence, the overall BPEL process instance can be uniquely identified using a full identifier of any of the communication identifiers between the BPEL process and any one of its partners. A correlation set is defined as the union of all unique identifiers used between the BPEL process and all of its communication partners.

For the scope of this paper, we will reduce the use of these correlation sets as follows. We assume that every communication of a BPEL-based business process is associated to exactly one correlation set that always provides appropriate full identifier for all of its incoming and outgoing communications. Then, every unique identifier from every communication partner can be traced back to all of the other communications that occurred during the execution of the same BPEL process instance. We further assume the unique identifiers to contain a sufficient amount of entropy to provide a minimum level of randomness, so that there is no efficient approach to "guess" these identifiers if they are not known a priori.

III. Right of Access as a Web Service

Based on the requirements analysis from the previous section, this section introduces a full solution to the stated problems. The rationale behind certain design decisions are provided in the discussion in Section IV. In the following we will first give a high level description of interactions, then we will give implementation details like message format, and finally talk about the underlying security design and assumptions.

A. Technical Architecture

The proposed architecture to automate the providing of information as to the processing of personal data is based on the assumption of an underlying web of service-oriented business processes. Thus, every personal data of a data subject has once been processed by business partners interacting according to a defined service composition. When exercising his right of access, a data subject may address any of those business partners. This initial request, which we call "Right of Access:Request"(ROAR), is directed towards a company within the service composition. The result of such a request will be a full report on that company's processing of personal data of the data subject. We call this report the ROAR response. Besides the ROAR request, a company may provide the additional service to ask its business partners about personal data of the data subject, insofar they were involved in the same business process. Hence, a company may request personal data of the data subject from its adjacent business partners, which will also be added to the ROAR response. This type of request, which we call "Right of Access:Delegated" (ROAD), is only triggered by a ROAR request or a previous ROAD request. It is carried out between business partners that know and trust each other, and only if both companies are involved in a business process that previously processed personal data of the requesting data subject. As with the original business processes itself, the ROAR and ROAD requests and responses are realized using Web Services technologies. Hence, every company provides a dedicated additional service interface for requesting information. This service interface is either embedded within the service's description itself, or realized as an independent, company-wide business unit, dedicated to the sole purpose of processing ROAR and ROAD queries. Though intentionally being rather alike, the two request types of ROAR and ROAD differ slightly, in particular regarding their message formats and security requirements. We will give a description of each in the following.

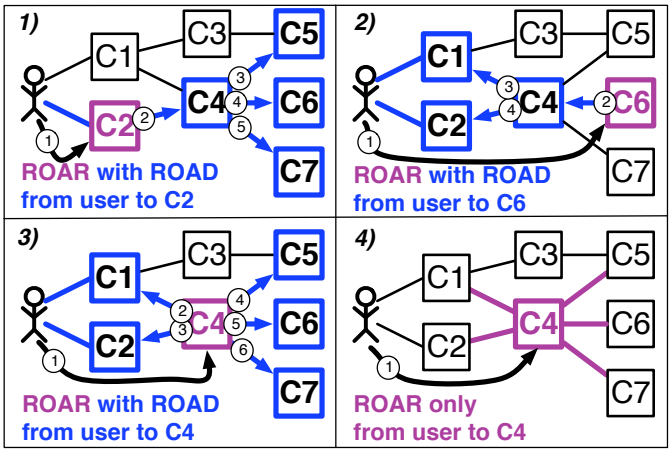

Fig. 1. Flow of requests: ROAR requests can induce further ROAD requests

1) Providing information to the Data Subject (ROAR): In this interaction, the request originates at the data subject and is directed to any controller within the service composition. This may be the company the data subject once directly interacted with (e.g. during a purchase), but may also be an arbitrary company the data subject came across for a certain purpose. To exercise his right of access, the data subject has to create a ROAR request message according to the ROAR's Web Service description (given in WSDL [6]), and send that message to the particular company's ROAR service endpoint. To provide the answer, the controller has to authenticate the request, collect all necessary information—including all information about sources and recipients of personal data—and send it back to the data subject. Thus, with a ROAR response the data subject gets information regarding the processing of his

personal data as far as the queried company was involved. We call this the *local view*.

2) Querying Information on Behalf of the Data Subject (ROAD): Typically, a data subject is not satisfied with the local view provided by a single company, if he wants to assess the impact of data processing in a whole business process. As an example, in Figure 1 case 4, the ROAR response of C4 might among other information state that a certain personal data item, e.g. an email address, was received from C1 and forwarded to C7. A typical continuation for the data subject would be to query C1 and C7 to determine the processing of the personal data in the whole business process. However, this approach would result in the data subject having to wade through many ROAR responses and reconstruct the internal linkings by himself.

To avoid this, a company wants to provide the data subject with the additional service to query its adjacent business partners *on behalf of the data subject*. If a company has passed personal data to its business partners, it has to query them first in order to provide an enhanced ROAR response. In that case, the company places a bunch of new requests to all of its adjacent business partners involved in that particular business process. We call this second type of request "Right of Access:Delegated" (ROAD), since the data subject delegates its right of access to a company that then acts on behalf of the data subject.

A company queried by a ROAD request spawns a number of own ROAD requests, aggregates their ROAD responses respectively, and completes its own ROAD response by adding information regarding its local data processing. In order to illustrate the overall process created by ROAR and ROAD, Figure 1 shows four different examples of ROAR and ROAD requests and responses. The first case shows the data subject querying its directly adjacent company C2 (for instance, this may be an online shop where the data subject once bought something). The user sends a ROAR request to C2's ROAR Web Service endpoint. Company C2 determines that one of the personal data items (e.g. data subject's email address) it holds was passed to the business partner C4 (e.g. a supplier). Hence, C2 creates a ROAD request querying C4 directly for information regarding that particular interaction only. In the same way as C2, C4 determines that it passed the data item to companies C5, C6, C7 (e.g. billing provider, delivery, and marketing service). C4 queries each of them using an appropriate ROAD request. C5, C6, and C7 determine that they had processed the data item themselves, but did not pass it to any other business partner. All of them respond to their particular ROAD request, providing information as to what they did with the personal data item in question, and for what purpose. Collecting these replies, C4 creates its own ROAD reply, which includes the ROAD responses of C5, C6, C7, and sends it to C2. Finally, C2 answers the initial ROAR request by the data subject with a full report on what happened to the personal data.

It is important to note that C4 did not ask C1 for input, even though it is possible that personal data may have been received via C1. That data is not part of the local view of the initially queried C2. Hence, it is not to be included in C2's ROAR response. However, C4 is able to determine which data of the data subject it received via C1 and which arrived via C2 facilitating the correlation sets (cf. Section II-C).

The second case of Figure 1 shows the process when the data subject directly queries C6. Again, C6 determines that it has not forwarded personal data, but that it has received personal data from C4. In order to give an enhanced answer to the posed ROAR request, C6 will trigger a ROAD request towards C4, asking for additional information regarding the data sources. Then, C4 determines that it has received personal data via C1 and C2 and propagates the ROAD request to both of them. Note that neither C5 nor C7 are queried by C4, as they do not belong to the process that lead to the source of the personal data ending up at C6. In the same way as before, the ROAD answers of C1 and C2 are aggregated in C4's ROAD response, which is incorporated in C6's ROAR response to the data subject.

Straightforward, case 3 of Figure 1 illustrates a ROAR request on C4. This implies ROAD requests being propagated in both directions: towards predecessors and successors of C4. Again, there are no ROAD requests sent to C3, as it is not part of the business processes involving C4. As before, the responses are collected and put in the ROAR response of C4.

For comparison, Figure 1 case 4 shows the result of case 3 if no ROAD requests are sent. Here, the data subject will not gain any information regarding companies beyond the local information of C4. It would be required to ask each of C4's adjacent companies directly (using ROAR) in order to gain the complete view on the personal data's trail.

B. Message Structure

On the technical side, the ROAR and ROAD services are realized using Web Services specifications. Consequently, putting message data to the wire is straightforward use of WSDL and SOAP standards. In the following, we will give a description on the message structure of both ROAR and ROAD requests and responses. Cryptographic means applied to these are introduced in the next section.

The request messages of ROAR and ROAD services are rather simple. For ROAR, the request message contains a data subject identifier along with an authentication token and a request ID. Once a ROAR request is placed at a company's ROAR service, that company may decide to request other companies of a certain business process. These ROAD requests are very similar to the ROAR requests, but additionally contain an identifier of the ROAD requester. That identifier is a correlation set instance (cf. Section II-C) identifying the exact service interaction. Besides that, a ROAD request also contains an "anonymity flag". This flag signals that data sources and recipients provided in the ROAD answers have to be categorized (instead giving of the companies name and its representatives). Thus, a company in the workflow can signal to preserve its trading secret. Once the anonymity flag is set,

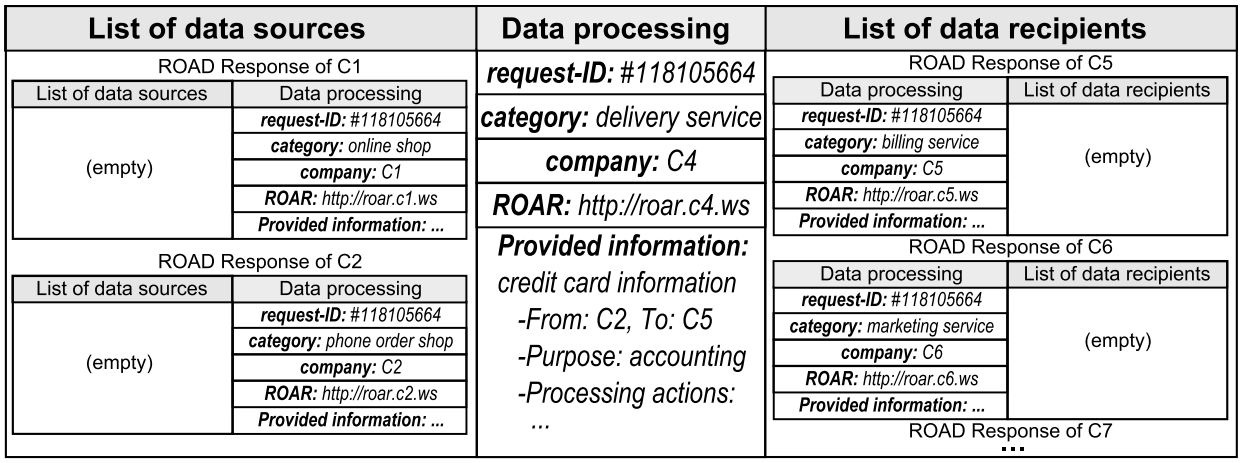

Fig. 2. ROAR response message for a ROAR request to C4 of Figure 1 case 3)

it remains set for every subsequent ROAD request. A more in-depth discussion on the issues of ROAR and ROAD requests—beyond their technical realization—is given in Section IV-B.

The structure of the ROAR and ROAD response messages is more complex (see Figure 2). As can be seen, a ROAR response message (here an example for a ROAR request towards `C4` in the case 3 of Figure 1) contains three distinctive blocks: data sources, data processing information, and data recipients. For each ROAR response, a set of ROAD source blocks is given (disclosing the personal data's origins, here `C1` and `C2`), as well as a set of ROAD recipient blocks of companies that directly received personal data from `C4` (here `C5`, `C6`, `C7`). The third block contains the unique request identifier of the overall ROAR request as well as all information regarding `C4`'s operations on the personal data. It lists the company's business category (e.g. delivery), full name and a legal representatives' identity (e.g. Federal Express Corporation Inc. and Frederick W. Smith), ROAR service endpoint (in case the data subject wants to place another ROAR request), and a description of all operations done to the personal data of the data subject within the company. Within the last field, every processing step of personal data within a company is listed, denoting that data's origin, processing steps, purposes of processing, and recipients (cf. Section II-A).

The ROAD blocks contained within the ROAR response illustrate the result of the subsequent ROAD requests triggered by the ROAR request (here to `C1`,`C2`,`C5`,`C6`,`C7`). They contain the same kind of information as for the ROAR response, but list either additional sources (for data origin ROADs) or additional recipients (for data forwarding ROADs). Each such ROAD block also contains the same set of data processing information as the ROAR response, i.e. category, company, and provided information. For the special case of the data source being the data subject itself (e.g. for `C1` and `C2`), the source list is left empty, and the data subject is denoted directly within the provided information field. The same applies for personal data being forwarded to the data subject, if ever.

Being a recursive message format, it is possible to illustrate the full spanning tree of data processing instances for the personal data. It discloses all companies involved in processing such data, along with their correlation. Hence, it gives the data subject a very broad view on the data processing network as a whole. Note that though it is possible for the chain of data processing instances to contain loops (i.e. a data item is sent back to a previous processing instance), this does not affect the proposed architecture, since the business process endpoint—and hence the processing logic—is different from the processing logic in the first pass. Hence, the ROAD response might contain the very same company more than once, but there is no threat of "infinite recursion".

C. Security: Assumptions, Design, and Cryptographic Means

The ROAR/ROAD services are designed to achieve the following security goals: Request authorization, confidential providing of information, and trade secret protection. We assume that a ROAR/ROAD message between two Web Services additionally fulfills the following requirements:

- a message's integrity is protected,
- its origin can be authenticated,
- it is not possible to repudiate transmission or receipt of the message, and
- a message's confidentiality is protected appropriately (partially or completely).

How trust in a certain company or human user is actually established goes beyond the scope of this paper. We use a sanitizable digital signature scheme (cf. [7]) for request authentication. In a nutshell, to validate the signature in a sanitizable signature scheme either the original data or a blinded version must be present. For ease of use we use a sanitization scheme which allows everyone to blind data items, so without exercising disclosure control [8]. Further, an asymmetric public key encryption scheme is used for confidentiality protection. We allow users to have more than one identity [9] by using a different key-pair for each identity. This, and the rational behind other design decisions is discussed in detail in Section IV.

1) Request Authorization: Each request message, by assumption, is from an authentic source, integrity protected and cannot be repudiated. Additionally, each ROAR or ROAD request carries a credential called *authenticator*. This message's authenticator is used to ensure that only the data subject or someone acting on his behalf shall be able to successfully place requests. We differentiate four types of authenticators:

- **data-subject-authenticator**: An unforgeable token that allows controllers to associate the data affected by the requested action with a data subject. As a prerequisite, this requires a slight change to the implementation of the underlying business processes. More precisely, the data subject is required to digitally sign the personal data once before it gives it to any company as part of a business process execution (cf. [10]). That signature is the data-subject-authenticator. It must be retained throughout all business processes along with the personal data. However, as personal data items may be split during a business process execution, this approach requires the use of sanitizable signature schemes in order to remove data fragments while keeping the signature's validity. In order to authenticate a ROAR request the data subject proofs fresh knowledge of the private key by signing the ROAR request using the same private key that signed the personal data that was used in the initial business process executions. Hence, the controller can verify that the private key used for signing the ROAR request matches the private key used for signing the personal data that the controller has on record. This way, a data subject can authenticate ROAR requests to any Web Service in the composition. Additionally, every controller can verify the request from a data subject without the need of a direct relationship to be established a priori, and without the need for a public key infrastructure.
- **controller-authenticator**: An unforgeable token identifying a controller as a participant of a certain service composition.
- **context-authenticator**: A token containing elements from a correlation set (cf. Section II-C). Thus, controllers are able to associate the affected data with all Web Service interactions that involved that particular requester. This authenticator works without any additional changes to the underlying business process implementations.
- **data-knowledge-authenticator**: This authenticator is not a single token. The controller demands the requester for values of certain data items from the personal data in question (i.e. name, postal address, and age). The controller compares if the requester's values match the ones she stored locally. If both match, the request is authenticated by proving data knowledge. This is a fallback, mimicking the existing process for postal requests.

Depending on the actual business process scenario, these authorization methods can also be used in combination.

2) Confidential Providing of Information: Each controller must make sure that only the valid data subject is provided with the information gathered by ROAR requests. We achieve this by encrypting the relevant parts with a public key that belongs to a private key that is known only to the data subject and was associated with him during the authentication (as explained above). In all ROAR and ROAD replies, the following parts are encrypted: `Provided information`, `List of data sources`, and `List of data recipients`.

3) Confidentiality of Trade Secrets: A company might not want to reveal the identities of its succeeding business partners, as that knowledge may be a trade secret (e.g. for resellers in a supply chain). Hence, when receiving a ROAD request from its predecessor (or a ROAR request from the data subject himself), that company may decide to keep its successors confidential even in the ROAD/ROAR responses. However, as this approach conflicts with the data subject's legitimate right of access, a company may decide to reveal the generalized category of its successors (e.g. "a marketing service"), but blind out the actual name and contact information (i.e. the `company` and `ROAR` fields of the particular ROAD response message). The same holds true for upwards directed (querying predecessors) requests.

In order to enable the data subject to verify that ROAD response's integrity, we use a sanitizable signature scheme for signing the ROAD responses so that it becomes possible to blind out those two data fields without breaking the message's signature. For the example given in Figure 2, company `C4` may decide whether it wants to reveal `C5`'s identity to the data subject or not. Nevertheless, the data subject can still verify the signature of `C5`'s ROAD block.

However, `C5` might have additional successors whose ROAD responses are contained within `C5`'s ROAD response. As these might name `C5` explicitly, we need the `anonymity flag` introduced in Section III-A in order to tell `C5` (and all of its successors) already within the ROAD request that they must not include that data in their ROAD responses.

In such a scenario, the data subject will get a ROAR response that lists all organizations that processed its personal data. If an organization's identity belongs to a trade secret, the overall business process part below that organization is reduced to information anonymized by category. Note that this nevertheless reveals far more useful information than the (more simple) approach of not propagating ROAD requests to "secret" business partners at all.

IV. Discussion

In this Section the merits and flaws of the proposed approach will be discussed. It is demonstrated that the approach not only suits well for the requirements stated in Section II but also improves requests to multiple controllers of a business process. In particular, it shows how the approach can help solving the conflict between the right of access and the trade secret, and therefore is creating added value for the data subject and the controllers.

The Directive 95/46/EC [1] is implemented by national law of each EU Member State. In the following, we will discuss the basis of the German implementation of data

protection law (BDSG [11]). This will help to give a more detailed understanding on how legal obligation can be fulfilled. However, the argumentation is valid for any EU Member State, since the law is unified by the Directive [1].

A. Merits and Flaws of using Web Services

On the merits side, our approach is beneficial as the service is available for everyone using the Web Service architecture. For the data subject, this is a known access point and the request is automatically in the right place. In comparison to non-technical requests, there is no need for internal forwarding to the official in charge. In the standard case, the approach requires no manual processing, since the response can be created automatically from the processed data and the existing data about the processing within the service architecture. Using the same architecture for providing information as the underlying business ensures scalability. Thus, saving human resources at the controller's side results in a reduction of the response time from weeks to minutes, following the laws intention of a prompt response ([12], BDSG §34, recital 16). Only special cases have to be processed manually.However, even for manual cases, the response can be given as a Web Services again, if the official in charge creates the ROAR or ROAD response manually (or technically supported).

On the flaws side, providing information on data processing beyond Web Services requires an interface and must be provided manually. This is no real flaw, because without Web Services it would have to be processed manually, too. Even then, manually processing ROAR and ROAD requests and responses could be possible, since Web Services use SOAP messages that can easily be converted to human-readable form (cf. [13]). For the same reason, you can participate even with manual processing in the otherwise fully automated ROAR/ROAD architecture. This provides resilience, if the automatic information fails. Adding the ROAD service suggests that a complete view on a business processes is provided to the data subject. This is not necessarily true, because responding on a ROAD request is technically and legally optional. However, a controller cannot remove herself from the response (but anonymize). The data subject is informed about the full list of recipients and sources and can at least assume further processing. In the unanonymized case, the data subject can direct a ROAR request to such controllers who are then legally obliged to answer.

To conclude, if ROAR and ROAD services are used, then there is no disadvantage in comparison with manually providing information.

B. Fulfilling Legal Obligations

For fulfilling legal obligations it is not only necessary to provide the required information, but also to check the legitimated interests and to ensure the confidentiality of the provided information. As a rule, the information has to be provided in writing in a human readable and understandable form. When ever appropriate, the information can be provided in an other form ([12], BDSG §34, recital 14). In addition, the controller has the due diligence to check the requester's identity and legitimate interest ([12], BDSG §34, recital 6). To a greater extent, providing the wrong person with the information is a *illegitimate service* (in terms of law) and can be punished as an *administrative offense* (ibid.). Thus, it is in the best interest of controller and data subject to ensure a reliable authentication. Since method and procedure of the authentication is chosen by the controller ([12], BDSG §34, recital 7), the controller is responsible for the effectiveness of the authentication.

1) Checking the Legitimate Interest: In the standard case, the controller can assume a legitimate interest, if the requester references a previous communication or proves knowledge on the stored postal address ([12], BDSG §34, recital 7). As an exception from the standard case, providing information via ROAR/ROAD architecture requires additional actions on authentication. Essential for checking the legitimate interest is that the requester is the same person that has given the processed data. If the underlying data processing uses a data-subject-authenticator (cf. III-C1), then it is already linked to the context. Using this authenticator for the request for information proves that it is the same identity, since we assumed that the authenticator is unforgeable and under the data subject's control. If there is no such authenticator, then the data subject has to use the data-knowledge-authenticator (cf. III-C1) and identify himself, e.g. using a Class 3 certificate for signing this information. Thus, in any case, the legitimate interest of the requester can be determined.

2) Confidential Providing of Information: The controller has to ensure the exclusive delivery to the legitimated data subject. If the answer is given by letter, then this is ensured by using the correct postal address. For our architecture however, we have to bind the ability to read the answer to the identity of the data subject. This is done by encrypting the response under the data subject's public key that was provided with the request and authenticated as the data subject's during the legitimate interest check. This might be important, if the controller has to prove a delivery to the correct subject or, in case of an illegitimate service, has to name the wrongly informed subject. Thus, the law's intention to provide all information to the correct data subject is fulfilled. To conclude, the information is provided securely, as shown. Furthermore, it forms the basis for a human readable and understandable visualization (e.g. as proposed by the PRIME Project [14]). Hence, our ROAR/ROAD architecture is suitable.

C. Right of Access and Protected Trade Secrets

Currently, the law offers only one possibility to protect data sources and data recipients as a trade secret: To categorize them in the answer. As a result, the data subject cannot gain knowledge on data processing beyond the point of categorization. As categorization minimizes his right of access it goes against the laws intention (cf. II-A). Thus, trade secret and right of access end up conflicting each other.

To resolve this conflict, we propose to delegate the request throughout the whole business process to collect information.

Since every controller knows its predecessor and successor, the request can also be continued if sources and recipients have to be categorized in the answer. It is upon the controller (or upon her predecessors or successors), if she categorizes her sources or recipients, if they are her trade secret. Thus, the controller must be able to sanitize signed answers before including them in the reply message. We achieve this by using a sanitizable signature scheme. Hence, we allow categorization to enable trade secret protection while providing a representation of the personal data's flow for the whole business process.

Note, categorization is only effective, if all descendants or ancestors of an entity also categorize their answers. We achieve this by setting the `anonymity flag` in the request (cf. III-B) to indicate that all further triggered ROAD requests are to use categorization too. If a controller ignores this flag, then she violates the business contract with his predecessors or successors. Forensic investigation would reveal the violation, since the answer is signed by the controller, and allows imposition of liquidated damages.

We achieve an efficient lookup as we restrict the delegated request to a specific context via correlation sets and allow only upwards (sources) or downwards (recipients) delegation.

Restricting the context allows delegation, but results in reduced provided information. However, providing full information is not always helpful. The data subject would like to assess the impact of processing his personal data in a specific business process. Establishing this from dedicated ROAR requests for every participant is hard or even not possible, since the provided information can originate from several, but not distinguishable, business processes. Our solution of a ROAD request restricts the context, but allows business-process-traversal. Thus we create an added value, as it reveals the personal data's flow regarding to an actual business process.

For legal compliance, the data subject has to authorize the controller to send ROAD requests. In this case, the validity of the mandate has to be verified by the receiver ([15], BDSG §33 recital 20). In our approach, the validity check is implemented by providing the data-subject-authenticator, the controller-authenticator, and the context-authenticator related to the correlation set. The data-subject-authenticator justifies the mandate, the controller-authenticator ensures correspondence with the correct mandate holder, and the context-authenticator addresses the concerned business process.

Overall, our architecture offers a trade-off between preserving trade secrets and providing information to the data subject. This is done in accordance with the law and also technically improves the process and enhances the provided information.

V. Conclusion

In this paper, we present a solution to automate the process of providing information according to the right of access for service-based business processes. We propose to add a Web Service for providing information to the data subject. To our consideration, the approach is sound and secure. We address the problem of scalability by using Web Services and provide solutions for the authentication and confidentiality. Additionally, we support keeping the trade secrets of the involved companies protectable as a great step forward in solving the general conflict between the economic goals of secrecy and informational self-determination.

Additionally, our solution provides several advantages compared to the manual process:

- reduction of overall communication traffic (when using the ROAD approach),
- lower costs for an automated solution (compared to customer support center for manual processing), and
- shortened response time for a request.

This is a first approach towards the target of full technical implementation of the data subject's indispensable rights of informational self-determination. Our solution builds upon existing technical architectures (Web Services), is motivated by business obligations, and implements one of these rights (right of access) according to the law. Future work is to analyze and implement the other rights.

VI. Acknowledgments

We thank Focke Höhne for discussions on legal aspects.

References

[1] EU, "Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data, Official Journal of 23 Nov. 1995, L 281, page 31 - 50," Nov. 1995.

[2] F. Kerschbaum and P. Robinson, "Security architecture for virtual organizations of business web services," *Journal of Systems Architecture, Special Issue on Secure SOA*, vol. 55;4, pp. 224–232, 2009.

[3] C. M. MacKenzie and et al., "Reference model for service oriented architecture 1.0," *OASIS Standard*, 2006.

[4] S. Weerawarana and et al., *Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More*. Prentice Hall PTR, 2005.

[5] D. Jordan and et al., "Web Services Business Process Execution Language Version 2.0 (WS-BPEL 2.0)," *OASIS Standard*, 2007.

[6] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web Services Description Language (WSDL)," *W3C Note*, 2001.

[7] G. Ateniese, D. H. Chou, B. D. Medeiros, and G. Tsudik, "Sanitizable signatures," in *ESORICS: Proceedings of the 10th European Symposium on Research in Computer Security*. Springer-Verlag, 2005, pp. 159–177.

[8] K. Miyazaki and et al., "Digitally signed document sanitizing scheme with disclosure condition control," *IEICE Transactions*, 2005.

[9] Z. Chen, "A scenario for identity management in daidalos," in *CNSR '07: Proceedings of the Fifth Annual Conference on Communication Networks and Services Research*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 176–183.

[10] H. C. Pöhls, "Verifiable and revocable expression of consent to processing of aggregated personal data," in *ICICS 2008*, ser. LNCS 5308, M. R. L. Chen and G. Wang, Eds. Springer, 2008, pp. 279–293.

[11] Federal Republic of Germany, "Bundesdatenschutzgesetz - BDSG (in German)," 2009.

[12] P. Gola and R. Schomerus, *Bundesdatenschutzgesetz (BDSG), Kommentar (in German)*. Beck Juristischer Verlag, 2007.

[13] S. Heinzl and et al., "The web service browser: Automatic client generation and efficient data transfer for web services," in *ICWS*, 2009, pp. 743–750.

[14] M. Bergmann, M. Rost, and J. Pettersson, "Exploring the feasibility of a spatial user interface paradigm for privacy-enhancing technology," *Advances in Information Systems Development: Bridging the Gap Between Academia and Industry*, p. 437, 2006.

[15] H.-J. Schaffland and N. Wiltfang, *Bundesdatenschutzgesetz (BDSG), Ergaenzbarer Kommentar nebst einschlaegigen Rechtsvorschriften (in German)*. Erich Schmidt Verlag, 2009.