

Supporting Evidence-Based Compliance Evaluation for Partial Business Process Outsourcing Scenarios

Philip L. Miseldine, Ulrich Flegel, Andreas Schaad
SAP Research, Vincenz-Prießnitz-Straße 1, 76131 Karlsruhe, Germany

Abstract

We present the challenges facing businesses wishing to outsource processes to service providers who must maintain regulatory compliance via data access control procedures. We argue that it is not currently possible to capture the necessary agreements, and supporting evidence, pertaining to the usage of data a client may send to a service provider. As a result, the richness of evidence and controls a client has available to it reduces when they choose to use an outsourcer, therefore lessening the business value of considering service outsourcing. The paper introduces a model to clarify these issues, which is implemented against a health-care scenario, to show how data usage in an outsourcing scenario can be better captured and controlled.

1. Introduction

Outsourcing complex and costly IT processes to 3rd party software suppliers has become an increasingly popular choice for enterprises looking to save on IT costs. Attracted by the potential savings in reduced infrastructure maintenance and development, outsourcing has become a principal driver of service oriented architecture adoption by business. By utilising services, suppliers can expose functionality that can be remotely invoked by a client on-demand, and thus be easily integrated into the existing processes of the client.

Services however, require client data to be sent to them for input. This can include personal data, or any company sensitive data. Consequently, outsourcing raises many issues with regard to privacy and data protection, as critical business data needs to be sent and processed externally. Such issues matter to an enterprise, as they must ensure their IT systems comply with industry, statutory, and established practices for risk management, and stakeholder protection. Failure to maintain compliance can have severe legal consequences that can lead to actual imprisonment [12]. Compliance management is the practice of providing evidence supporting existing compliance suitable for audit, and the

definition of measures ensuring continued compliance over enterprise-wide processes. Accordingly, as the enterprise seeks to outsource business processes, so the task of compliance management becomes more difficult to achieve, as the compliance procedures of the outsourcer, and associated evidence must be ascertained. Coupled with this, if the outsourcer operates in a different country, the type of regulations it must comply with may well differ from the client it is supplying. Thus, there is an urgent need for compliance management to manage outsourced IT processes in as detailed, and automated fashion as possible.

1.1 Motivation

Similar to compliance, Quality of Service (QoS) issues require explicit agreements to be reached between a client and supplier. In the realm of QoS, these agreements are formalised using Service Level Agreements (SLAs), that form a contractual obligation between the client and supplier of a service, using languages such as WSLA [8]. These ensure that the requirements of the partnership are clearly defined and can be assessed. The implication for a SLA definition is that for it to be successful, concrete identifiers must be chosen that can be used to assess whether the contract was successfully upheld. For this reason SLAs traditionally focus on a technical level, as measurable indicators can be easily sought. Common examples include guarantees on protecting, and assuring the performance of services, such as network packet loss [9].

To be useful, all compliance frameworks must allow evaluation of the controls they either explicitly define or that have been implemented as part of a compliance refinement process. An explicit compliance control could be one that demands that some provable system features be in place. Accordingly, the evidence attesting to this control needs no real assessment; either the feature is in place, or it is not. An example of such a framework is HIPAA [11], where the privacy of patient data is protected by tangible controls, such as ensuring access to patient data via security keys [7]. As such, traditional SLA definitions can be applied that demand the presence of these features in the same way as the

low level Quality of Service issues. Frameworks such as Sarbanes-Oxley (SOX) however, define more conceptual, implicit controls that must be enforced and assessed with a posteriori evidence. There are also situations where evidence can only be defined and generated at a specific point or time-span, such as in data access control. In this regard, it is imperative to support the generation and retrieval of evidence, such that it can be later assessed. Further, SOX requires identity and access management that can both control and validate user-machine interactions. To support this on an implementation level, this implies that evidence is obtained about exactly *who* accesses *what* financial information, *how*, and *when* [6]. Thus, supporting evidence of the actual usage of data is required that can later be assessed to demonstrate compliance.

In an outsourcing scenario, this evidence must relate directly to the data being processed so that all uses of the data can be captured as evidence for the client. However, the evidence supporting the assertions and agreements the supplier is contracted to supply is commonly not provided to the client at the granularity of individual service calls. Similarly, contracts of usage for the data in a service call typically do not allow references to data usage constraints. Rather, an aggregated overview of the control procedures of the outsourcer is provided so that the client can trust the compliance of the outsourcer to regulatory standards. For the client, this implies that they cannot get concrete evidence of how data sent via a particular service call was used, nor specify concretely how actual data in the service may be used. Thus, the client of an outsourcer will lack the detailed supporting evidence they require to demonstrate compliance to implicit controls. This also places the burden of maintaining evidence with the outsourcer, who must then take steps such as anonymisation to protect the data for later analysis. Similarly, as compliance agreements must be reached on a broader level, determining how the ramifications of these agreements relate down to providing sufficient controls needed at the service level introduces complexity in reaching a settled agreement between partners. In contrast, processes internal to the client can be monitored on an individual basis to provide supporting evidence, e.g. capturing evidence of which user executed a particular transaction. As a result, this produces a mismatch in richness of the evidence and control a client can have when they choose to use an outsourcer, lessening its business value.

As its major contribution, this paper seeks to remove this mismatch. We outline a model that encapsulates and persists audit data and contractual obligations within an object sent via service invocation. The object can then be returned to the client enriched with evidence that can show if the obligations were upheld. Requirements for common interfaces to facilitate the definition of control, and evidence pertinent to, compliance measurements are defined, allowing

contractual concepts similar to SLAs to be applied directly to usage of individual service call data.

2 Model Definition

To provide evidence of the usage of data sent by a service, a client needs to be able to specify what evidence they require from the supplier, and the supplier needs to be able to convey this evidence in a manner that is readable by the client. These definitions must be at the granularity of the data access, so that the supporting evidence pertaining to the usage of data sent in a service call can be collected. Data sent to a service is typically encapsulated within properties within an object. As the contract, and evidence of this data usage is wholly related to this object, it follows that they should be encapsulated within the same object as well. Thus, an object sent to a service can be enriched to describe what the service supplier should provide as evidence and controls needed in the form of a contract, and the supplier returns the object enriched with this evidence in the form of an audit log. The benefits of this encapsulation are numerous. Firstly, the supplier need not provide their own infrastructure to store evidence, as this data is contained within the object itself. Second, the object can reference previous evidence regarding its usage. This is important when considering contracts that include temporal aspects, such as separation of duty constraints. In these scenarios, where the same user cannot access two subsets of data, it must be ascertained what else a user has accessed previously. With the evidence encapsulated, the object has available to it the necessary information to determine whether such a constraint could be violated. In section 5, we acknowledge the fact that for this to be achieved in a secure way, appropriate safeguards must be in place.

There will be data that needs to be provided as evidence that is not encapsulated within the object being accessed. An identifier of the user who is accessing the object will not be a property of the object, but rather form part of the context of this request. The contract of usage therefore states firstly what accesses should be captured, and constraints on what the context provided for the request should contain. As an illustration, let $O(a, b, c, e)$ represent an object O with the properties a , and b , and a contract c , and evidence e . It is required that each read access of a be logged along with the user making the request. Accordingly, a contract would be defined that states that $c = r(a) \mapsto u$, where u represents a user ID, and $r(a)$ represents a read access of a . The mapping forms the contract, relating this access definition to a context, such that it can be implied that when a read access to a is made, u must be provided as context. e records all references made in c , giving $r(a)$. For a separation of duty constraint, where a , and b cannot be accessed by the same user, a contract can be defined such

that $c = r(a) \wedge r(b) \mapsto (u_1 \wedge u_2) \wedge (u_1 \neq u_2)$. Here, to read both a and b , two users are required as evidence (u_1 and u_2), however they cannot be equal. Here, e is $r(a), r(b), u_1$, and u_2 . In summary, an object encapsulating the data sent to a service will have associated with it two classes: one representing the contract of usage, and one the evidence.

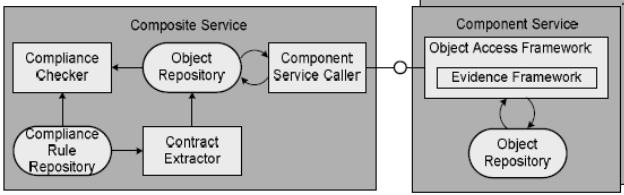


Figure 1. Architectural Support

This model presumes that the implementation of the object on the supplier side does indeed correctly implement the notion of logging accesses to data properties, and thus fulfills the contract provided. It also presumes that the supplier will keep the data within this object, and not copy it to another object, where capturing data accesses can be lost. A reference architecture is therefore needed, that can provide an implementation that can be trusted, based on certification according to an appropriate Common Criteria profile. The presence of this architecture, and its correct implementation, can then act as evidence of controls being in place to protect data access issues. The integrity of the implementation could be attested by leveraging TPM hardware [4].

A candidate architecture is presented in Figure 1. For the client, a composite service is defined, that receives an object from a repository and embeds a contract of usage, produced from a *Compliance Rule Repository*, and implied evidence needed. The *Contract Extractor* extracts from the wider compliance rules, those rules applicable to the data object being sent, allowing the access control constraints to be defined in the contract. This is then sent to the supplier, where the *Evidence Framework* executes the contract to provide the evidence, and required control. On return, the *Composite Service Caller* updates the *Object Repository* with the new object state, and assesses the object via a *Compliance Checker* to ensure contextual obligations were upheld.

For the evidence to be trusted to be complete, all possible usages of the data sent to the service must be logged in accordance to the contract. In an object implementation, the logging of accesses to a data property can be controlled using accessors. Accessors are a construct in most object-oriented languages that allow logic to be executed when a read or write access of a property is performed. Therefore, accesses to properties can be logged via accessor logic to provide the requisite evidence of usage. It is here too that analysis of the evidence can take place, and constraints on usage be determined that require temporal, historical data.

Indeed, the accessor can make access request evaluations, such as returning a null value to the request for a denial based on analysis of the calling context, and existing evidence. As these behaviours are all encapsulated within the object, any access to the data properties of the object can be guaranteed to be logged.

3 Application Scenario

As an illustrative example as well as initial evidence on the validity of our approach we consider the collaborative process of clinical trials in an overall pharmaceutical development process. Parties involved in the development process include researchers, clinical scientists, government and regulatory bodies, investigators, sponsors, and possibly more. To communicate effectively together, services are crucial. During the trial, expertise from statistician services may be called upon for data evaluation, as well as allowing cost reductions for the required processing infrastructure. This setup is thus characterised by autonomous participants; a high degree of data collection and processing activities; possible outsourcing at several stages of the trial; strict operational compliance demands as well as regulations such as HIPAA [11].

Only partial information of the patient related to the statistical analysis should be available to the statistical service. More precisely, it needs read access to laboratory results; read access to a patient's history as well as personal data such as residence and income group for a correlation with other statistics; but only write access to statistical analysis results. Using the definitions given in Section 2, let the set P represent all properties of an object O representing a trial record, that are related to identifiable personal data, and the set L representing laboratory results, H the set relating to patient history, and S the statistical results. w is defined to be a write access, and $role$ defines a role. The following contract can thus be defined:

$$\begin{aligned}
 c = r(P) &\mapsto u \wedge role = \text{"statistician"} \\
 r(L) \wedge r(H) &\mapsto u \wedge role = \text{"statistician"} \\
 w(S) &\mapsto u \wedge role = \text{"statistician"}
 \end{aligned}$$

In our approach this would imply that the trial record will evaluate property accesses through the accessor construct, as enforced by the Object Access Framework, and grant access to the required read operations and data fields in accordance with the specified policies. e here is $role$, and u , for each r or w request, which are themselves included. This will be generated by the Evidence Framework. Our approach also supports context-dependent decisions on how a trial record will react to access requests. For example, a statistician may have a need to read prior statistical analysis results, but only if later messages to the object regard-

ing a write to the actual analysis results originate from another principal, i.e. the Object Access Framework supports dynamic separation of duty policies based on the evidence collected within the object. After analysis the statistical service will return the trial record objects containing the collected evidence, which the Component Service Caller will update in the Object Repository. At regular intervals the Compliance Checker will verify compliance with the policy in the Compliance Rules Repository. Note that these rules may also refer to objects processed by different or by several component services, thereby capturing compliance issues that could not be detected locally by any component service Object Access Framework instance alone.

4 Related Work

EPAL is a language [10] that allows the specification of fine-grained enterprise privacy policies. It concentrates on core privacy authorisation while abstracting from all deployment details such as data model or user-authentication. Methods for arriving at the definition of these rules are available [3]. An EPAL policy allows *purposes* of data to be defined, *actions* that model how the data should be used, and *conditions* that specify contextual constraints on the actions. These elements are then used to formulate authorisation rules that allow or deny actions to categories of data. EPAL is a candidate for representing the contracts as defined in this model, however is geared specifically for evaluation, rather than specifying requirements for generating evidence. The notion of *sticky policies* is promoted by Bandhakavi et. al [2], where a policy sticks to the data, travels with it, and can be used to decide how the data can be used. This is similar to the notion of embedding the contract of usage within an object sent to a service, however again, does not specifically focus on the evidence accrual.

Protecting data via access is also a core concept of Hippocratic Databases [1]. In these systems, a context is provided and evaluated when an access is requested. An evidence store is then captured with these potential accesses. This work is a motivating example of a similar solution to data protection, however in this paper we provide similar support at the service level, rather than requiring the database itself to be queried.

5 Conclusions and Future Work

This work represents an early stage analysis on the issues of defining and implementing contracts for data usage in outsourcing scenarios, with support for the necessary evidence generation. A model is proposed that was applied to a medical healthcare scenario. Many outstanding issues remain. Defining how the components in the architecture would precisely produce the behaviours required is

an outstanding issue. For instance, how the Contract Extractor would relate high-level compliance rules down to the level of a service, is a challenge that requires further investigation. Securing the objects used between the client and the service supplier has not been considered, and thus the ability for 'man-in-the-middle' attacks to augment the evidence or contracts contained within the object cannot be discounted. Work conducted in [5] describes scenarios and issues involved in promoting the secure transportation of data in outsourcing scenarios, and thus would help further elaborate the model defined here. Other open issues include the performance overhead for a supplier of having to maintain the object across its operations, as including contractual evidence will certainly increase its structural complexity, and memory footprint.

References

- [1] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Hippocratic databases. In *VLDB*, pages 143–154. M. Kaufmann, 2002.
- [2] S. Bandhakavi, C. C. Zhang, and M. Winslett. Super-sticky and declassifiable release policies for flexible information dissemination control. In *WPES '06*, pages 51–58, New York, NY, USA, 2006. ACM.
- [3] T. Breaux and A. Anton. Analyzing regulatory rules for privacy and security requirements. *IEEE Transactions on Software Engineering*, 34(1):5–20, Jan.-Feb. 2008.
- [4] A. Hohl, L. Lowis, and A. Zugenmaier. Look who's talking - authenticating service access points. In *SPC*, volume 3450 of *LNCIS*, pages 151–162, 2005.
- [5] Y. Karabulut, F. Kerschbaum, F. Massacci, P. Robinson, and A. Yautsiukhin. Security and trust in it business outsourcing: a manifesto. *ENTCS*, 179:47–58, 2007.
- [6] N. N. Kulkarni, K. M. S. Kumar, and D. S. Padmanabhuni. Reckoning legislative compliances with service oriented architecture a proposed approach. In *SCC '05*, pages 16–23, Washington, DC, USA, 2005. IEEE.
- [7] P. E. Pancoast, T. B. Patrick, and J. A. Mitchell. Physician PDA use and the HIPAA Privacy Rule. *J Am Med Inform Assoc*, 10(6):611–612, 2003.
- [8] SESSION: Service and Network Upgrades. Defining and monitoring service-level agreements for dynamic e-business. In *LISA '02*, pages 189–204, Berkeley, CA, USA, 2002. USENIX Association.
- [9] J. Sommers, P. Barford, N. Duffield, and A. Ron. A framework for multi-objective SLA compliance monitoring. *IN-FOCOM 2007. IEEE*, pages 2446–2450, May 2007.
- [10] W. H. Stuffelbeam, A. Antón, Q. He, and N. Jain. Specifying privacy policies with p3p and epal: lessons learned. In *WPES*, page 35. ACM, 2004.
- [11] United States DoH. *Medical Privacy - Standards to Protect the Privacy of Personal Health Information*.
- [12] US. Code Collection. *Destruction, alteration, or falsification of records in Federal investigations and bankruptcy*, chapter 18.1.17.1519. US Government, 2002.