A Toolkit for Automating Compliance in Cloud Computing Services

Nick Papanikolaou*, Siani Pearson*, Marco Casassa Mont*, and Ryan Ko**

*Cloud and Security Lab, HP Labs, Bristol, UK

**Cloud and Security Lab, HP Labs, Singapore

Abstract: We present an integrated approach for automating service providers' compliance with data protection laws and regulations, business and technical requirements in cloud computing. The techniques we propose in particular include: natural-language analysis (of legislative and regulatory texts, and corporate security rulebooks) and extraction of enforceable rules, use of sticky policies, automated policy enforcement and active monitoring of data, particularly in cloud environments. We current work on developing a software tool for semantic annotation and natural-language processing of cloud terms of service and other related policy texts. We describe our implementations of two parts of the proposed toolkit, namely the semantic annotation editor and the EnCoRe policy enforcement framework. We also identify opportunities for future software development in the area of cloud computing compliance.

Keywords: cloud computing, compliance, accountability, natural language processing, policy enforcement.

This paper is an expanded and revised version of a paper entitled *Automating Compliance in Cloud Computing Services* presented at CloudSecGov 2012, Porto, Portugal, 18-20th April 2012.

1. Introduction

This paper presents tools and techniques for automating compliance with law, regulations, and other requirements, particularly in the context of cloud computing. The most widely used definition of cloud computing is by NIST (Mell and Grance, 2011):

"Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models."

What makes compliance difficult for providers of cloud computing services (referred to heretofore as *cloud service providers*) is the sheer number and complexity of laws and regulations that need to be understood and enforced in their systems. Cloud service providers tend to host their customers' data and the computing infrastructure they use in several, disparate data centres, which are physically located in several different jurisdictions. If a customer's data is stored in a data centre located in Germany, for

example, it will be subject to German data protection law, which is much more restrictive than data protection law in many other countries. In addition to national laws and regulations, there are international agreements and treaties regarding the transfer of data between different jurisdictions (aka. **trans border data flows**), and the US-EU Safe Harbor agreements are a well-known example. Cloud service providers are expected to take all the relevant rules into account and take appropriate measures.

The way a cloud service provider handles its customers' data is usually specified in a written contract or agreement which comprises the ToS (Terms of Service) and SLA (Service Level Agreement). No commonly accepted standard exists for the format or content of cloud ToS and SLAs, nor any consensus about the expected security and privacy practices of service providers.

This poses difficulties for customers and providers alike, who have expectations (and duties) with regards to a given service offering. End-users require clarity and understanding on issues such as:

- how long a provider keeps data which has been stored or exchanged through its cloud services;
- how and when such data is destroyed;
- what remediation procedure exists in case of data loss and in case of data breach,
- to what extent data will be shared with parties external to the service provider and for what purpose (e.g. targeted advertising).

Enterprise customers typically require assurances regarding:

- service availability (e.g. estimated downtime per calendar month);
- cost of basic services versus added-value offerings;
- how data stored by a provider is kept isolated from other customers' data (particularly for multi-tenancy arrangements);
- encryption methods used, if any, and authentication technologies;
- backup methods and regularity of backup;
- remediation procedures and compensation offered in cases of data loss and data breach.

Although the field of cloud computing still lacks well-defined standards and best practices, they are actively being developed, and it is likely that cloud service providers will have a business need to adopt them in the future. This introduces another level of compliance and, unless cloud service providers are equipped with appropriate controls and tools, much manual effort may be required to achieve it.

There is also a need for tools that ensure what we might call **self-compliance**, namely compliance of a cloud service provider with its own stated policies. To date there is no obvious way of ensuring that the Terms of Service stated by cloud service providers are actually adhered to fully in practice.

We are interested in developing software tools to enable cloud service providers to be accountable with regards to their data governance practices. In this context, **accountability** refers to the goal of preventing harm to a cloud provider's customers by enforcing adequate protections on these customers' data, and having available effective reporting and auditing mechanisms (Ko et al., 2011a; Pearson and Charlesworth, 2009).

This paper presents ongoing work on developing software tools to automate compliance in the cloud, particularly natural-language processing of cloud terms of service; we show how such tools fit within a framework enabling cloud service providers to achieve accountability. Finally the paper identifies several classes of software tools to develop in the future, in order to further automate accountability in the cloud.

2. Techniques for Extracting and Enforcing Security and Privacy Rules In Cloud Computing Infrastructure: Previous and Related Work

We are working on tools to automate many of the processes required to ensure that a provider is accountable, although we recognise the difficulty of mapping and linking legal and regulatory requirements - which are high-level and expressed in natural language - to technically enforceable policies on particular data items.

Key techniques that should be used to achieve a significant degree of automation include:

- natural-language processing of laws and regulations
- knowledge extraction and learning from policy texts
- use of sticky policies
- automated policy enforcement
- active monitoring

In the next few sections we discuss each of these, surveying previous and related work in some detail.

Natural-Language Processing of Laws and Regulations

In particular, extraction of policy rules from legislative and regulatory texts and corporate rulebooks; these rules should be represented in a form that can be interpreted by a technical enforcement mechanism (esp. a Policy Enforcement Point or PEP), but possibly also so that they can be incorporated into a compliance checker of information governance software (cf. Governance / Risk Management Compliance (GRC) Platforms, widely used in industry). It should be noted here that no natural-language processing system can operate with 100% accuracy, but use of such systems can help to reduce significantly the overall amount of human intervention in the process of policy creation and management. In this paper we present two techniques involving natural-language processing, that we are currently investigating:

- automated information extraction
- segmentation and tagging of terms of service for decision support

Clearly it is unreasonable to expect a computer program to fully understand a legal or other policy text. However there are a variety of techniques and programs for analysing, annotating and extracting information from texts, and there have been various attempts at applying such techniques in the context of privacy. Furthermore there is much work on formalizing privacy and representing privacy-related properties in unambiguous logical form, which lends itself better to automated analysis. As for the enforcement of privacy rules and requirements, there exist rule-based systems that distil privacy knowledge into a form that can be executed directly by a machine. Of course, such systems have their limitations, but there are numerous practical benefits, especially as they reduce the effort required to ensure compliance significantly.

In the literature there is a tendency for the tasks and objectives in the previous paragraph (parsing and analysis of source texts, knowledge extraction, semantic representation, and enforcement of privacy rules) to be considered each in isolation. However these tasks may be seen as essentially interlinked processes in a privacy compliance lifecycle, so that the output of one task is the input of the next. This is a lifecycle because the processes need to be continually repeated in order to account for new privacy rules that emerge as societal needs, laws and regulations change. It should be noted here that there is some work being done in the European research project

CONSEQUENCE to automate the full lifecycle involved in managing data sharing agreements, from capturing agreements in a pseudo-language to translating them into enforceable policies; however the mapping from legal texts to this pseudo-language representation is a human process and does not seem to involve automated natural-language analysis.

First we consider practical approaches – namely, tools and techniques - to the task of natural-language processing of legal and regulatory texts. It is worthwhile to note that the main techniques for analysis of such texts tend to have many similarities across different domains, whether the texts refer to healthcare regulations, business best practice, or privacy rules: the common element is the prescriptive nature of the texts. In particular, texts that consist exclusively of detailed descriptions of rules often use standard sentence structures and patterns, which can be identified and formalized to a significant degree. What changes with different application domains, naturally, is the vocabulary, and the frequency of particular word clusters (see Stamey and Rossi (2009) for statistical results regarding the vocabulary and phrases common to privacy policies in particular). The papers by Moulin and Rousseau (2002), Michael, Ong and Rowe (2001), Brodie, Karat and Karat (2006) and the thesis of Ong (2001) all describe different tools for analysis of prescriptive texts, and we review these next. Also of note is work in the IBM REALM project.

Moulin and Rousseau (2002) describe a "knowledge acquisition system" known as SACD, implemented in Prolog, which has been developed for the analysis of regulatory texts. SACD was used to process the National Building Code of Canada, so that its stipulations could be represented in a machine-readable, indexable and searchable form. What is particularly interesting about SACD is that it can adapt the knowledge representation structures it uses automatically, as it processes input. Furthermore it provides a graphical user interface during the process of syntactic analysis, which allows for user intervention when a particular text fragment has been decomposed incorrectly or inaccurately. SACD makes use of chart-parsing algorithms and Prolog definite-clause grammars, which are ideally suited to the low-level analysis of sentence structure and meaning. The authors do mention the fundamental limitation of their approach, namely that the built-in grammars need to be repeatedly revised and extended to be able to parse new language elements, vocabulary and usage patterns. However, the system is capable of adapting its representations of knowledge, as text is parsed. Moulin and Rousseau's paper describes work that falls into almost all of the categories in our classification, including knowledge representation and learning; the link to compliance is mentioned, but the authors do not explain how it might be automatically achieved using their tool.

Michael, Ong and Rowe (2001) and Ong (2001) developed an architecture and concrete tool for analysing texts with prescriptive rules. Their approach alludes to knowledge extraction and logical representation of rules, but the tool they present is specifically an extractor which turns a prescriptive sentence into a 'meaning list'; how this meaning list can be used by a handler or enforcement mechanism is beyond the scope of their work. In terms of textual analysis, their approach is to use an off-the-shelf part of speech (POS) tagger and to process its output to determine whether the input sentence describes an obligation, permission or interdiction; the meaning list resulting from analysis of the sentence identifies the different actors involved and their interrelationships.

Brodie, Karat and Karat (2006) present a tool, SPARCLE, designed for authoring technically enforceable privacy policies using natural language. The tool is designed to parse and interpret English text describing privacy rules, and generate from that text appropriate XACML policies. In particular, the tool provides a structured policy authoring environment. What is appealing about SPARCLE is the ability to link access

control statements to the original natural language requirements; this aids both understanding and transparency.

Knowledge Extraction and Learning

Antón and a number of different collaborators (see Antón et al. (2004); Breaux and Antón (2005); Kiyavitskaya et al. (2007); Breaux, Vail and Antón (2006); Breaux and Antón (2008)) have used textual mining techniques to analyse privacy policies and a number of different privacy and privacy-related regulations. For example, in (Antón et al., 2004) the authors focus on privacy policies from financial institutions which claim to be compliant with the Gramm-Leach-Bliley Act (GLBA). The papers (Antón et al., 2004) and (Breaux and Antón, 2005) refer to the use of a tool called PGMT (Privacy Goal Management Tool), which is a tool for representing and analysing rules arising in privacy regulations as restricted natural-language statements. In (Kiyavitskaya et al., 2007) the authors discuss the extraction of structured rules from source texts using an NLP platform called *Cerno.*

Breaux, Vail and Antón (2006) presents those authors' approach of *semantic parameterization* to represent the US HIPAA (Health Insurance Portability and Accountability Act) Privacy Rule as a set of restricted natural-language statements, classified as rights, constraints or obligations. They identify standard phrases appearing in the legislative document, and note the frequency of their occurrence and the corresponding modality (right/obligation/interdiction etc.). They also discuss how to handle ambiguities. This work is extended further in (Breaux and Antón, 2008), where the authors develop a detailed classification of constraints and introduce means of handling complex cross-references arising in the legal text of the HIPAA.

Delannoy et al. (1993) combine a template-matching technique with machine learning in order to match rules from the Canadian 1991 tax guide with text describing case studies of particular individuals; this approach in principle allows one to see which tax rules apply in a given situation. The paper describes an architecture and tool called MaLTe, which is capable of learning how to apply rules to different input texts.

Delisle et al. (1994) describe in detail a framework for extracting meaning from the structure of technical documents. Their approach is relevant to the analysis of prescriptive texts in that they assume that input documents are highly structured and somewhat predictable. The authors propose a number of techniques for identifying patterns in texts and converting sentences to Horn clauses. The Horn clauses represent knowledge about the domain in question; through the use of machine learning techniques, this knowledge is extended and refined as more documents are supplied.

Stamey and Rossi (2009) use singular-value decomposition and latent semantic analysis techniques to analyse privacy policy texts. They identify commonly occurring topics and key terms and their relations. They are also able to detect similar word meanings; the strength of their approach is that they are able to pick out ambiguities in privacy policies and make them visible to the user. The tool *Hermes* developed by the authors allows automated analysis of an entire privacy policy text, outputting an overall ranking of the policy (when compared to a reference text).

Use of Sticky Policies

By strongly binding policies to the data they are associated with, it is easier for providers to control accesses to data within their cloud infrastructure and there is no need for a central policy repository. From the point of view of automating accountability, the use of sticky policies is a very useful technique. Sticky policies provide a means of data

encryption, since the data which a policy is bound to cannot be accessed unless that policy is complied with.

The central idea of this approach is as follows: end users allow service providers to have access to specific data based on agreed policies and by forcing interactions with specific certified system components (possibly with the involvement of interchangeable independent third parties called Trust Authorities). The access to data can be as fine-grained as necessary, based on policy definitions, underlying encryption mechanisms (supporting the stickiness of policies to the data) and a related key management approach that allows (sets of) data attribute(s) to be encrypted specifically based on the policy. By these means users can be provided with fine-grained control over access and usage of their data within service provider eco-systems, and an audit trail can be provided about usage and sharing of their data that can be inspected by the end users, and also potentially by other authorised parties such as regulators.

The original 'sticky policy' paradigm was espoused by (Karjoth et al., 2002), and specifies that privacy preferences should flow with personal data to make sure that they can always be enforced. But, no method for strong enforcement was suggested. A variety of techniques for binding data to disclosure policies specifying or constraining how it is to be used are possible, ranging from relatively weak logical bindings (for example, where the personal data is sent in clear and linked to the policies) to strong bindings that use cryptography to encrypt the data, and only provide the decryption key if the conditions specified by the preferences are verified (Pearson, Casassa Mont, and Kounga, 2011). Furthermore, the personal data and policies can be digitally signed to provide evidence about the conditions under which the data may be used.

Automated Policy Enforcement

The deployment of control points throughout a cloud provider's infrastructure where policy rules can automatically be enforced and human users only notified in case of failure or error is essential. We refer to the following current and future HP Labs European and TSB research projects for more related work on policy enforcement: EnCoRe (2011a), Information Stewardship in the Cloud (see Baldwin et al., (2011)) and Trust Domains (HP, 2012).

In previous work the authors have developed technical mechanisms for controlling the flow of data in an IT infrastructure, notably through the use of privacy controls (Casassa Mont et al., 2010), sticky policies (Pearson, Casassa Mont, and Kounga, 2011), and policy enforcement. Although the cited works do not specifically focus on cloud computing scenarios, we expect these techniques to be readily extendable and adaptable to suit the needs of a cloud service provider.

The EnCoRe research project (EnCoRe, 2011a) implements a platform for expressing and enforcing privacy preferences for personal data; recent case studies include a system for managing data held within an enterprise's HR systems, and health data stored about individuals and tissue samples in a biobank. Through the use of a suitable policy enforcement architecture, legal and regulatory privacy rules, along with individuals' privacy preferences, can be automatically enforced so that unauthorized and/or unsuitable access to data is prevented. In (Casassa Mont et al., 2010) we proposed a simple conceptual model for representing privacy rules, which can be directly mapped to technically enforceable access control policies (expressed e.g. using XACML).

Pearson et al. (2009) propose a tool for providing decision support with regards to privacy-sensitive projects that arise in an enterprise. Decision support systems are built on knowledge bases with rich sets of rules, and the process of translating legal texts, regulations and corporate guidelines into technically enforceable rules is complex and laborious. For this reason a conceptual model is a useful aid.

We are not aware of any previous work that addresses the whole lifecycle of natural-language analysis of privacy texts with the goal of enforcing suitable rules, e.g. in an enterprise setting (although the EU CONSEQUENCE project mentioned before does take an holistic approach it does not involve natural-language analysis). As stated in the Introduction, achieving compliance with privacy legislation and regulations is a central concern in enterprises, and means of automating compliance are highly desirable. Since new privacy rules are almost exclusively expressed using natural-language, means of automatically analysing the appropriate texts and extracting rules from them necessary – the resulting rules can then be incorporated into existing enterprise rule-bases, such as those used in compliance checkers or information governance (GRC) platforms.

Active Monitoring for Compliance

We believe that it is fundamental for cloud providers to have in their infrastructure mechanisms for automatically detecting compliance problems and potential sources of such problems. The most common mechanism for this purpose is *logging.* As part of the TrustCloud framework, we have been involved in the development of extensive logging tools for clouds.

Currently, there are only tools (e.g. HyTrust (2012)) which are able to log virtual level logs and system health monitoring tools for virtual machines. There is still a lack of transparency of (1) the linkages between the virtual and physical operating systems, (2) relationships between virtual locations and physical static server locations, and (3) how the files are written into both virtual and physical memory addresses. This information is currently not available as a single-point-of-view for the customers of cloud service providers. Ko, Jagadpramana, and Lee (2011) have developed a file-centric logger for monitoring file transactions in clouds known as Flogger to address this issue, as part of the lower layer of the TrustCloud architecture (Ko et al., 2011a).

3. Integrating the Approaches to Build an Automated Compliance Toolkit

The first contribution of this paper is the integration of the previously mentioned techniques into an integrated toolkit that automates the extraction and enforcement of

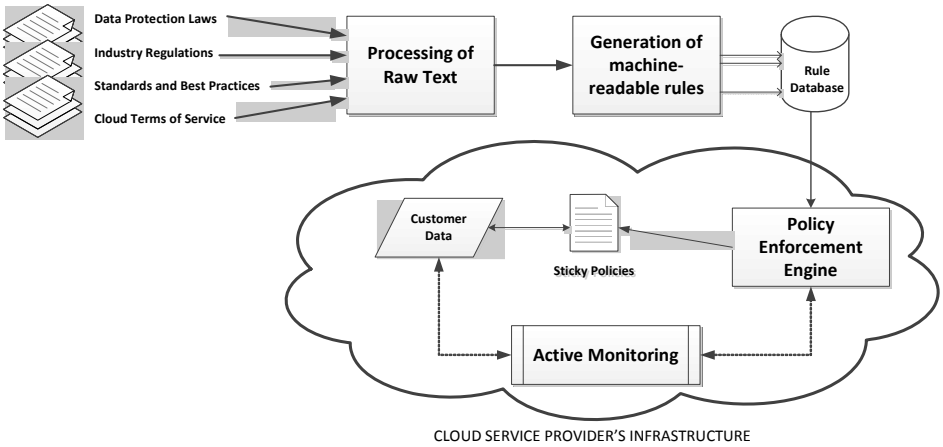


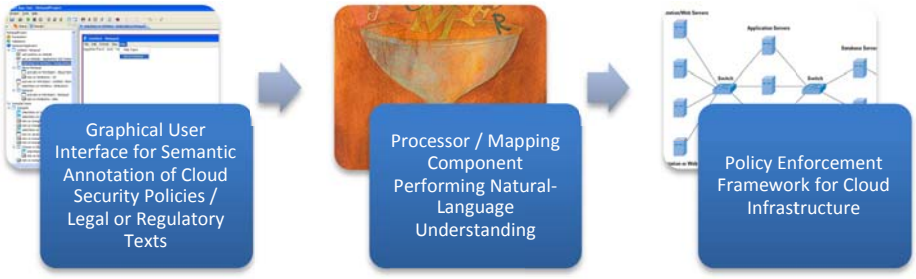
Figure 1. Integrating the approaches into an automated compliance framework: A high-level view.

privacy and security requirements in a cloud service providers' infrastructure.

Figure 1 depicts the integration of the techniques mentioned as a pipeline; each arrow shows a flow between processes. The processing of raw text describing laws, regulations, business rules and terms of service as well as the generation of machine-readable rules are to be typically performed outside the cloud service provider's infrastructure, while the resulting machine-readable rules are fed into the infrastructure to enforce appropriate control on the customers' data. The mechanisms of policy enforcement, and particularly the use of sticky policies which are attached to data, are to be implemented within the cloud service provider's infrastructure.

We are developing natural-language processing tools to analyze and extract information from legal and regulatory texts automatically. We focus on texts that specifically describe policies – rules, prohibitions, necessary measures that need to be put in place to provide assurance in cloud computing. In some cases, the texts to be analyzed are large pieces of legalese; this would include, for instance, national data protection legislation in Europe, or the Health Insurance and Portability and Accountability Act in USA. In the context of cloud computing, there are cloud security standards, as well as Terms of Service that are particularly worthy of analysis and mutual comparison.

Our toolkit comprises three parts, depicted in the figure below. We describe the functions of each component in turn, and detail the graphical semantic annotation tool in more detail in section 4.



Graphical Semantic Annotation Tool: We are currently developing a visual editor to enable domain experts in security and privacy to perform *semantic annotation* of texts containing rules that must be applied within a cloud infrastructure. Human experts are needed for this stage, even though the central objective of this work as a whole is to automate the analysis of texts as much as possible; the experts will use the editor to highlight and mark up portions of texts that are to be translated to machine-readable rules. This is the only stage where human interaction is required, and is essential in order to signpost the different parts of a rule (namely, the actor or subject of a rule, the action taken by the actor/obligation of the actor/restriction on the actor, the object of the rule, and the exceptions that apply).

It is important to note that the use of the editor is intended to attach semantic information to parts of the legal/ regulatory texts/cloud security policies in question, in a form that is understandable by the processor component, described next. The natural language processing algorithms that the processor implements handle most aspects of syntax, but assume that semantic information is also available.

Processor/Mapping Component: This component performs the 'natural-language understanding' and may be considered the 'intelligent' part of our solution. The automation argued for earlier in this paper is largely due to the functions of this component. The processor analyses source texts which contain security and privacy rules (assumed to be in ordinary English), detects patterns in the use of language that describe

typical security features of cloud services, and extracts from the text entities and relationships between them (the cloud service provider, third parties, components of infrastructure, mechanisms, practices that are described in the texts). We are currently developing this component by manually building a database of concepts and relationships that appear in cloud service providers' terms of service, but envisage the eventual use of machine learning algorithms to make an adaptive, self-modifying tool.

Policy Enforcement Framework: The final element of our solution is a policy enforcement framework, namely a system of Policy Decision and Policy Enforcement Points that can are deployed within a cloud service providers' infrastructure in order to implement the policy rules produced by the processor component above. Notice that the policies used in this framework are low-level, machine-readable policies expressed in a language such as XACML. The idea is that these policies will directly implement the rules coming out of the legal, regulatory and other texts that have been passed through the above components. Furthermore, note that there will be a significant number of security rules, dictated for example by the law or by cloud security standards, which map directly to simple access-control policies that can be directly enforced in the cloud infrastructure.

In section 4, we discuss the semantic annotation tool in detail. Section 5 details the EnCoRe policy enforcement framework, which implements the functionalities described above and integrates several of the approaches presented in section 2.

4. Implementation: A Software Tool For Semantic Annotation of Cloud Terms of Service

Figure 3 presents our current model for analysing cloud terms of service. We are developing a tool for marking up and extracting information from cloud terms of service, namely, the contract documents that describe a customer's relationship with a cloud service provider. Our tool is not fully automated as it requires, as a first step, a human user to indicate which sections of such documents describe which types of rules; this process is referred to as semantic annotation. Our tool provides a text editor with functions to highlight portions of text that describe restrictions, obligations, and other types of constraint with a particular colour. Output from the tool includes a marked-up version of the original contract, with semantic tags. This output can then be fed into a separate processor, which is work in progress, whose functions include information extraction and rule generation. These functions are described in more detail next.

Figure 3. Extracting and enforcing cloud terms of service using a semi-automated tool.

Automated Information Extraction

Key characteristics of cloud Terms of Service include:

- Cloud ToS are almost always formatted as rich-text web documents with headings and numbered paragraphs ("clauses" – in the legal sense, not the grammatical sense of the word).
- Significant portions of these texts contain disclaimers, enabling the service provider to refuse being held accountable in certain cases (these parts of the ToS actually

state what the provider will not be expected to do, rather than what the provider's actual practices are).

- If a service provider has several similar offerings (e.g. in the case of AWS) there will typically be two documents of interest – (i) a core agreement which sets out the main terms of service, and (ii) separate ToS for each of the different offerings (e.g. in the case of AWS offerings include: EC2, S3, EBS, SQS, SNS, SES, VPC, FWS, SimpleDB, GovCloud).

A recent legal research paper (Bradshaw, Millard, and Walden, 2010) documented the features and caveats of different cloud service level agreements, including discussions of both the general service descriptions and the terms and conditions available online.

While a cloud service provider may employ legal experts to draw up their terms and conditions in writing, it is the developers and system administrators that are responsible for making sure these terms are indeed enforced in the infrastructure used for a particular cloud offering. It is in the interest of the latter to have machine readable rules that are in a one-to-one correspondence with the statements made in the written ToS.

Natural-language analysis of the written ToS can certainly assist in the creation of such rules; if the written style of an ToS is very prescriptive, enforceable rules are easier to generate automatically. Otherwise human intervention will be required to ensure that generated rules are:

- **correct:** namely, that they express what actions a system needs to implement to make sure the requirements of the ToS are fulfilled on a constant basis;
- **as complete as possible:** namely, that the machine readable rules capture all those aspects of the ToS that can be enforced automatically.

We are not aware of any previous work that addresses the whole lifecycle of natural-language analysis of privacy texts with the goal of enforcing suitable rules, e.g. in an enterprise setting (although the EU CONSEQUENCE project mentioned before does take an holistic approach it does not involve natural-language analysis). As stated in the Introduction, achieving compliance with privacy legislation and regulations is a central concern in enterprises, and means of automating compliance are highly desirable. Since new privacy rules are almost exclusively expressed using natural-language, means of automatically analysing the appropriate texts and extracting rules from them necessary – the resulting rules can then be incorporated into existing enterprise rule-bases, such as those used in compliance checkers or information governance (GRC) platforms.

The most naïve analysis seeks to find in the text of an ToS occurrences of particular verbs, namely verbs which are prescriptive by nature; examples include:

"The Provider will provide a backup of data […]";
"The User will not upload pornographic images to the service"

since these typically arise in statements expressing duties and obligations (see also Breaux, Vail and Antón (2006)). Certain verb groups appear in phrases expressing rights, typically rights of the customer but not necessarily:

"The Customer may request in writing a full copy of data held [...]"
"The Provider can refuse to provide access to the service at any time [...]"

In the case of simple prescriptive sentences it is possible to represent the information given by a triple *(verb, subject, object).* In a Prolog program this would be declared as a Horn clause of the form

```
verb(subject, object).
```

Such a representation says nothing of the nature of the rule or (legal) clause appearing in the ToS, but may assist a service provider in automatically generating a set of access control rules for enforcement within its infrastructure. Our tool uses a form of markup referred to as a formal requirements specification language (RSL); the RSL we are using is due to Breaux and Gordon (2011).

Our tool is designed to detect delimiters and punctuation, so that long-winded sentences of legalese may be separated into their constituent parts. In a given sentence, those secondary clauses, which serve only to explicate and/or amplify the main thrust of the sentence, may be ignored (subject to interpretation and the judgment of a human user, of course; this suggests the process cannot be completely automated), and a semantic representation can be built of the remaining constituents of the sentence.

An interesting toolkit that we are considering to use in future work to automate part of this task is GATE ("General Architecture for Text Engineering") (Cunningham et al., 2011), whose user interface provides a helpful facility for tagging and colour-coding portions of text of particular semantic relevance. The technique that applies here is known as semantic annotation. We believe that such an approach is highly beneficial for the visual representation of the terms and conditions contained in a given cloud ToS.

5. Implementation: The EnCoRe Policy Enforcement Framework

EnCoRe (EnCoRe, 2011a; EnCoRe, 2011b) is a UK collaborative project that involves contributors from the social, legal and technological areas. EnCoRe aims at providing user-friendly and reliable consent and privacy management capabilities to individuals and organisations. Specifically, the objective of EnCoRe is to: provide data subjects with better control on their personal data once disclosed to organisations, by enabling explicit consent on how data should be handled, via the definition of privacy preferences and supporting later changes; enabling organisation to fully enforce these privacy preferences, along with security and privacy policies, inclusive of authorization and obligation policies.

The main area of technical innovation in EnCoRe consists of the overall end-to-end mechanisms and capabilities for consent and privacy management, spanning across the various stakeholders: users, organizations and third parties. These capabilities are provided by the EnCoRe D2.3 Technical Architecture (EnCoRe, 2011b) and related EnCoRe Framework including:

- explicitly capturing end-users (data subjects)' consent by means of privacy preferences;
- storing and processing preferences along with associated personal data;
- explicit representation and enforcement of privacy-aware authorization and obligation policies when handling data, accordingly to stated consent;
- tracking data whereabouts, within and across organizations;
- sharing personal data, across organizational boundaries in a secure and accountable way by leveraging the HP sticky policies approach and technology.

Figure 4 shows these key, privacy & data management capabilities provided by EnCoRe within and across organisations:

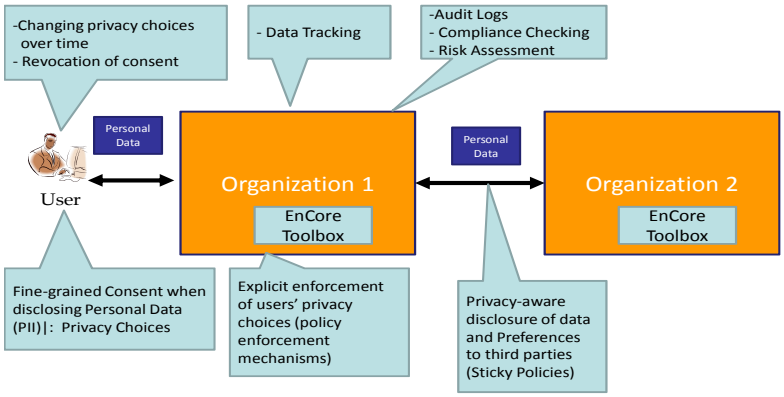


Figure 4. Key capabilities provided by EnCoRe.

The "EnCoRe Toolkit" refers to the set of technical EnCoRe capabilities. Figure 5 shows the high-level EnCoRe Technical Architecture underpinning this toolkit:

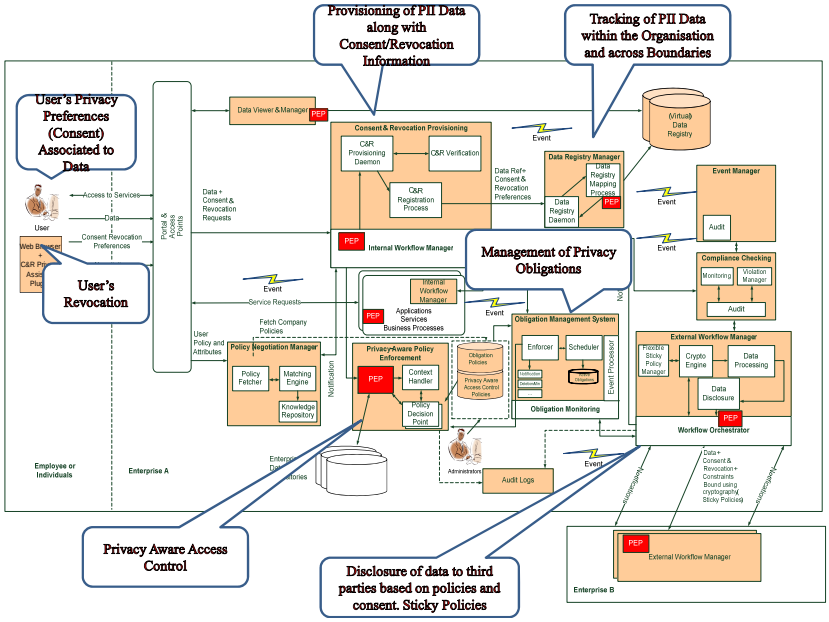


Figure 5. EnCoRe Technical Architecture.

Specifically, Figure 5 provides an overview of the key EnCoRe components, including:

- a user side plug-in to capture users' consent for specific personal data items, by means of privacy preferences;
- a back-end provisioning component, to store personal data, references to privacy preferences (in a data registry) and configure authorization policies for access control and obligations;
- a data registry to store the actual privacy preferences and the data whereabouts;
- a privacy-aware access control module to provide access control on data, driven by security & privacy constraints and preferences;
- an obligation policy module to deal with duties dictated by privacy preferences (e.g. data deletion, notification, data minimization, etc.);
- an external workflow manager to handle interactions with third parties and exchange personal data, along with consent using the sticky policy mechanism;
- auditing/logging capabilities.

These components have been designed and implemented as independent, configurable services: they support secure communication and audit/logging capabilities. They can be flexibly deployed within an organization and at the end-user site, depending on needs and leveraging existing middleware, such as Identity and Access Management (IAM) solutions.

It is important to notice that the EnCoRe Framework implements explicit capabilities to handle (privacy-aware) authorization policies for access control and obligation policies. These policies (EnCoRe, 2011b) are flexible and support a representation of the authorization and obligation policies in a way that can be enforced and monitored.

The authorization and obligation constraints are translated into an internal programmatic representation, based on XML (W3C, 2008), which captures the various conditions along with references to data items. This includes:

- Constraints dictating agreed purposes for accessing and disclosing data;
- Constraints on which entities can/cannot access the data and or data can/cannot be disclosed to;
- Constraints on deletion, notification, data minimisation, etc.

An example of EnCoRe authorization policy for access control follows:

```
<policy>
    <name>policy-ID1</name>
    <target>Credit Card</target>
    <trigger>
            <expression>
                    <and>
                    <condition>Request.Obj.Location=="PII.DB"</condition>
                    <condition>Context.Request.Purpose contained in
                            Context.PrivacyPreferences.Purpose
                    </condition>
                    </and>
            </expression>
    </trigger>
    <rules>
            <rule>
                    <expression>
                            <and>
                            <condition>Request.ThirdPartyDisclosure ==
```

```
                        Context.PrivacyPreferences.AllowedThirdParties
                        </condition>
                        <condition>Request.Purpose ==
                        "Business_Transaction"
                        </condition>
                        </and>
                </expression>
                <action>
                        <decision>yes</decision>
                </action>
        </rule>
    </rules>
</policy>
```

This authorization policy is a general purpose policy; this is an example of a specific policy statement:

> *"My credit card data can be shared with Service Provider A and Service Provider B for business transaction purposes."*

The EnCoRe system, at enforcement time, knows (for each data subject) what the specified (privacy) preferences are, including preferences in terms of purposes, third parties where data can/cannot be disclosed to, etc. This information is available as contextual information.

An example of an EnCoRe obligation policy is as follows:

```
<obligation>
        <name>obligation-ID1</name>
        <type>one-off</type>
        <target>attributes</target>
        <eventList>
                <event>Event_Access_Granted</event>
        </eventList>
        <trigger>
                <expression>
                        <and>
                        <condition>Request.Obj.Location=="PII.DB"</condition>
                        <condition>Context.PrivacyPreferences.Notify==YES
                        </condition>
                        </and>
                </expression>
        </trigger>
        <actions>
                <action>
                        <do>send notification to data subjects</do>
                        <onViolation>Log error</onViolation>
                </action>
        </actions>
</obligation>
```

This obligation policy can be used to represent the following obligation:

> *"I want to be notified by email every time my data is accessed."*

At enforcement time, EnCoRe detects when the end-users (data subjects)' data is accessed and, in such a case, it sends notifications to data subjects, if they needed to. A

detailed description of the representation of EnCoRe policies and their enforcement capabilities is discussed in (EnCoRe, 2011b).

The EnCoRe Framework supports four common use cases:

- *An end-user discloses personal data along with consent/privacy preferences*: the system captures these via user-side plug-ins; the information is sent to the back-end provisioning component for internal configuration (via policies and the data registry). This includes setting privacy obligations in the privacy obligation manager, driven by user preferences, e.g. about notifications (on usage/disclosure of data), data deletion, etc.;
- *Employees and/or applications trying to access data for specific purposes* (e.g. marketing, transaction processing, research, etc.): the privacy-aware access control module intercepts these requests (via SQL interception and/or specific interception points within applications) and grants/denies access based on the evaluation of authorization policies for access control. These policies not only describe security constraints (who can access what) but also privacy constraints based on users' preferences (e.g. purposes for using data, black/white lists of entities that can handle data, etc.);
- *End-user changes his consent/privacy preferences*: the end-user can change, at any time, their privacy preferences. This triggers a chain of updates of stored privacy preferences within the organisation (via the back-end Service Framework), including updates of the data registry, authorization policies for access control and obligations. If the updated preferences relate to data shared with third parties, these parties will also receive update notifications;
- *Personal data is disclosed to a third party*: the system intercepts the attempt of applications to disclose data to third parties (via locally deployed agents). If the transfer of data is authorized by the access control component, then the personal data is disclosed to the third party via the external workflow manager, by using the sticky policy mechanisms that bundle data to policies and privacy preferences [9]. The degree of stickiness (simple association or strong cryptographic binding) can be configured. The data registry is updated accordingly about the data whereabouts.

Various use cases carried out with customers and government organisations demonstrated that the EnCoRe framework and solution can be easily integrated with existing enterprise data management and Identity & Access Management (IAM) solutions, for example by leveraging IAM Provisioning solutions.

6. Future Work: New Classes of Tools for Automating Compliance in the Cloud

Here we discuss applications of the above techniques and particularly, what other types of tools need to be developed to improve compliance in cloud computing.

Decision Support Tools

We believe that being able to efficiently (and automatically) extract security and privacy stipulations from cloud ToS is also a key business advantage, enabling decision support in enterprises for the selection of cloud services and providers as necessary during the course of their daily operations. We are looking to develop new tools for risk management in the cloud as part of the forthcoming European project A4Cloud, as well as linking our current toolkit with existing decision support systems.

Logging Tools for Tracing File Accesses in and Across Clouds

In Ko, Lee and Pearson (2011b), a framework for building tools that assist cloud service providers in achieving accountability and auditability is proposed. The emphasis is on logging mechanisms that allow file accesses in clouds to be traced and actively monitored. File loggers for cloud service providers are meant to support what the authors refer to as the Cloud Accountability Life Cycle, which consists of the following phases:

Policy Planning: In the beginning, CSPs have to decide what information to log and which events to log on-the-fly. There are typically four important groups of data that must be logged: *event data* – a sequence of activities and relevant information, *actor data* – the person or computer component (e.g. worm) which trigger the event, *timestamp data* – the time and date the event took place, and *location data* – both virtual and physical (network, memory, etc) server addresses at which the event took place.

Sensing and Tracing: The main aim of this phase is to act as a sensor and to trigger logging whenever an expected phenomenon occurs in the CSP's cloud (in real time). Accountability tools need to be able to track from the lowest-level system read/write calls all the way to the irregularities of high-level workflows hosted in virtual machines in disparate physical servers and locations. Also, there is a need to trace the routes of the network packets within the cloud.

Logging: File-centric perspective logging is performed on both virtual and physical layers in the cloud. Considerations include the lifespan of the logs within the cloud, the detail of data to be logged and the location of storage of the logs.

Safe-keeping of Logs: After logging is done, we need to protect the integrity of the logs prevent unauthorized access and ensure that they are tamper-free. Encryption may be applied to protect the logs. There should also be mechanisms to ensure proper backing up of logs and prevent loss or corruption of logs. Pseudonymisation of sensitive data within the logs may in some cases be appropriate.

Reporting and Replaying: Reporting tools generate from logs file-centric summaries and reports of the audit trails, access history of files and the life cycle of files in the cloud. Suspected irregularities are also flagged to the end-user. Reports cover a large scope: virtual and physical server histories within the cloud; from OS-level read/write operations of sensitive data to high-level workflow audit trails.

Auditing: Logs and reports are checked and potential fraud-causing loopholes highlighted. The checking can be performed by auditors or stakeholders. If automated, the process of auditing will become 'enforcement'. Automated enforcement is very feasible for the massive cloud environment, enabling cloud system administrators and end-users to detect irregularities more efficiently.

Optimising and Rectifying: Problem areas and security loopholes in the cloud are removed or rectified and control and governance of the cloud processes are improved.

Software Tools for Visualising and Understanding Policies

It has often been noted that presenting privacy policies and similar documents describing terms and conditions directly to end-users rarely draws their attention, and often users tend to click through any agreements of this sort if they require access to a service, thus ignoring details which could have significant consequences to them and their data. Since cloud services are almost exclusively purchased online, and terms and conditions are always presented on-screen to users, it is unlikely that customers of these services will pay due attention to the fine print; we believe that security and privacy policies should be presented in a more visually appealing fashion, which aids comprehension and allows users to compare competitors' data handling practices. This idea is certainly not new, and several previous authors have developed and demonstrated ways to help users visualise and understand terms and conditions; the P3P policy language (Cranor et al., 2002) was designed in part to allow the development of visual tools to understand privacy policies. Research projects such as PRIME, PrimeLife, and EnCoRe have developed user interfaces and dashboards for privacy settings and preferences. Clearly these efforts need to continue and be extended to applications specific to cloud computing.

Through analysis of cloud ToS, it should certainly be possible to generate comprehensible visual representations of a service providers' security and privacy practices. Of course, unless such representations are standardised, this task will be non-trivial.

Software Tools for Checking Compliance of Cloud Terms of Service with Prevailing Laws, Regulations and Standards

Cloud service providers are likely to audit their systems on a regular basis to ensure that their policies are valid and conform to current law, standards and best practices, adapting ToS and actual practices as necessary.

From this perspective, natural-language analysis can be used to extract rules from legislation and standards; these rules can then be compared and contrasted to ToS rules, triggering changes and extensions as required.

The extraction and representation of policy rules can then be seen as but the first part in a larger process or lifecycle. ToS have to be maintained, adapted, enforced, and audited. One can envisage how metrics for similarity of ToS can be defined or other measures for determining the degree of compliance to a particular industry best practice. This is clearly a very promising direction for investigation, with important implications for enterprises.

Software Tools for Generating Model or Template Cloud Terms of Service

Natural-language analysis of cloud ToS can help to detect language patterns that are common to such texts. This could be extremely useful in designing templates or 'model ToS'. To have industry agreement on what constitutes a model ToS would be an important step for cloud computing, and hopefully pave the way for the establishment of standard policies and commonly agreed security levels.

Taking this further, it is possible to develop natural-language generation tools which mechanically produce the text of cloud ToS for particular applications. If standards were to be established for the security levels specified by cloud ToS, the format and content of these documents would be well-established, making document generation significantly automatable.

7. Conclusions

We believe that it is beneficial and possible for cloud service providers to automate a number of tasks related to the requirement of accountability. We have identified some specific techniques, namely: natural-language analysis of law, regulation and corporate guidelines on security and privacy of customer data in order to generate technically enforceable policies; use of sticky policies to achieve a strong binding between data and the stipulations that apply to the use and dissemination of that data; and active monitoring of a cloud provider's infrastructure to detect potential compliance problems. More in-depth analyses of ways to achieve accountability in the cloud are available in some of our previous work (see also Casassa Mont et al. (2010); Ko et al. (2011); Ko, Lee and Pearson (2011); Mowbray, Pearson and Shen (2010); Pearson (2011); Pearson, Casassa Mont, and Kounga (2011)).

Our main contribution in this paper has been to describe our current work on developing software tools for automated information extraction of cloud terms of service, and to identify classes of related software tools needed to achieve full accountability in cloud computing. There is clearly much work to be done to achieve this important goal for the sake of future cloud service users.

References

Antón, A. Earp, J.B., He, Q., Stufflebeam, W., Bolchini, D., and Jensen, C. (2004) 'Financial Privacy Policies and the Need for Standardization', in *IEEE Security and Privacy* **2**(2), pp. 36-45.

Baldwin, A., Pym, D., Sadler, M. and Shiu, S. (2011) 'Information Stewardship in Cloud Ecosystems: Towards Models, Economics, and Delivery' in *Proceedings of the Third IEEE International Conference on Cloud Computing Technology and Science (CloudCom 2011)*.

Bradshaw, S., Millard, C., and Walden, I. (2010) *Contracts for Clouds: Comparison and Analysis of the Terms and Conditions of Cloud Computing Services.* Queen Mary University of London, School of Law Legal Studies Research Paper No. 63/2010. Available from http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1662374 (Accessed 14 May 2012).

Breaux, T. D., Vail, M.W., and Antón, A.I. (2006) 'Towards Regulatory Compliance: Extracting Rights and Obligations to Align Requirements with Regulations.' in *Proceedings of 14th IEEE International Requirements Engineering Conference (RE'06)*.

Breaux, T.D. and Antón, A.I. (2005) 'Deriving Semantic Models from Privacy Policies', in *Proceedings of the Sixth International Workshop on Policies for Distributed Systems and Networks (POLICY'05)*.

Breaux, T.D. and Antón, A.I. (2008) 'Analyzing Regulatory Rules for Privacy and Security Requirements', in *IEEE Transactions on Software Engineering* **34**(1), pp. 5-20.

Breaux, T.D., and Gordon, D.G. (2011) *Regulatory Requirements as Open Systems: Structures, Patterns and Metrics for the Design of Formal Requirements Specifications.* Technical Report CMU-ISR-11-100, Institute for Software Research, Carnegie-Mellon University. Available from reports-archive.adm.cs.cmu.edu/anon/isr2011/CMU-ISR-11-100.pdf (Accessed 14 May 2012).

Breaux, T.D., Vail, M.W., and Antón, A.I. (2006) 'Towards Regulatory Compliance: Extracting Rights and Obligations to Align Requirements with Regulations', in *Proceedings of 14th IEEE International Requirements Engineering Conference (RE'06)*.

Brodie C. A., Karat C., Karat J. (2006) 'An Empirical Study of Natural Language Parsing of Privacy Policy Rules Using the SPARCLE Policy Workbench', in *Proceedings of Symposium on Usable Privacy and Security (SOUPS)*.

Casassa Mont, M., Pearson, S., Creese, S., Goldsmith, M., and Papanikolaou, N. (2010) 'A Conceptual Model for Privacy Policies with Consent and Revocation Requirements' in Proceedings

of PrimeLife/IFIP Summer School 2010: Privacy and Identity Management for Life, Lecture Notes in Computer Science, Springer (2010).

Casassa Mont, M., Pearson, S., Creese, S., Goldsmith, M., and Papanikolaou, N. (2010) 'A Conceptual Model for Privacy Policies with Consent and Revocation Requirements', in *Proceedings of PrimeLife/IFIP Summer School 2010: Privacy and Identity Management for Life*, Lecture Notes in Computer Science, Springer.

Cranor, L., Langheinrich, M., Marchiori, M., Presler-Marshall, M., and Reagle, J. (2002) *The Platform for Privacy Preferences 1.0 (P3P1.0) Specification*. W3C Recommendation. Available from http://www.w3.org/TR/P3P/ (Accessed 14 May 2012).

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M.A., Saggion, H., Petrak, J., Li, Y., and Peters, W. (2011) *Text Processing with GATE (Version 6)*. Department of Computer Science, University of Sheffield. ISBN 978-0956599315.

Delannoy, J. F., Feng, C., Matwin, S. and Szpakowicz, S (1993) 'Knowledge Extraction from Text: Machine Learning for Text-to-rule Translation', in *Proceedings of Machine Learning and Text Analysis Workshop (ECML-93).*

Delisle, S., Barker, K., Delannoy, J., Matwin, S., Szpakowicz, S. (1994) 'From Text to Horn Clauses: Combining Linguistic Analysis and Machine Learning', in *Proceedings of Canadian AI Conference (AI/GI/CV '94).*

EnCoRe (2011a). The EnCoRe project. See http://www.encore-project.info/.

EnCoRe (2011b). The EnCoRe Technical Architecture D2.3, http://www.encore-project.info/deliverables_material/D2_3_EnCoRe_Architecture_V1.0.pdf.

HP (2012). The TrustDomains Project (Brochure).
See http://www.hpl.hp.com/research/cloud_security/TrustDomains.pdf/.

HyTrust (2012). "HyTrust Appliance" See http://www.hytrust.com/product/overview/.

Karjoth, G., Schunter, M., Waidner, M. (2002) 'Platform for Enterprise Privacy Practices: Privacy-Enabled Management of Customer Data', in *Proceedings of the 2nd Workshop Privacy Enhancing Technologies (PET 02)*, LNCS 2482, Springer, pp. 69-84.

Ko, R.K.L, Jagadpramana, P., Mowbray, M., Pearson, S., Kirchberg, M., Liang, Q., and Lee, B.S. (2011a) 'TrustCloud: A Framework for Accountability and Trust in Cloud Computing', paper presented at 2nd IEEE Cloud Forum for Practitioners (ICFP), IEEE Computer Society, Washington DC, USA.

Ko, R.K.L., Jagadpramana, P., and Lee, B.S. (2011). 'Flogger: A File-Centric Logger for Monitoring File Access and Transfers within Cloud Computing Environments', Technical Report HPL-2011-119, Hewlett Packard Laboratories. Available at http://www.hpl.hp.com/techreports/2011/HPL-2011-119.pdf.

Ko, R.K.L., Lee, B. S., and Pearson, S. (2011b) 'Towards achieving accountability, auditability and trust in cloud computing.', in A. Abraham et al. (Eds.), ACC 2011, Part IV, CCIS 193, pp. 432–444, Springer-Verlag, Heidelberg.

May, M., Gunter, C., Lee, I., and Zdancewic, S. (2009) 'Strong and Weak Policy Relations.' in *Proceedings of the 2009 IEEE International Symposium on Policies for Distributed Systems and Networks (POLICY '09).* IEEE Computer Society, Washington, DC, USA, pp. 33-36.

Mell, P., and Grance, T. (2011) *The NIST Definition of Cloud Computing: Recommendations of the National Institute of Standards and Technology.* NIST Special Publication 800-145. Available from http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf (Accessed 14 May 2012).

Michael, J. B., Ong, V. and Rowe, N.C. (2001) 'Natural-Language Processing Support for Developing Policy-Governed Software Systems', in *Proceedings of 39th International Conference and Exhibition on Technology of Object-Oriented Languages and Systems (TOOLS 39)*, pp. 263—274.

Moulin, B. and Rousseau, D. (2002) 'Automated Knowledge Acquisition from Regulatory Texts', in *IEEE Expert* **7**(5), pp. 27-35.

Mowbray, M., Pearson, S. and Shen, Y. (2010) 'Enhancing privacy in cloud computing via policy-based obfuscation.' *Journal of Supercomputing*. DOI: 10.1007/s11227-010-0425-z.

N. Kiyavitskaya, N. Zeni, T. D. Breaux, A. I. Antón, J. R. Cordy, L. Mich, J. Mylopoulos (2007) 'Extracting Rights and Obligations from Regulations: Toward a Tool-Supported Process', in *Proceedings of ASE'07*.

Ong, V. L. (2001), *An Architecture and Prototype System for Automatically Processing Natural-Language Statements of Policy*. Master's thesis, Naval Postgraduate School, Monterey, California.

Papanikolaou, N., Creese, S., and Goldsmith, M. (2011) 'Refinement checking for privacy policies.' *Science of Computer Programming*. Article in Press, DOI:10.1016/j.scico.2011.07.009.

Pearson, S. (2011) 'Toward Accountability in the Cloud', *IEEE Internet Computing*, IEEE Computer Society, July/August issue, vol. 15, no. 4.

Pearson, S., Casassa Mont, M., Chen, L., and Reed, A. (2011) 'End-to-End Policy-Based Encryption and Management of Data in the Cloud', in *Proceedings of CloudCom 2011: 3rd IEEE International Conference on Cloud Computing Technology and Science*, Athens, Greece, pp. 764-771. DOI: 10.1109/CloudCom.2011.118.

Pearson, S., Casassa Mont, M., and Kounga, G. (2011) 'Enhancing Accountability in the Cloud via Sticky Policies.' in *Secure and Trust Computing, Data Management and Applications*, Communications in Computer and Information Science, vol. 187, Springer Verlag, Heidelberg, pp. 146-155.

Pearson, S. and Charlesworth, A. (2009) 'Accountability as a Way Forward for Privacy Protection in the Cloud', in *Proc. 1st CloudCom 2009*, M.G. Jaatun, G. Zhao, C. Rong (eds.), Beijing, Springer LNCS 5931, pp. 131-144, December 2009.

Pearson, S., Rao, P., Sander, T., Parry, A., Paull, A., Patruni, S., Dandamudi-Ratnakar, V., Sharma, P. (2009) 'Scalable, accountable privacy management for large organizations', in *Proceedings of 13th Enterprise Distributed Object Computing Conference Workshop (EDOCW 2009)*, pp. 168-175.

Stamey J. W., and Rossi R. A. (2009) 'Automatically Identifying Relations in Privacy Policies', in *Proceedings of SIGDOC'09*.

W3C (2008). Extensible Markup Language (XML) 1.0. Available at http://www.w3.org/TR/REC-xml/.