Journal of King Saud University – Computer and Information Sciences (2015) 27, 364–375

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

Building an Islamic financial information system based on policy managements

Izzat Alsmadi [a,*], Mohammad Zarour [b]

[a] Computer Science Department, Boise State University, 1910 University Drive, Boise, ID 83725, United States
[b] College of Computer Science and IT, Prince Sultan University, P.O. Box 66833, Rafha Street, Riyadh 11586, Saudi Arabia

Received 29 March 2014; revised 27 June 2014; accepted 6 November 2014
Available online 20 June 2015

KEYWORDS

Islamic financial systems;
Islamic banking;
Policy managements;
XACML

Abstract For many banks and customers in the Middle East and Islamic world, the availability and the ability to apply Islamic Shariah rules on financial activities is very important. In some cases, business and technical barriers can limit the ability to apply and offer financial services that are implemented according to Shariah rules.

In this paper, we discuss enforcing Shariah rules from information technology viewpoint and show how such rules can be implemented and enforced in a financial establishment. Security authorization standard XACML is extended to consider Shariah rules. In this research XACML architecture, that is used and applied in many tools and system architectures, is used to enforce Shariah rules in the banking sector rather than its original goal of enforcing security rules where policy management systems such as XACML are usually used.

We developed a model based on XACML policy management to show how an Islamic financial information system can be used to make decisions for day to day bank activities. Such a system is required by all Islamic banks around the world. Currently, most Islamic banks use advisory boards to provide opinions on general activities. The gap between those high level general rules and decision for each customer business process is to be filled by Islamic financial information systems.

The flexible design of the architecture can also be effective where rules can be screened and revisited often without the need to restructure the authorization system implemented. Authorization rules described here are not necessarily the perfect reflection of Shariah opinions. They are only shown as a proof of concept and a demonstration of how such rules can be written and implemented.

* Corresponding author.
E-mail addresses: izzatalsmadi@boisestate.edu (I. Alsmadi), mzarour@psu.edu.sa (M. Zarour).

Peer review under responsibility of King Saud University.

ELSEVIER Production and hosting by Elsevier

1. Introduction

According to Islamic Shariah, there are certain rules and regulations that should control financial activities between money lenders and borrowers before making it legal from religious perspectives to prevent Riba (El-Gamal, 2000; Ahmad, 1995). Recently, Islamic banking witnessed huge investments

http://dx.doi.org/10.1016/j.jksuci.2014.11.001
1319-1578 © 2015 The Authors. Production and hosting by Elsevier B.V. on behalf of King Saud University.

worldwide (McLean, 2013). Islamic banking is not limited to Islamic countries but spread over other countries such as the United States, Europe and the Far East (Imran et al., 2011). The customers of Islamic banks are not limited to Muslim ones only, they are well-received by non-Muslims as well, see for example Abdullah et al. (2012). This can be referred to the fact that Islamic banking through its various products and services promotes equity (Dhumale and Sapcanin, 1999).

Despite the widespread of Islamic banking and its services, nowadays, Islamic banking is facing several challenges, some of them are technology independent that are related to regulatory issues, awareness and Shariah aspects, see for example Al-Omar and Iqbal (1999), Akhtar and Talreja (2012) and Karbhari et al. (2004). Other challenges are related to how to enforce Shariah rules in the day-to-day banking operations.

As stated in ITS (2011) "One of the greatest challenges in launching an Islamic bank is having access to a well-established independent and active Shariah board. While this process can be rapidly increased by utilizing already existing Shariah boards, it is the implementation of the Shariah board's rulings into the day-to-day running of banking operations, product development and product rollout that serves as the greatest hurdle to Shariah compliant operations and transformation".

Accordingly, one of the main challenges is related to the nature of such rules and how to present them to the banking or financial systems in a proper format. This can be related to the different levels of abstraction between religious codes of conducts and what is permitted or not. A second challenge which is also related to the different religious opinions is particularly related to some transactions and whether they are religiously legal or not and how could such rules be applied to real cases. Those for example can receive conflicting opinions from the different scholars. For more information about Islamic financial products and challenges facing Islamic banking see El-Gamal (2000), McLean (2013), Imran et al. (2011), Siddiqi (2006), Hassan et al. (2013) and Hassan and Lewis (2007).

This research paper is focusing on another dimension (i.e. third challenge) that is related to technical challenges of implementing or enforcing Shariah rules in financial institutes' daily activities. Few research contributions were conducted to take this to technical levels not only from financial perspectives, but also from information technology perspectives. For example, it is important to automate or semi-automate banking systems, with little human intervention to understand Shariah rules, how can these rules be implemented and where should they be applied.

Focusing on the last technical challenge, an information system (proposed) is then expected to handle these three obstacles or challenges to be a good candidate solution.

In this context, this research work proposes using XACML (the Extensible Access Control Markup Language) from OASIS Committee (2013) to enforce Shariah rules in the banking sector. XACML is a standard for access control policy implementation and management.

The remainder of this paper is organized as follows: Section 2 presents briefly the concept of Islamic banking or finance. Section 3 illustrates the XACML language and its architecture. Section 4 presents the related work. Section 5 discusses the use of XACML to enforce Islamic policies. Section 6 presents the conclusion and future work.

2. Islamic banking or finance

The general definition of Islamic banking or finance is that the banking or financial system that is abided by Islamic financial rules or rules that are related to monetary issues. For example, Islamic Sharia prohibits what is called "Riba". This is where the money lender has a fixed, inflating or floating amount of money on the money borrower. This is particularly forbidden when money is borrowed and returned. This can be applied to gold or silver where the same type is borrowed and returned. Sharia then prohibits this loan process. Modharaba in Sharia is the process where the money lender is subjected to loss and gain and the process does not always guarantee lender gain in all scenarios. This is then called a risk-sharing rather than risk-free transaction.

Sharia also prohibits money inflation (e.g. accumulative interest rate) where the loan amount increases if the borrower delayed the payment process. The added amount to the loan should be fixed through the whole period.

There have been some trials to generalize Islamic banking in comparison with traditional banking. For example, an article posted in the website (AlBaraka Bank Group) indicated four principles for Islamic banking: Prohibition of Interest or Usury, Ethical Standards, Moral and Social Values, and Liability and Business Risk. Most Islamic banking systems or entities include policy guidelines on how to deal with making sure that transactions are conducted according to Islamic financial rules. See for example, Islamic Financial rules of Dubai Financial Services Authority (DFSA) (The Islamic Finance Rules (IFR), 2013).

Islamic banks in Muslim countries claim to follow Sharia laws and guidelines. However such processes cannot be audited or verified on a daily basis or for each transaction since such processes depend on human domain experts or on those who work as religious advisors for the banks. Hence there is a serious need to build a knowledge management system through which all Sharia rules can be documented, interrogated, evaluated, etc. Our proposed Sharia policy management system can be either part of the Islamic finance knowledge management system or one of its components.

As the first step in the Islamic policy system, we will describe the major general agreed upon financial processes (in the following subsections) that are currently implemented by most Islamic banks. Those high level concepts will be used as parents or even grandparents of the policy management systems. In policy terms, those will be considered as (Policy_Sets) where many policies can be generated as children of those that we will call for now policies. Names and details of the following Islamic financial activities are taken from a wide range of Islamic resources through the Internet, see for example Islamic Development Bank and Institute of Islamic Banking and Insurance, where the first website is for the Islamic development bank that is established from many countries to sponsor Islamic banking and finance in general and the second website is for a non-profit organization established in UK with the general goal of establishing an Islamic banking system. The

website or organization has its own magazine (New Horizon Magazine) and books published in this specific field.

2.1. Modharaba or profit sharing

A basic principle or even policy that many of those policies are extracted from is that in Islamic money cannot be borrowed for money. In Modharaba, both the bank and the customer can invest money with the other side or partner (e.g. the bank or the customers). The business process will then be subjected to gain and loss and each partner will be affected by either case. Usually the money investor partner will lose the money and the other partner will lose the effort or time.

2.2. Mosharaka or joint venture

This is somewhat similar to profit sharing where investors can contribute to a new business with their money, support, place, equipment and the other partner with the idea, the effort, etc. The main idea from Islamic Sharia perspective is that both partners are subjected to winning or losing money (aka risk sharing rather than risk free for one partner).

2.3. Murabaha or cost plus

In most cases Murabaha refers to the cases where customers decide to invest their money with the bank. The bank can use the money for building houses, buying lands, establishing businesses, etc. The profit or loss that comes from such investment can then be shared with the customers based on their investment amount.

2.4. Wadeea'a or safe keeping

In those cases, the customers hold their money in the bank as Wadeea'a for safe keeping. The amount of money is fixed and does not inflate with interest as in other typical banking systems (aka checking not saving accounts).

2.5. Ijar or leasing that ends with ownership

This is usually used when customers buy apartments or houses. A partnership is formed between both partners where each one will pay a certain amount of money to buy the apartment or the house. The bank usually pays the majority of the capital and hence the customer pays money to the bank for renting of their part of the apartment. The rent amount keeps decreasing as the customer share keeps increasing with the bank share decreasing. Some Islamic banks or scholars may still not agree on this process or some of its details.

2.6. Gardh Hassan or free loan

Some banks offer to their customers loans with no interest at all. In some cases, processing fees are accepted as they are not considered as add-ons on the capital but as loan processing fees.

2.7. Bai-Assalam

A contract is made between a buyer and a seller. The buyer pays money in advance for some goods that will be received later on. This is usually applied to crops. For the process to be Islamically legal, the sold items should be specified in detail.

2.8. Rent or Ijar

Islamic laws legalize rent with conditions that rent should specify not only the payment amount but should also specify the period. The rent can be not only for a house, car, land, apartment, but it can be also for a service or for using some equipment or material.

2.9. Sokook or Islamic bonds

Sokook are financial certificates that have no interest. The money in such a process can be utilized under one of the legalized activities mentioned earlier.

2.10. General loans

Islam legalizes loans in general within certain conditions. As mentioned earlier money cannot be borrowed and returned as money. Same thing is applied for gold or silver. Typically if a customer for example decides to take a loan from an Islamic bank to buy a car, the bank should own the car first and pay its capital completely. The customer can then buy the car from the bank under a typical loan process. Further interest is fixed and decided at once and cannot then be changed. Nonetheless, most Islamic banks recently modified this scenario and allowed customers to refinance in certain times and under certain conditions.

3. XACML

XACML is an XML-based standardized language which is developed to replace application specific and proprietary access control policy languages (Liu et al., 2011). It is also considered a security policy creation and management application. XACML includes components to define a security policy to access computer resources (e.g. a data base, an application, and a web service). It also includes rules to specify users and their permissions or privileges. Fig. 1 shows XACML authorization elements including: Policy component, policy set, policy, policy administration point, rules, targets, actions, resources, subjects and environments. Details on those components and their rules can be found in XACML documentations. We will, later on, describe these components with a context example related to the paper subject.

Fig. 2 below shows a context diagram for XACML showing its major architectural components. The figure shows that XACML develop, regulate, implement and test rules through four components: PAP, PDP, PEP, PIP.

1. Policy Administration Point (PAP). This includes the management component that includes policies' repository. Different rules can be written in one or more policies that are stored and managed by PAP.

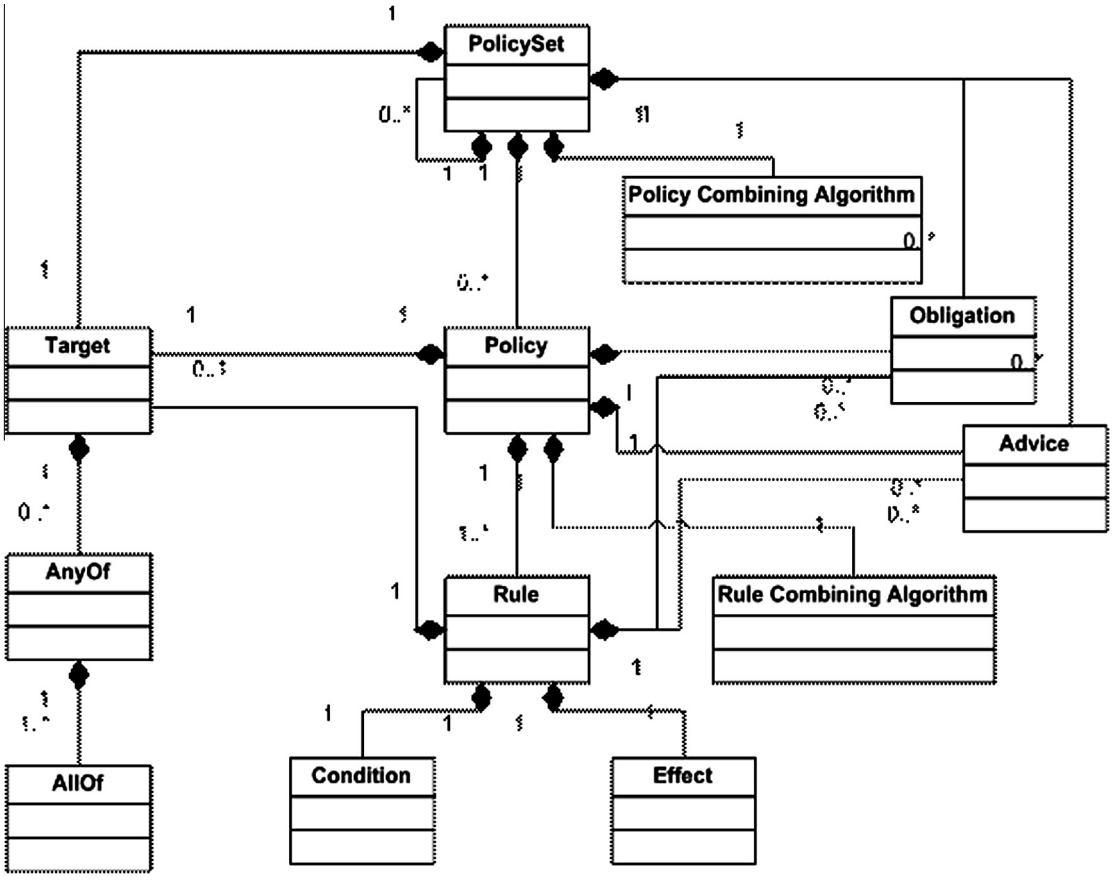

Figure 1 XACML policy authorization elements (conceptual diagram) (A. El-Gamal, 0000).

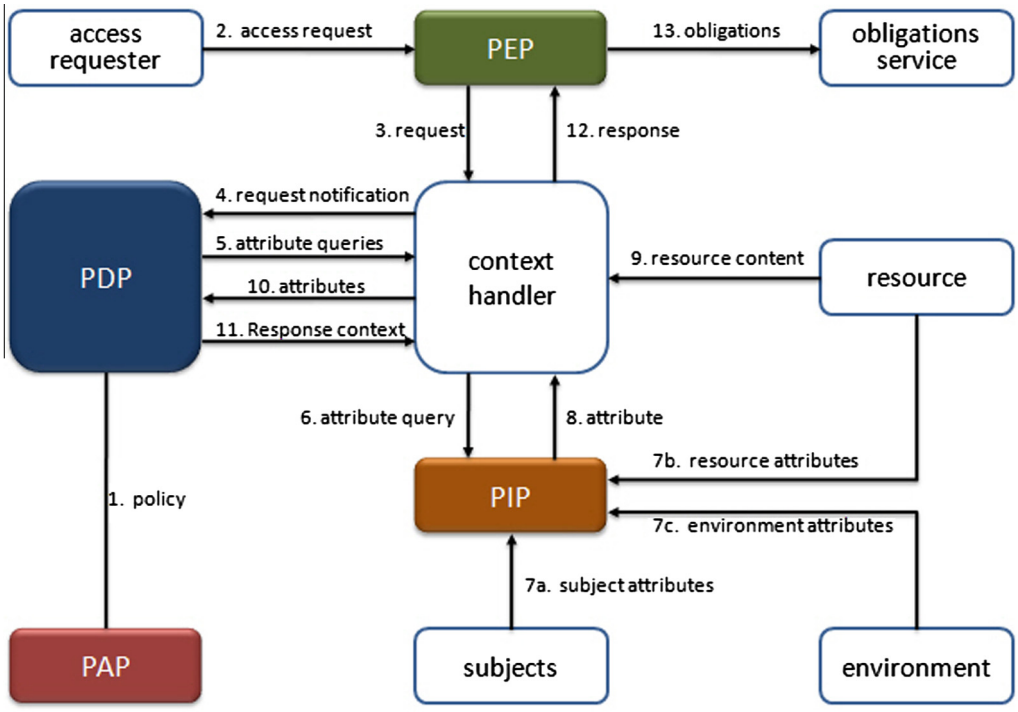

Figure 2 XACML context and data flow diagram (Committee, 2013).

2. Policy Enforcement Point (PEP). This is the interface of the whole XACML to the system or the users. It receives access requests and evaluates them with the help of other components (especially PDP). Decision to permit or deny access to the subject resource is then taken and communicated to the user by PEP.
3. Policy Decision Point. This is the decision engine for access request. Data is collected by PDP from other components. The component includes an analysis system or component to make inference decisions.
4. Policy Information Point (PIP). This represents the memory or the kitchen where all necessary information from other components, resources, or environment are collected and processed.
5. Fig. 2 shows also the steps to make the decision starting from access request step by users or access requesters till making a decision (response) and also decision related obligations. Obligations are related to making some alternative choices. For example, a user with a guest account who is searching for items in an e-commerce system is permitted to search and to temporarily reserve items. However, if the user wants to buy an item, the response will be the denial of access to the resource (e.g. check out resource/service). Obligation can then for example be offered to the user to alleviate their account or switch to another account with the proper user privilege.

4. Related work

XACML (OASIS Committee, 2013), represents the most effective and accepted solution for controlling access in distributed environments (Ardagna et al., 2009). Many products are using and deploying XACML (OASIS Committee, 2011). Several models based on XACML are used to define access rules, for instance Role-Based Access Control (RBAC) is used to enforce policies (Sohr et al., 2008; National Standards Institute Inc., 2004; Sandhu et al., 1996; Lampson, 1971). Another model is known as Attribute Based Access Control (ABAC) where the access control decisions are made based on a set of attributes, see for instance (Yuan and Tong, 2005; Kuhn et al., 2010; Shen and Hong, 2006). Controlling access to security-critical systems used by organizations, such as financial institutes, hospitals, and military organizations without violating the underlying access control policies is a challenging task (Sohr et al., 2008).

XACML is used at large enterprises such as: Bank of America which is also one of the main contributors to XACML (OASIS Committee, 2013). Pardal et al. (2012) showed the utilization of XACML authorization for the traceability of supply chain activities. A tool is developed to utilize XACML information and audit policies in Wegdam (2012) where Attribute Based Access Control (ABAC) policies were derived by the banking need for mobility and the cloud. The presentation in Wegdam (2012) showed a case study of using XACML in a Dutch bank as a pilot study. The case study showed the feasibility of applying such policy management in banking systems. Key advantages sought include: Centralization of taking authorizations from applications in one central point. Attributes of security are important and should be enhanced in XACML to work well for dynamic attributes and not only to static ones.

Islamic banking and finance are emerging more and more as viable alternatives to conventional interest-based banking and financing (Siddiqi, 2006). The rules and regulations of Islamic finance have been extensively studied in the literature see for example Ahmad (1995), Dhumale and Sapcanin (1999), Siddiqi (2006), Hassan et al. (2013), Beck et al. (2013), Errico (1998), Ahmed (2013) and Samad et al. (2005). Despite that, Islamic banking is rarely investigated from the technical perspective especially from information technical design and modeling viewpoints. Islamic banking information systems (IBIS) are information systems that include within their architecture and design rules to agree with Islamic regulations. In most Islamic banks however, these rules are understood and applied by humans and not machines. Accordingly, one of the main challenges facing Islamic banking nowadays is having access to a well-established independent and active Shariah board (ITS, 2011). Currently, active boards provide policies that control the Shariah compliant in general. Performing this task on day-to-day banking operations and transformations is very challenging.

In general, there is a wide spread agreement on the need for current banks in the Islamic world to be able to handle systematically financial transactions that are approved by Shariah regulations (Hassan and Lewis, 2007; Errico, 1998; Ben-Arab and Anas, 2008). Islamic Development Bank (IDB) has been leading an effort recently to develop systems according to such regulations (Islamic Development Bank; Islamic Research and Training Institute). The research proposals focus on investigating traditional banking activities and determine what necessary changes are required in the whole banking system framework to be changed to accommodate these regulations. The approach should also absorb Islamic financial activities' specialties without losing the connection with the general worldwide banking systems.

The research work presented in our paper aims to use XACML to develop banking policies that are compliant with Shariah rules and can be used effectively for day-to-day banking operations.

5. Using XACML to enforce Islamic policies

We claim that this is the first paper to try to tackle the technical issues of Islamic banking with a solution model. In particular, we proposed using a policy management framework (XACML) for: Designing, enforcing and evaluating a financial system where Islamic Shariah laws will be used as part of this financial system. We will demonstrate the proposed system using several examples of Shariah laws and how can they be implemented, enforced and evaluated. Ultimately, the goal is to collect all required information to the components described in XACML class and context diagrams shown earlier. This information can be completed with the assistance of expert domains from both religion and financial sectors. In the following section, we will continue focusing only on the technological design aspects of the proposed system.

5.1. XACML suitability for policy enforcement

We explained earlier how XACML is used for security authorization. In the following steps, we will demonstrate how XACML can be used in another context. This is the context

of the implementation and enforcement of Islamic financial Shariah rules.

In principle, the same architectural components that are used in XACML (i.e. PAP, PDP, PEP, and PDP) can be also used here. This is one of the major advantages sought in this paper to implement a widely used popular system where banks can use such systems for both security and the implementation of Shariah rules as well.

The following example demonstrates how Shariah laws can be enforced through policy management. Table 1 shows examples of authorization rules that can be part of one or more XACML policies. We want in the first example to demonstrate how to implement and evaluate the first rule. This will be shown in this section and with real examples in the next section. Fig. 3 shows an instance of applying Rule 1.

We can alter cases of other users or transactions and show authorization system decisions based on that. We will save space in demonstrating further instances and include details in the experiment section. Although the example is very simple and can be implemented with a small constraint without the need of the whole policy manager, however here are some things to notice:

1. XACML infrastructure is built based on isolating authorization rules from the underlying system, implementation and more importantly from resources.
2. A more complicated authorization rule that needs to check on several criteria can be also applied in the same approach.
3. Authorization rules are abstracted. They can be easily reused, modified, deleted, etc. More rules can be added. These activities can be implemented without the need to change any system's structure or component.
4. As it will be demonstrated later, XACML has many components and implementations using commercial and open source architectures. Policies then can be easily tested before their implementation.

Steps to implement XACML policies can be summarized as follows:

1. Define all the policies in "plain text" in a dataset visible to PAP.

Table 1 Islamic Shariah rules hypothetical examples.

Rule	Description
1	Deny users to withdraw an amount which is more than their balance
2	Prevent Insurance of type: life insurance

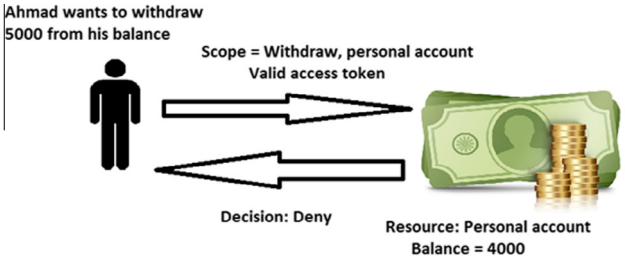

Figure 3 An example of implementing an authorization rule.

2. Create a policy set or relational structure to depict the structure of all policies and relations between them.
3. Separate XACML financial rules' and policies from security policies.
4. Create policy XACML files for all major coarse-grained policies.
5. Create policy XACML files for fine-grained policies. These can be applied only if they do not object with anyone of the major coarse grained policies.
6. In real time cases, convert client's request into a XACML request format.
7. Evaluate the decision and get response in XACML format presented to user/bank employee and obligations.

5.2. Islamic Shariah policies in the context of XACML

In order to demonstrate the Islamic Financial Policy system (IFPS), we will use then 10 legalized Islamic transactions mentioned in the literature of Islamic banking resources. We will show how policies and requests can be designed for each one of them. To reduce the size of the paper and make data better visible, we will use tables rather than XML files. Attributes can then be easily extracted from those tables into XML files. We will show implementation details for 3 policies of the 10 Islamic financial processes mentioned earlier. By convention, we considered those as our top level policies. For each one of the selected policies, we described the policy and two requests, one to permit and the other one to deny the request. Lower level or more granule policies can be then extracted based on those policies. Notice that in this case study we will not consider complex examples such as using hybrid policies to make judgments on requests. For size limitations we showed the tables for the first three policies. Table 2 shows policy for Modharaba Islamic financial process described earlier in Section 3.

As we can see from the requests that represent test cases or actual instances to test policies through, that we can have a large number of cases to either permit or deny. In some cases if rules or constraints contradict with each other, the Policy Decision Point (PDP) may not be able to come up with a clear permit or deny action. In that case, it may produce: Intermediate or Not Applicable status. Such status can then be revised by an auditor to see why constraints contradict each other.

Table 3 shows: Policy, requests and responses for the second policy: Mosharaka or Joint Venture.

Table 3 shows Policy, requests and responses for the second policy. There are some similar aspects with the first policy. It should be mentioned here that those policies are at the very high level in the proposed policy architecture. They hence represent parents or policy sets. Table 4 shows the third policy: Morabaha or cost plus.

We demonstrated through these three high level policies how to formally express Islamic Finance policies. We showed that for a complete system, a large number of policies should be further created from the original 10 policies that highly described Islamic legalized financial transactions.

As an actual policy implementation, we demonstrated several instances of activities related to the utilization of XACML policies using WSO2 application and identity server components (WOS2 - Middleware Solutions). The open source

Table 2 Modharaba or profit sharing: policy.

Description		Bank and customers can share money in an investment and share both profit and loss This policy can be further divided into several policies To CRUD (create, read, update or delete) the Modharaba transaction	
Target	Subject	Bank, customer, manager	
	Resource	Modharaba process (including bank, amount, customer, amount)	
	Action	CRUD	
	Effect	Allow/deny	
Rules [1...n]	Description	This first rule makes general guidelines for one to permit Modharaba process (e.g. a customer with a certain balance, profile, the nature of Modharaba activity, risk level, etc)	
	Effect	Permit (give pre-conditions or constraints)	
	Condition	1. Customer should have a balance above amount A 2. Customer should have an account for no less than B years 3. Modharaba activity should be listed in the bank permitted business activities	
	Target	Bank money reservoir (e.g. how much to authorize to put in the Modharaba process)	
	Request 1: example		**Request 2: example**
Target	Subject	Customer: Hamza	Customer: Yman
	Resource	Hamza balance = A (more than minimum balance); Hamza (time with bank = B), Modharaba business listed in permitted activities = Yes	Yman balance = A1 (less than minimum balance); Yman (time with bank = B), Modharaba business listed in permitted activities = Yes
	Action	Permit	Deny
	Environment	Date of Modharaba is in June 2015	Date of Modharaba is in June 2015
	Request_response_results		Request_response_results
	Permit		**Deny**

Table 3 Mosharaka or joint venture: policy.

Description		Bank can contribute with a customer to invest in establishing a new business or company This policy can be further divided into several policies To CRUD (.e. create, read, update or delete) the Mosharaka transaction	
Target	Subject	Bank, customer, manager, new business (children policies can include one policy for each one of those subjects)	
	Resource	Mosharaka process (including bank, amount, customer, amount)	
	Action	CRUD (create, read, update, or delete): account, transaction, etc	
	Effect	Allow/deny (the permission for the CRUD process)	
Rules [1...n]	Description	This first rule makes general guidelines for one to permit the Mosharaka process (e.g. a customer with a certain balance, profile, the nature of Mosharaka activity, risk level, etc)	
	Effect	Permit (give pre-conditions or constraints)	
	Condition	Customer should have a balance above amount A Customer should have an account for no less than B years Mosharaka activity should be listed in the bank permitted business activities	
	Target	Bank money allocated amount (e.g. how much to authorize to put in the Mosharaka process)	
	Request 1: example		**Request 2: example**
Target	Subject	Customer: Hamza	Customer: Yman
	Resource	Hamza balance = A (more than minimum balance); Hamza (time with bank = B), Mosharaka business listed in permitted activities = Yes	Yman balance = A1 (more than minimum balance); Yman (time with bank = B), Mosharaka business listed in permitted activities = No
	Action	Permit	Deny
	Environment	Date of Mosharaka is in June 2015	Date of Mosharaka is in June 2015
	Request_response_results		Request_response_results
	Permit		**Deny**

system allows building a complete enterprise architecture based on reusable components that can be easily added and integrated as part of an enterprise system. Fig. 4 shows a sample view of our developed policies.

WSO2 is used in implementing and evaluating XACML security components. Policy administration section (PAP) shown in Fig. 4 allows management of policies. Users can create their own policies using one of several methods shown in Fig. 5.

Fig. 6 shows the use of a simple XACML policy editor to create the policy examples described earlier.

Requests and test cases can be then created to test the validity of the policy (Fig. 7). Those requests or test cases represent instances of usage for this policy. Decision in XACML generally can be one of four: Permit, deny, Intermediate and Not-Applicable.

A Policy Decision Point (PDP) is the location where decisions are made. PDPs make their decisions by evaluating a

Table 4 Morabaha or Cost Plus: Policy.

Description		Customer can decide to select part of his/her money to be used in one of the bank investments. Amount of money should exceed the minimum: ABC. The customer will not be allowed to withdraw from the balance allocated for the Morabaha process This policy can be further divided into several policies To CRUD (create, read, update or delete) the Morabaha transaction	
Target	Subject	Bank, customer, manager, Morabaha transaction	
	Resource	Morabaha process or transaction (including bank, amount, customer, amount, nature of investment)	
	Action	CRUD	
	Effect	Allow/deny	
Rules [1...n]	Description	This first rule makes general guidelines for one to permit Morabaha process (e.g. a customer with a certain balance, profile, the nature of Morabaha activity, risk level, etc)	
	Effect	Permit (give pre-conditions or constraints)	
	Condition	Customer should have a balance above amount A customer should have an account for no less than B years Customer cannot withdraw the money allocated for the Morabaha process	
	Target	Customer balance	
	Request 1: example		**Request 2: example**
Target	Subject	Customer: Hamza	Customer: Yman
	Resource	Hamza balance = A (more than minimum balance); Hamza (time with bank = B)	Yman balance = A (more than minimum balance); Yman (time with bank = B)
	Action	Permit	Deny
	Environment	Date of customer withdrawal after Morabaha start date? No	Date of customer withdrawal after Morabaha start date? Yes
	Request_response_results		Request_response_results
	Permit		**Deny**

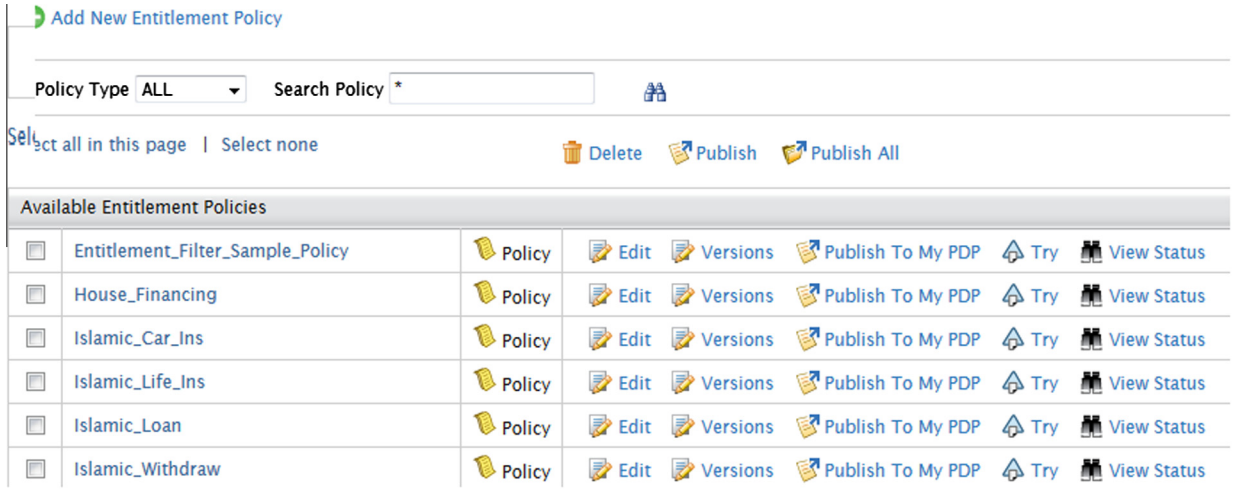

Figure 4 WSO2 application and identity server.

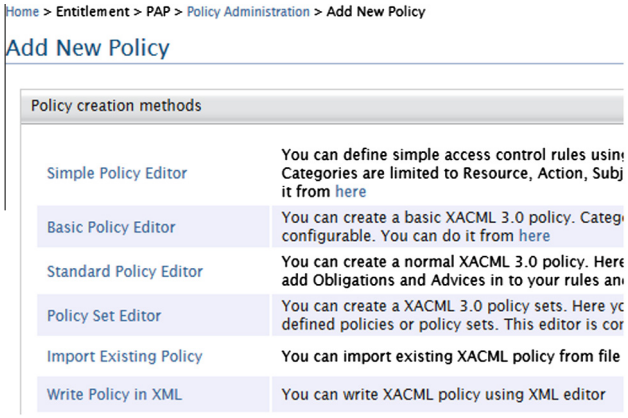

Figure 5 Adding a new policy section.

subject request that represents the issue or the context at hand (e.g. checking whether a certain customer should be authorized a certain amount of money or a transaction) against the relevant policy or policy set.

5.3. Shariah laws policy extractions for Islamic banking

Although our focus here is in Islamic Shariah policy rules, however we noticed that policies related to banks businesses in general and many other fields exist in theory. On the other hand, we could not find any serious technical solutions or implementations for information systems that can: Implement and evaluate those policies automatically for day to day business transactions. This is of course with the exception of security policies.

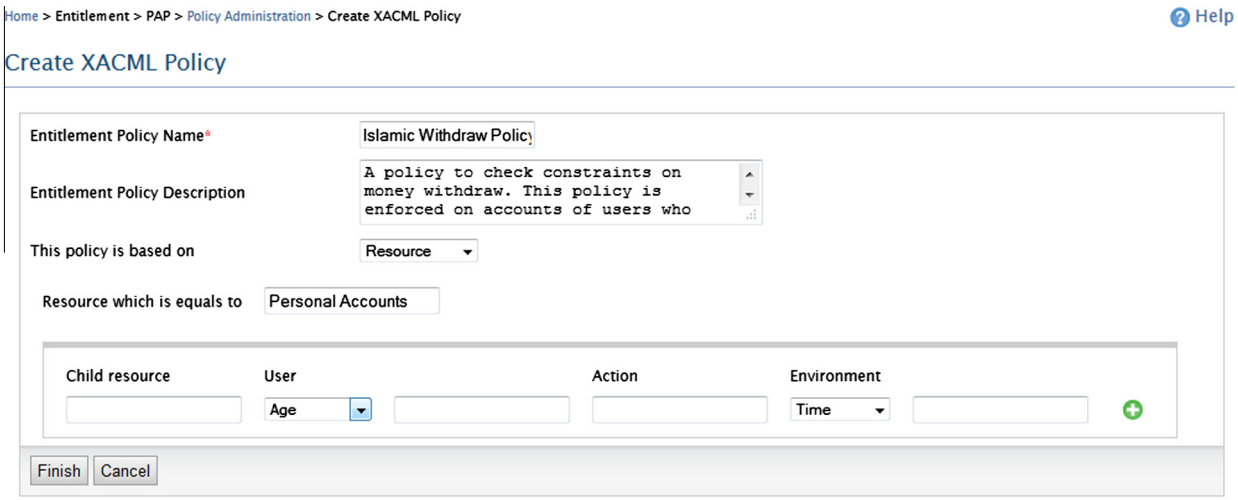

Figure 6 Creating an XACML policy.

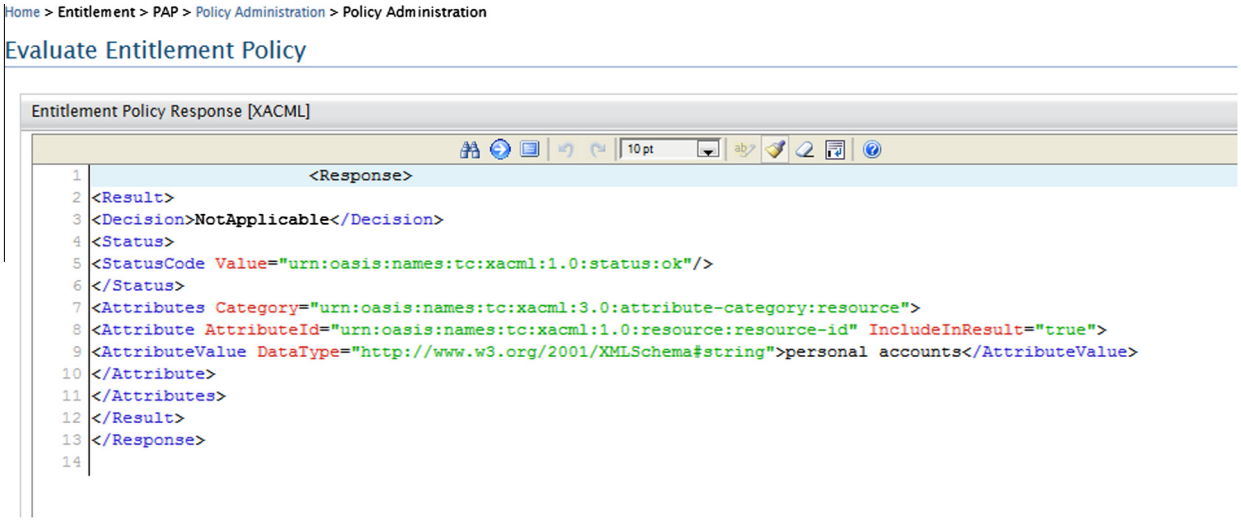

Figure 7 Results of testing a policy.

The problem we are tackling hence in this paper is two fold: The first one is related to how to formalize all Islamic financial rules. Such a problem is largely not technical and related to domain experts or religious experts in the area of Islamic financing. We used in this paper some generic examples as more granule cases need to be judged by domain experts. The main issue here is to propose a formal policy management system through which these rules should be written.

A policy formal language is then required to help extracting Islamic Shariah financial laws in a way that can be easily interrogated by information systems. This is since Software and application systems are incapable of taking as inputs: Quran verses, Hadeeths of Prophet Muhammad (PBUH) or other Shariah sources from their high level sources and be able to come up with judgment or make decisions based on those rules. This is where XACML or security policy approaches are proposed here to fill the gap between high level semantic Shariah laws and information systems. A formal policy language should have a syntax and semantic. Input should be checked for the correct formality. Automatic tools can then deal easily with this formal syntax and use it for further processing including all policy activities described earlier through XACML components: PIP, PEP, PDP, and PAP.

It should be also mentioned here that policies can inherit from each other. We can have a very high level policy on the top of our policy management. Lower or sub-policies can be also included. Lower level policies should elaborate and not violate higher level policies.

A typical banking system can be modeled from several different perspectives. Fig. 8 shows a class diagram representing main entities.

XACML policy defines rules as targets, effects and conditions. Targets include resources, subjects, and actions. A condition is a function that evaluates to: True, False or Indeterminate. Indeterminate indicates that there was not enough information to determine an exact answer.

The resource can be: Data, service or system component (e.g. customer, account, database, etc.).

Decisions are made based on several factors related to the values of the attributes that are pulled from different resources

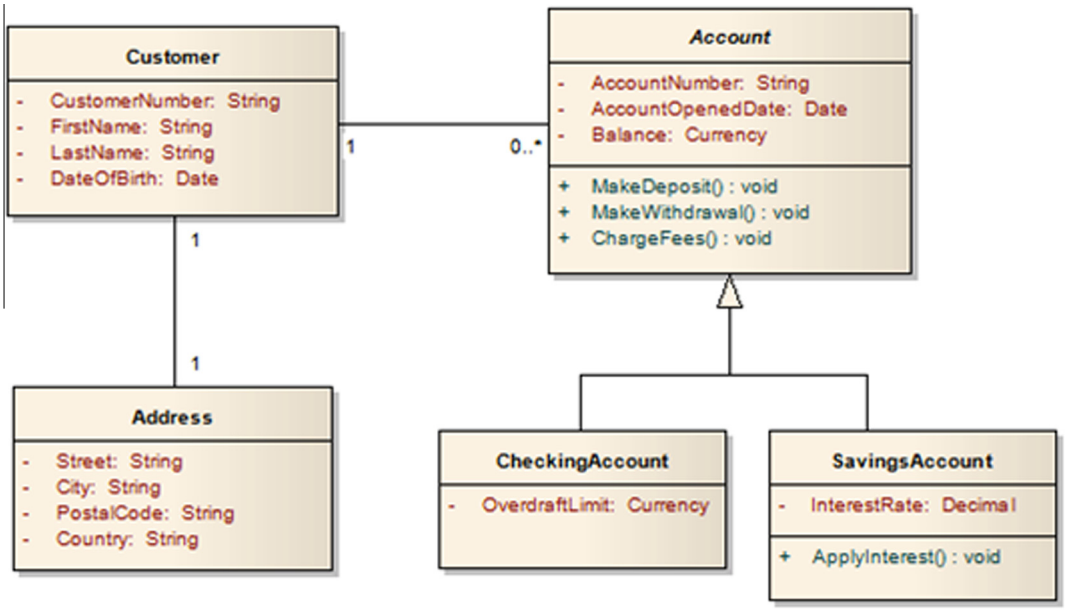

Figure 8 A typical banking system class diagram.

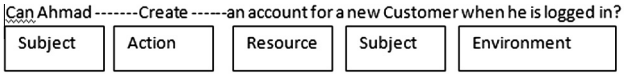

Figure 9 Identifying attributes from a predicate or target.

(e.g. User name, customer type, age, customer number, branch, transfer type or amount, account balance, time, location, etc.). Characteristics or attributes can belong to a subject, resource, action or environment that may be referenced in a predicate or a target, See Fig. 9. The rule can be a target, an effect, a condition and (optionally) a set of obligations or advices. Those rules should be evaluated in the given context. The subject refers to the actor whose attributes may be referenced by a predicate.

An action represents an operation on a resource. PDP is responsible also to specify applicable policies. Those are the policies that should be applicable to the subject context. The decision is returned from the PDP to the PEP.

An effect represents the intended impact of applying the specific or satisfied rule. The environment represents the set of attributes that is relevant to an authorization decision and that is independent of a particular subject, resource or action.

The obligation is then operation that is specified in the rule, policy, or policy set and that should be performed or enforced by the PEP. Fig. 10 below shows a sample content for the

```
Request
!   Subject id = Ahmad!
! Job title = Branch manager!
!   Action!
! Action id = create!
!   Resource!
! Resource type = Customer Account!
! Account# = 1777!
!   Environment!
! Location = branch!
! Authentication type!
! Encryption strength!
```

Figure 11 Pseudo code for a policy request.

banking system in general. Notice that policy in Fig. 10 is a generic one that can be related to security or authorization in general and not to Shariah laws.

Fig. 11 represents an instance or a request for the same example in Fig. 10.

The actual content of XACML policies and requests (e.g. Fig. 12) are written in XML format. This is since XML is widely acknowledged and readable by a large number of tools and applications especially in the web.

6. Conclusion and future work

A large number of banks in the Islamic region try to attract their customers by confirming to the monetary rules in Islamic laws or Shariah. Nonetheless, the major challenge for

```
Policy
!   Subject-id: "ABC_Islamic_Bank Branch Admin"
!   Target!
! Subject: job title=Branch manager
! Resource: customer accounts
! Action: create, read, edit, delete
! Environment: from the branch!
!   Rules!
! Effect: Permit!
!   obligations: result is PERMIT if the branch manager is using
own account and is logged in.
```

Figure 10 A sample content of a banking policy.

```
<Request>!  <Subject>!
 <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
DataType="http://www.w3.org/2001/ XMLSchema#string">!
 <AttributeValue>Ahmad</AttributeValue>!  </Attribute>!
 <Attribute AttributeId="urn:oasis:names:tc:xacml:2.0:subject:role"
DataType="http://www.w3.org/2001/ XMLSchema#string">!
 <AttributeValue>manager</AttributeValue>!
 </Attribute>!  </Subject>!  <Resource>!
 <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/ XMLSchema#string">!
 <AttributeValue>1234</AttributeValue>!  </Attribute>!
 <Attribute AttributeId="http://example.com/resource/type"
DataType="http://www.w3.org/2001/XMLSchema#string">!
 <AttributeValue>printserv</AttributeValue>!
 </Attribute>!  </Resource>!  <Action>!
 <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/ XMLSchema#string">!
 <AttributeValue>create customer account</AttributeValue>!
 </Attribute>!  </Action>!  <Environment/>! </Request>!
```

Figure 12 A request in XML format.

those banks is how to adopt or implement those rules in daily transactions. This is since currently most of these rules are judged case by case and based on opinions from an expert religious board usually assigned by the bank.

We believe hence that this is the first practical trial for a proposal to make such Islamic financial information system documented and structured. Currently, policies that can be implemented and monitored by tools are largely related to security: authentication and authorization issues. However, we believe that this same process can be extended to policies from the different domains such as those policies that can be extracted from Shariah laws.

In this paper, we showed how XACML security authorization system and policy management can be used to implement and enforce financial policies related to Islamic Shariah guidelines. XACML is typically used for security policies. However, due to the resemblance of the two cases, and due to the wide use and applicability of XACML, we believe that such a flexible and out of the box solution can help banks and other financial related entities such as insurance companies in the Middle East and Islamic countries to comply with Islamic Shariah rules and provide their willing customers with such services specially for daily transactions or activities.

We defined 10 financial processes acknowledged by Islamic banking systems. We used these as the starting point to create the policy management system for Islamic finance. An automated system can then be developed that can formally interact with those policies with the least human involvement. The ability to systematize and automate the Islamic financial policies was the main goal and contribution of this paper. We showed the feasibility of the approach. Future extensions of this paper should include the layout of the complete policy management system architecture along with a large set of policy requests or test cases.

References

Abdullah, A., Sidek, R., Adnan, A., 2012. Perception of non-muslims customers towards Islamic banks in Malaysia. Int. J. Bus. Soc. Sci. 3, 151–163.

Ahmad, Z., 1995. Islamic banking: state of the art. IDB Prize Lecture. Islamic Research and Training Institute, Islamic Development Bank, Jeddah.

Ahmed, I., 2013. Islamic banking growth role of technology. In Business Administration, University of Atlanta, p. 72.

Akhtar, R., Talreja, B., 2012. Why India needs Islamic banking. Infosys-Finacle. www.infosys.com/finacle.

AlBaraka Bank Group. Principles of Islamic Banking. <http://www.albaraka.com/default.asp?action=article&id=46>. Last visit: 23/9/2014.

Al-Omar, F., Iqbal, M., 1999. Challenges facing Islamic banking in the 21st century. In: Second Harvard University Forum on Islamic Finance: Islamic Finance into the 21st Century, Cambridge, Massachusetts, Center for Middle Eastern Studies, Harvard University, pp. 243–253.

American National Standards Institute Inc., 2004. Role Based Access Control. ANSI-INCITS 359-2004.

Ardagna, C.A., Vimercati, S., Paraboschi, S., Pedrini, E., Samarati, P., 2009. An XACML-based privacy-centered access control system. In: 1st ACM Workshop on Information Security Governance. ACM, Chicago, Illinois, USA.

Beck, T., Demirgüç-Kunt, A., Merrouche, O., 2013. Islamic vs. conventional banking business model, efficiency and stability. J. Banking Finance 37, 433–447.

Ben-Arab, M., Anas, E., 2008. Managing risks and liquidity in an interest free banking framework: the case of the Islamic banks. Int. J. Bus. Manage. 3.

Dhumale, R., Sapcanin, A., 1999. Islamic banking—promoting equity with a range of tools. A study by the Regional Bureau for Arab States, UNDP, in cooperation with the Middle East and North Africa Region, World Bank.

El-Gamal, M.A., 2000. A basic guide to contemporary Islamic banking and finance. Rice University, Houston, TX.

Errico, L., 1998. Islamic banking: Issues in prudential regulations and supervision (IMF working paper). International Monetary Fund, Monetary and Exchange Affairs Dept.

Hassan, K., Lewis, M., 2007. Handbook of Islamic Banking. Edward Elgar Publishing Limited.

Hassan, M., Kayed, R., Oseni, U., 2013. Introduction to Islamic Banking & Finance Principles and Practice. Pearson Education Limited, Essex, UK.

Imran, M., AbdulSamad, S., Masood, R., 2011. Awareness level of Islamic banking in Pakistan two largest cities. J. Manag. Sci. 5, 1–20.

Institute of Islamic Banking and Insurance. <http://www.islamic-banking.com>. Last visit: 23/9/2014.
Islamic Development Bank. <http://www.isdb.org/>. Last accessed 23/9/2014.
Islamic Research and Training Institute. http://www.irti.org/, last visit: 10/8/2014.
ITS, 2011. Islamic Banking and Technology Challenges. Gartner, Issue 2. <http://www.its.ws/upload/ITS_issue2_1542.pdf>.
Karbhari, Y., Naser, K., Shahin, Z., 2004. Problems and challenges facing the Islamic banking system in the west: The case of the UK. Thunderbird Int. Bus. Rev. 46, 521–543. http://dx.doi.org/10.1002/tie.20023.
Kuhn, R., Coyne, E., Weil, T., 2010. Adding attributes to role-based access control. IEEE Comput. 43, 79–81.
Lampson, B.W., 1971. Protection. In: Proceedings of 5th Princeton Conference on Information Sciences and Systems. Princeton, pp. 437–443.
Liu, A., Chen, F., Hwang, J., Xie, T., 2011. Designing fast and scalable XACML policy evaluation engines. IEEE Trans. Comput. 60, 1802–1818.
McLean, D., 2013. World Islamic banking competitiveness report 2012–2013. In: 19th Annual World Islamic Banking Conference (WIBC 2012), Bahrain.
New Horizon Magazine. Institute of Islamic Banking and Insurance. <http://www.islamic-banking.com/IIBI_magazine.aspx>. Last visit: 21/9/2014.
OASIS Committee, 2011. XACML references and products, Version 1.85. <https://www.oasis-open.org/committees/download.php/42588/xacmlRefs-V1-85.html#Products>.
OASIS Committee, 2013. Organization for the advancement of structured information standards. eXtensible Access Control Markup Language (XACML) Version 3.0, OASIS Standard. <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html>.
Pardal, M., Harrison, M., Sarma, S., Marques, J., 2012. Performance assessment of XACML authorizations for supply chain traceability web services. In: 8th International Conference on Next Generation Web Services Practices.
Samad, A., Gardner, N., Cook, B., 2005. Islamic banking and finance in theory and practice: the experience of Malaysia and Bahrain. Am. J. Islamic Soc. Sci. 22, 69–86.
Sandhu, R., Coyne, E., Feinstein, H., Youman, C., 1996. Role based access control models. IEEE Comput. 29, 38–47.
Shen, H.-B., Hong, F., 2006. An attribute-based access control model for web services. In: Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'06). IEEE.
Siddiqi, M., 2006. Islamic banking and finance in theory and practice: a survey of state of the art. Islamic Econ. Stud. 13, 1–48.
Sohr, K., Mustafa, T., Bao, X., Ahn, G.-J., 2008. Enforcing role-based access control policies in web services with UML and OCL. In: Annual IEEE Computer Security Applications Conference (ACSAC). IEEE, Anaheim, California, USA.
The Islamic Finance Rules (IFR), 2013. Rule-making Instrument (No. 125), Dubai Financial Services Authority (DFSA), Dubai, UAE IFR/VER7/07-13.
Wegdam, M., 2012. Using XACML to implement context enhanced authorizations. XACML pilot at a large Dutch bank, XACML seminar.
WOS2 – Middleware Solutions. <http://wso2.com>, Last visit: 10/5/2014.
Yuan, E., Tong, J., 2005. Attributed based access control (ABAC) for web services. In: IEEE International Conference on Web Services (ICWS 2005). IEEE.