# Trustworthy emergency information brokerage service (TIBS)

J. K. Zao[1], K. T. Nguyen[1,3], Y. H. Wang[1], A. C. H. Lin[2],
B. W. Wang[2] & J. W. S. Liu[2]
[1]*Department of Computer Science,*
*National Chiao Tung University, Taiwan*
[2]*Institute of Information Science, Academia Sinica, Taiwan*
[3]*Department of Information Technology and Communication,*
*Hong Duc University, Vietnam*

## Abstract

Availability and timeliness of relevant information is of paramount importance during disaster response. Structural and situational information should be made available lavishly to decision makers, field workers, victims, even the general public provided that it would be possible to track down potential abuses. The trustworthy emergency information brokerage service (TIBS) described in this paper was developed based on such a conviction. This pervasive information flow control system offers information desensitization, flow traceability and use accountability services in two separate phases of disaster management: (1) in the preparatory phase, a prospective P-TIBS subsystem will provide information filtering and fusion tools that can help resource owners to desensitize organizational/structural information and store them in a virtual repository deployed pervasively on many points of service (POS); (2) during the disaster responsive phase, a retrospective R-TIBS subsystem will lease the desensitized information and offer information flow traceability as well as use accountability services according to pre-specified information release and accountability policies. This accountability approach will be more scalable and responsive than the "break-the-glass" access control overriding mechanism available in many hospital information systems as it alleviates the need to authenticate individual users and authorize their emergency information access. This paper provides an overview of the TIBS system architecture and the enabling technology it employs.

*Keywords: disaster response, information accountability, authorization override.*



WIT Transactions on The Built Environment, Vol 133, © 2013 WIT Press
www.witpress.com, ISSN 1743-3509 (on-line)
doi:10.2495/DMAN130221

## 1 Introduction

Recent epics of disaster reminded us that when a disaster strikes, accurate structural information and timely situational information can be vital for carrying out efficient evacuation/rescue operations and performing effective damage control provided that such information can be made widely available to decision makers, field workers, victims, even the general public while the physical and social infrastructures are in turmoil. Two major obstacles currently hinder effective information dissemination during disasters. First, the owners of vital information often set up security and privacy protection to prevent illegal access. For examples, owners of private buildings often safeguard their structural details; mobile service providers refuse to disclose the locations and use patterns of their customers; even government agencies may conceal the extent of a flood due to their concerns about public perception. Then, ironically, in this "information age", we still lack a pervasive robust information/communication infrastructure that can withstand disasters and offer highly available services. This paper tries to address the first issue while leaving the second issue to a companion paper on Open Information Gateways [1].

The obstacle introduced by security and privacy protection may be circumvented in part by adding the emergent "break-the-glass" (BTG) extensions to standard role based access control model [2, 3]. Using these extensions, one can specify a hierarchy of emergency access control policies based on the security overriding requirements at different levels of emergency. Usually, BTG polices for more severe emergency cases allow more flexibility in overriding nominal access control decisions while imposing more extensive auditing requirements. These BTG extensions are quite capable of handling the emergency access control cases arise among electronic health record systems, which usually require permissions to be granted to healthcare professionals so that they can access unconscious patients' medical records without their consent [4]. In those cases, the group of subjects, e.g. healthcare professionals in a hospital, is usually small and can be authenticated by a functioning hospital information system. When a disaster strikes, however, thousands of victims and ad-hoc rescue workers may need to find their ways through collapsed buildings and flooded streets. With servers down and cell phones barely connected to one another via Bluetooth or Wi-Fi, it would be impossible to authenticate any of the subjects; yet, accurate and timely information must be given to them in these life-or-death situations. Furthermore, useful information may have to be gathered from different organizations residing within separate security or network domains. Existing BTG access control override simply do not work in these cases.

This paper presents a pervasive information flow control system, named the Trustworthy Information Brokerage Service (or TIBS for short), as a more scalable and responsive alternative. Unlike conventional access control system, TIBS offers *information desensitization*, *flow traceability* and *use accountability* services in two separate phases of disaster management. First, in the pre-paredness phase, a prospective or P-TIBS subsystem will provide information filtering/fusion services to help resource owners to desensitize organizational or

structural information and store them in a *virtual repository* distributed on points of service (POS) that are deployed pervasively for the sake of availability and responsiveness. Then, in the response phase, a retrospective or R-TIBS subsystem will offer information flow traceability and use accountability services based on pre-specified *information release* and *accountability policy clauses* while largely eliminate user authentication requirements. Among them, the information release clauses specify the *conditions* under which information may be released to the public while the use accountability clauses state the *obligations* such as providing mobile phone numbers and generating attestation meta-data tags that the recipients must fulfil. Just as the BTG conditions can be derived from information requirements in emergency cases, so are the release and accountability conditions of each piece of information correspond to different levels of emergencies.

The rest of this paper provides a stream-lined presentation of TIBS. An overview of its structural and functional architectures is included in Section 2. The technologies developed for or employed in the P-TIBS and R-TIBS subsystems are described in Sections 3 and 4 respectively. As a conclusion, Section 5 summarizes the completed tasks and outlines a future plan.

## 2 System overview

First of all, TIBS does not work alone. It functions as the frontend as well as the backend of an emergency information management framework, known as the Virtual Repository (VR) [5]. By itself, VR is a distributed survivable Linked Data framework that provides services and tools for publishing information relevant to disaster mitigation and response. It adopts Resource Description Framework (RDF) as its data model, Universal Resource Identifiers (URI) as its resource identifiers and semantic web ontology as its vocabularies [6].

On the next page, Figure 1 shows the functional architecture of VR and its relationship with the applications and the information sources it serves. Following are the components crucial to the interoperation between TIBS and VR.

- Information Virtualization Layer or VR Tools: they are assorted information filtering, access and translation tools that convert gathered information from multiple sources to suite different applications and services.
- Intelligent Active Storage Service (IASS) or VR Core: this is a distributed middleware layer that performs event-triggered, push-based information delivery according to Event-Condition-Action and QoS (ECA+Q) rules.
- Points of Service (POS) Servers: these are lightweight servers deployed in various public places equipped with computing, data storage and communication resources. The best examples are the public Wi-Fi stations installed in many café or convenience stores.
- Mobile Assistants for Disasters (MAD) [7] or similar VR client applications: these applications may search and retrieve disaster mitigation and response information under the control of TIBS information flow traceability and use accountability services.

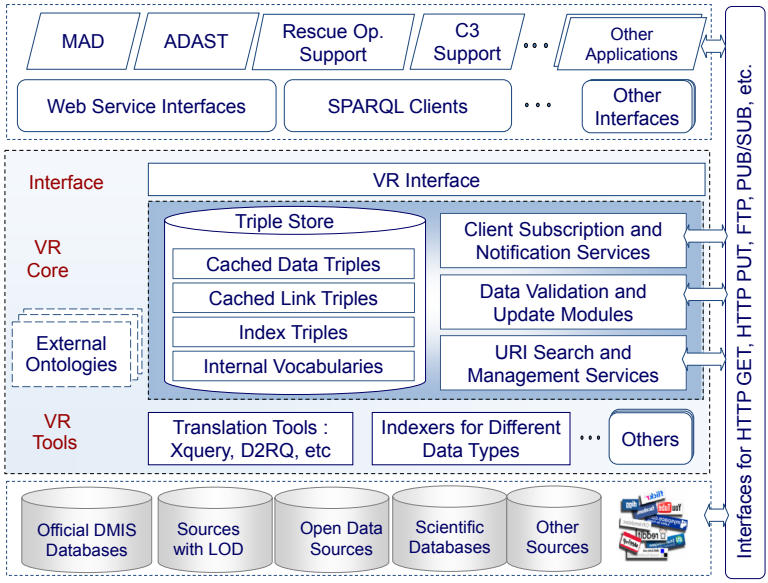


Figure 1: Functional architecture of virtual repository (VR).

As mentioned, TIBS can be divided into a prospective P-TIBS subsystem that helps resource owners to desensitize information for public release and a retrospective R-TIBS subsystem that offer information flow traceability and use accountability services based on information release and accountability policies. Their interactions with resource owners and VR are illustrated in Figure 2.

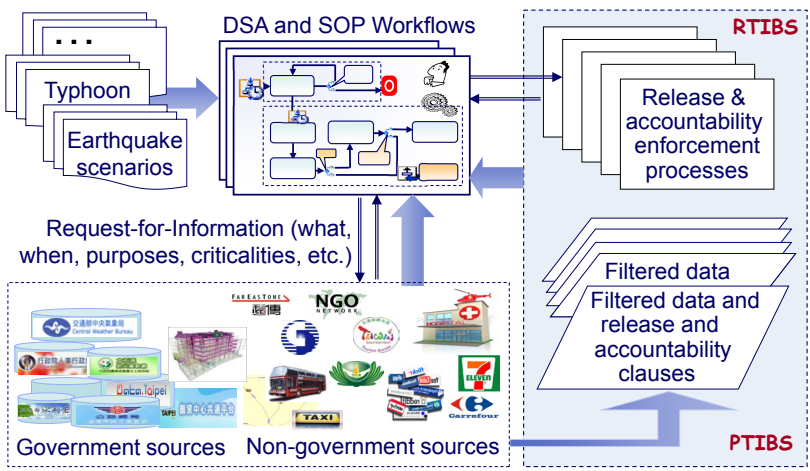


Figure 2: Interactions between TIBS, resource owners and Virtual Repository.

Specifically, P-TIBS serves as the backend gatekeeper for VR. It enforces multi-domain attribute/role-based access control between various resource owners and the VR-IASS core. In addition, P-TIBS enables resource owners to use the filtering and fusion tools available in the VR Information Visualization Layer to desensitize the information to be released and place them into the VR Triple Store. Since these transactions will be carried out in normal circumstances during the disaster preparedness phase, they resemble the data transfers among private hospital information systems and a government-run electronic health record repository. In both cases, a multi-domain access control system must be in place to manage the transactions according to the access control policies established with each resource owner. Section 3 provides an overview of the technologies employed by P-TIBS.

R-TIBS, on the other hand, serves as the frontend gatekeeper for VR. It implements a distributed infrastructure among VR POS servers and its client application similar to Weitzner's Information Accountability Framework [8]. Following are the essential components.

- Accountability Appliances: these are trustworthy devices and programs that mediate information accesses, maintain data transfer logs and provenance records. POS servers and MAD are typical examples.
- Provenance Controllers: these are trusted servers that enforce information release/accountability policies and also operate as the trusted third party for the user intention and usage restriction handshake [9, 10] in the HTTPa protocol, which we employed to conduct information assurance with accountability assurance. POS servers play this crucial role.
- Distributed Transaction Logs: Accountability Appliances must all take up the responsibility of recording all information transfer and use instances so that information usage can be monitored and users can be held accountable in case they abuse the information. POS and MAD will all keep transaction logs in Accountability-in-RDF (AIR) format [11] and upload these logs opportunistically onto the VR Triple Store.

## 3 Prospective TIB technology

According to XACML architecture [12], a role/attribute-based access control (A/RBAC) system consists of Policy Administration Points (PAP) that manage the access control policies, Policy Decision Points (PDP) that make the authorization decisions, Policy Enforcement Points (PEP) that carry out the decisions and Policy Information Point (PIP) that supply attribute values of subjects, resources, actions and action environments. In multi-domain systems, these components may be installed in distributed servers and make access control decisions based on policies and role/attribute assignments associated with individual domain. For P-TIBS, we employed PERMIS, a mature open-source authorization engine as its PDP. We also developed a generic object-based A/RBAC policy specification and proposed two extensions to the OAuth 2.0 protocol standard. All these are discussed in the following paragraphs.

### 3.1 PERMIS Access control authorization engine

PERMIS (PrivilEge and Role Management Infrastructure Standards) [13, 14] is an access control authorization system that provides the necessary facilities for users to manage authorization policies and for applications to make authorization decisions. PERMIS supports distributed assignments of roles and attributes to users via multiple attribute authorities and uses X.509 attribute certificates to maintain these attributes. PERMIS can be integrated with virtually any application and any authentication mechanisms including usernames/passwords, Kerberos, Shibboleth, public key infrastructures and OpenID. In addition to mundane policy and credential management, PERMIS provides two essential services: (1) the *credential validation service* that validates users' roles according to *user-role assignment rules* and (2) the *authorization decision engine*, which embodies a Policy Decision Point (PDP), evaluates users' access requests according to *role-permission assignment rules*.

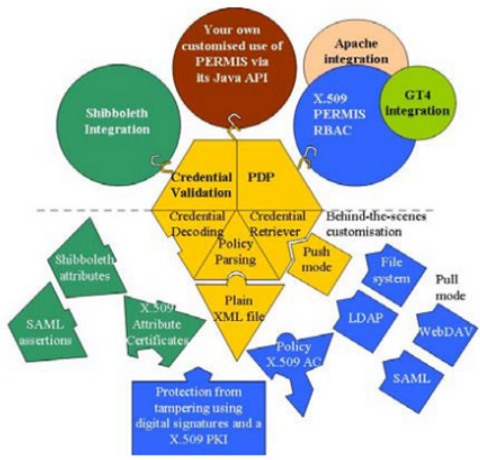


Figure 3: Functional architecture of PERMIS [13].

### 3.2 Generic object based A/RBAC policy specification

Mimicking the generic object-oriented programming paradigm, we devised a generic A/RBAC policy specification as a modular, expressive and reusable way to specify attribute/role-based access control policies in multi-domain environments. Beside of developing a *polymorphic typing scheme* to articulate the role hierarchies, we also implemented a *policy translator* and a *static policy checker*, which can convert Java-like generic policy specification into standard XACML policies. The entire system can run in the Eclipse integrated programming environment and used as a policy pre-processor of PERMIS authorization decision engine. This schema was devised originally for specifying access control policies of electronic health records. It was adopted into P-TIBS to

express multi-domain access control policies that are enforced among different information owners and providers.

The generic A/RBAC policy specification was built upon three polymorphic typing principles: First, for the sake of mutual independence and symmetry, Object Roles were added as a first-class component alongside with Subject Roles and Actions. Figure 1 shows the symmetric bindings exist among all the first-class components. Note that there are two kinds of entity-role-action bindings: the Subject Sessions that assign specific Subjects to each Subject Role as well as the Object Sessions that bind different Objects to the Object Roles. These sessions can be established and dismantled asynchronously and hence greatly enhance the dynamics of subject/object permission bindings while maintaining the static nature of the A/RBAC policies.

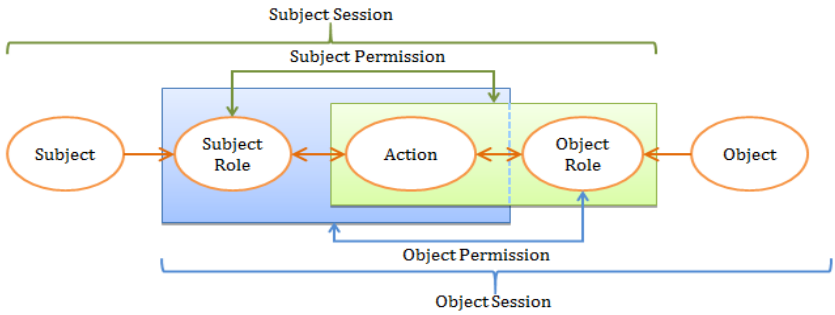


Figure 4: Components of generic A-RBAC policy specification.

Second, *type parameters* were introduced to cast Subject and Object Roles as *bounded polymorphic types*. In the following example, `Dept`, `Rank` and `Status` are all type parameters of the polymorphic type, **`Nurse`**.

```
Nurse <Dept, Rank, Status>
Patient <Dept>
Dept   ∈   {Oncology, Cardiology, Urology, GI, … }
Rank   ∈   {Graduate, Registered, Practitioner}
Status ∈   {Observer, OnDuty, InCharge}
```

The polymorphic nature of Subject and Object Roles enables policy administrators to articulate role inheritance hierarchies in terms of subtype relations. Most importantly, this polymorphic typing system allows the subtype relations among Roles to be specified in terms of parallel subtype relations existing among their type parameters. For example, the following policy, which is applicable to *any* nurse on duty:

```
Nurse <ANY, ANY, OnDuty>}{
permission(GUIDE, Patient<ANY:Dept>);
}
```

It will be inherited by a registered nurse of a specific department *d*,

```
Nurse <d, Registered, OnDuty>{
  permission(INJECT, Patient<d>);
  permission(GUIDE, Patient<ANY>);
}
```

These *type inference rules* not only greatly reduce the number of explicit role specifications — which has always been a major issue in RBAC policy model — they also simply the expressions of policy specification.

Finally, static type checking was employed to verify the consistency among policy specifications. To aid this process, the ranges of action binding with both Subject and Object Roles were specified explicitly as follows:

```
action endoscopy (Doctor<GI, Proctologist, OnDuty>,
Patient<GI, colon-cancer, ANY>)
```

These *action range specifications* were prescribed as an integral part of the policy template for an application. They enable us to build the *policy translator* and the *static checker* for generic A/RBAC policies.

### 3.3 OAuth subject role and target scope extensions

In multi-domain access control, information providers do not have the right to know the identity of information users. However, the providers should have the prerogative to limit the scope of information accessible to the users. To enforce these principles of least privilege, we also proposed two extensions to the standard OAuth 2.0 authorization protocol [15]. These extensions have been added to the open OAuth reference implementation and will be submitted as an Internet Draft to the IETF OAuth working group.

## 4 Retrospective TIBS technology

Enforcing information accountability during disaster response is both a massive and a relatively straightforward task. It is *massive* because the operation often involves unforeseeable number of people devouring large amount of highly dynamic situational information. It is nonetheless *relatively straightforward* because the disclosed information should be consumed within a limited space and time span. Users who want to obtain similar information outside of the emergency situations should do so via regular venues. Consequently, among the four components of Weitzner's Information Accountability Framework [7], *policy specification* and *policy reasoning tools* shall be rather straightforward to implement: the policy specification should prescribe the space/time limits for legitimate use of disseminated information while the policy reasoning tools should prevent the spread of information beyond those limits. The challenges lie with the detection of off-limit information use and the backtracking of information flow by means of *transaction logs* in order to hold the culprits accountable for their acts. Following are the brief discussions of these issues. Although most of the development work is still on the drawing board, we are trying to employ existing technologies in our design.

### 4.1 Information flow and transaction logs

As Sloan and Warner pointed out in [16], in order to assure accountable information usage, one may require to log "every transaction everyone makes everywhere". These logs must be kept by every endpoint in the distributed computing

framework. To make this task a bit more manageable for TIBS, we intend to log *every information transfer* across the boundaries of *every information domain*. Each domain is defined as a group of users and their computing devices that possess a piece of information. For example, if the layout of a damaged building was given to a rescue team, sharing that layout among the computing equipment owned by the team members is regarded as *intra-domain transfers* and not to be logged; however, sharing it with a member of another team will be regarded as an *inter-domain transfer* and hence must be logged. Since information may be shared through peer-to-peer communication, *every* VR client application as well as *every* POS server must play the role of an Accountable Appliance and perform the transaction logging function faithfully.

Each transaction log must record the source and destination, the date and time as well as the provenance record of the information transfer. Since strong authentication may not be enforced in emergency situations, information source and destination may be tentatively tied to users' mobile phone numbers or personal identities. Once the Host Identity Protocol (HIP) [17] is widely deployed, host identities with implicit certification capability will be used as the official source and destination identities.

In order to maintain syntactic compatibility with the other data kept in the VR Triple Store, we adopted the Accountability in RDF (AIR) policy language [18] developed by Weitzner's Decentralized Information Group (DIG) at MIT. AIR uses the Terse RDF Triple (Turtle) language [19] and the Notation 3 logic framework [20] to support AMORD-like deductive reasoning on dependency. The transaction logs in AIR can be used to support dependency tracking and help to trace an information abuse back to the plausible culprits.

Finally, R-TIBS must enable POS servers and VR client applications to conduct information exchanges freely with accountability assurance support during emergency situations. For that purpose, we employed the HTTPa protocol [9] also developed by MIT DIG to provide blanket protection to these transactions because all web-based client-server exchanges are conducted via HTTP. The crucial role of the Provenance Controllers of released information shall be played by the POS servers that constitute the VR storage core.

### 4.2 Accountability policy specification and reasoning tools

R-TIBS policy specification consists of two parts: the *information release clauses*, which specify the conditions under which information may be released to the public and the *information use accountability clauses*, which state the obligations that the recipients of information must fulfil.

The information release clauses extend the event-condition-action (ECA) rules used in the active database systems with an addition of quality-of-service (QoS) requirements. Together, they prescribe the events that trigger the release of information stored in the POS servers, the transformations that convert the information into releasable and useful forms and the QoS requirements that the data dissemination processes are expected to satisfy. The actions of information release and transformation will be carried out by the VR Interfaces while the

QoS requirements will be passed to the Open Information Gateways and enforced by the overlay network.

The information use accountability clauses state the space and time spans as the conditions under which the information dissemination processes are allowed to proceed and the specifics of the transaction logs as the obligations that the dissemination processes are expected to fulfil. Both the release and the accountability clauses will be expressed as XACML v.3 policies and parsed by the Policy Enforcement Points (PEP) embedded in the Accountable Appliances.

## 5 Summary and future work

The entire OpenISDM project has just finished its first year. In the past year, the TIBS subproject has been focused on the development of P-TIBS subsystem, especially the construction of a distributed multi-domain A/RBAC authorization platform to aid the release of organizational and structural information crucial to disaster preparation, mitigation and response. A PERMIS based proof-of-concept prototype (Figure 5) has been built and tested. In addition, a development plan for R-TIBS subsystem has been laid out and the key technologies chosen. In this and the next year, an information accountability infrastructure for disaster response will gradually be put in place. Innovative techniques will be spawned off naturally during this process.

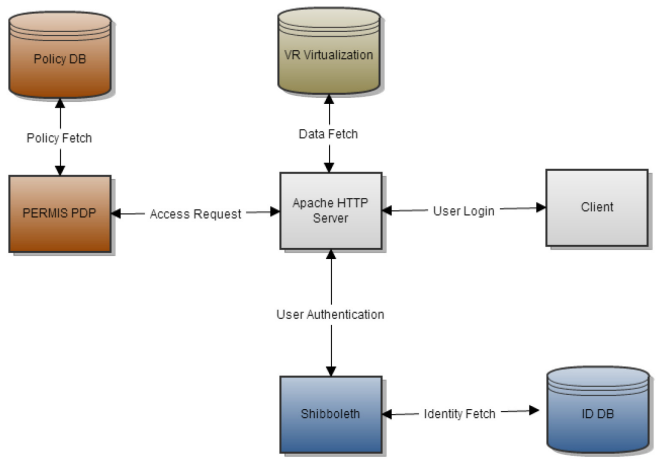


Figure 5: Functional modules of P-TIBS proof-of-concept prototype.

## Acknowledgement

This work is supported by the Academia Sinica OpenISDM (Open Information System for Disaster Management) project.

## References


[1] C.S. Shih, *et al*., "Distributed Service Recovery for Open Information Gateways", Disaster Management 2013, July 9–11, A Coruña, Spain.
[2] A.D. Brucker and H. Petritsch, "Extending Access Control Models with Break-Glass," SACMAT'09, June 2009.
[3] J. Alqatawna, *et al*, "Overriding of access control in XACML," POLICY'07, 2007.
[4] M. Davis, "Health care requirement for emergency access", Department of Veteran Affairs, January 2009.
[5] Y.Z. Ou, *et al*., "A Linked-Data Based Virtual Repository for Disaster Management Tools and Applications," Disaster Management 2013, July 9–11, A Coruna, Spain.
[6] T. Berners-Lee, "Design issues: Linked Data", http://www.w3.org /DesignIssues/LinkedData.html.
[7] Y.A. Lai, *et al*., "Virtual Disaster Management Information Repository and Applications Based on Linked Open Data". RITMAN Workshop, co-located with SOCA, December 2012.
[8] D.J. Weitzner, *et al*. "Information accountability." Communications of ACM 51.6 (2008): 82-87.
[9] O. Seneviratne, and L. Kagal. "HTTPa: Accountable HTTP." IAB/w3C Internet Privacy Workshop. 2010.
[10] O. Seneviratne, and L. Kagal. "Usage Restriction Management for Accountable Data Transfer on the Web."
[11] L. Kagal, C. Hanson, and D. Weitzner. "Using Dependency Tracking to Provide Explanations for Policy Management." POLICY 2008.
[12] Extensible Access Control Markup Language (XACML). http://xml.coverpages.org/xacml.html#URLs.
[13] D. Chadwick, *et al*. "PERMIS: a Modular Authorization Infrastructure." Concurrency and Computation: Practice and Experience 20.11 (2008): 1341-1357.
[14] PrivilEge and Role Management Infrastructure Standards (PERMIS). http://sec.cs.kent.ac.uk/permis/.
[15] OAuth 2.0. http://oauth.net/2/.
[16] R.H. Sloan and R. Warner, "Developing Foundations for Accountability Systems: Informational Norms and Context-Sensitive Judgments", Proc. 2010 ACM Workshop on Governance of Technology, Information, and Policies.
[17] P. Nikander, A. Gurtov, and T.R. Henderson. "Host Identity Protocol (HIP): Connectivity, Mobility, Multi-Homing, Security, and Privacy over IPv4 and IPv6 Networks." Communications Surveys & Tutorials, IEEE 12.2:186-204, 2010.
[18] AIR Policy Language. http://dig.csail.mit.edu/TAMI/2008/12/AIR/.
[19] Terse RDF Triple Language. http://www.w3.org/TeamSubmission/turtle/.
[20] Notation 3 Logic. http://www.w3.org/DesignIssues/Notation3.