

Towards Cross-domain Security Properties supported by Ontologies

York Sure¹ and Jochen Haller²

¹Institute AIFB, University of Karlsruhe
76128 Karlsruhe, Germany
sure@aifb.uni-karlsruhe.de

²SAP Research
Vincenz-Priessnitz-Str. 1, 76131 Karlsruhe, Germany
jochen.haller@sap.com

Abstract. Security is considered as a major driver for the success of E-Business, especially in a business-to-business environment. Current research activities in this area are conducted in European Union funded research projects, such as TrustCoM putting an emphasis on the collaborative aspects of business processes across administrative and trust domains. With respect to the tendency of business partners to set up their own security islands, e.g. based on isolated Public Key Infrastructures (PKIs), this development introduces a contradiction for collaborative business process. Clearly expressed process related security requirements across domains can not be met by domain specific security infrastructures. This contribution explores the possibility to bridge the identified gap using semantic relationships contributed by ontologies.

1 Introduction

1.1 Motivation

Current research in European Union funded research projects is focussing on electronic business (e-business) in a business-to-business (B2B) setting. B2B is perceived in terms of collaborating organizations, so called virtual organizations (VOs) being dynamically-evolving virtual business entities. In a virtual organization different member entities, such as companies, individuals or government departments, form a construct where the participating entities pool resources, information and knowledge in order to achieve common objectives. These objectives cannot be met by one or a subset of members alone. It is envisioned that such VOs are self-managed spanning different administrative domains enabling the on-demand creation of networks of collaborative business processes. In B2B scenarios, security is clearly stated as a major requirement for the success and acceptance of business processes being enacted across multiple administrative domains [1]. A company participating in a VO still considers its business processes as assets which must not be exposed to the outside world in an uncontrolled manner. The outside world is hereby seen as everything outside the company's administrative domain, e.g. the company intranet. Research in the area of secure business process

enactment in VOs is for instance conducted in the TrustCoM¹ project. Security infrastructures as well as the security mechanisms have to meet VO security requirements across the multiple administrative domains spanned by the business processes. Such infrastructures are currently unavailable since security infrastructures are systematically deployed in a domain specific fashion. Security infrastructures, even on multiple layers ranging from network to application layer, were not meant to interact or collaborate across domains. Popular examples are e.g. firewalls on the network layer, protecting per definition everything outside a pre-set domain [2], considering everything outside as “evil” or Public Key Infrastructures (PKIs) [3], already offering configuration mechanisms for interaction and trust across domains. But in the case of PKIs, these are typically deployed only to be used in one domain being unaware of other PKIs, specifically unaware of what is certified and how the certification is conducted. Especially larger enterprises are frequently running an own PKI which is only used company internally. “Introducing” a PKI to another, a process commonly called cross-certification, requires a certain amount of administrative effort, meaning in terms of VOs a hindrance in the required dynamic, on demand security support for collaborative business processes.

In summary, security infrastructures are currently rather isolated, administrative islands of security domains. Since already running domain specific security infrastructures were in most cases expensive to deploy, this contribution explores an approach to connect such existing set-ups rather than entirely replacing running systems. The approach taken is to provide an additional semantic layer on top of existing security infrastructures, particularly using the concept of ontologies for describing declaratively the security properties, thus enabling an automatic approach for cross-domain certification.

1.2 Outline

Section 2 introduces the basic concepts and technologies of currently existing domain specific security set-ups. Further on in Section 3 a concrete example, namely a Virtual Organization in the setting of an aerospace collaborative engineering, is described to illustrate the practical relevance. Section 4 introduces the notion of ontology and presents how ontologies can be used to declaratively describe security properties. The paper concludes with related work in Section 5 and an overall conclusion in Section 6.

2 Current Domain Specific Security Set-Ups

Entities participating in a virtual organization (VO) are in first instance pursuing their own objectives, particularly their business interest. A participating entity therefore manages its own IT infrastructure, including security related systems, in its own administrative domain. When coming together in a VO context, they start a collaboration in order to achieve a common goal. Section 3 will provide a tangible example of this arrangement. This is also the start of an effort to inter-connect heterogeneous IT systems owned by different entities and most importantly managed and administered following domain

¹ <http://www.eu-trustcom.com>

specific policies. An approach published in [4] similar to the one described in this contribution, focussing on harmonizing policies across domains, supports the adopted solution. The described heterogeneity is one of the gravest hindrances in achieving proactive and automated secure collaboration across domains. Before introducing the envisioned solution by adding a semantic layer spanning domain specific security set-ups, a target example for later solution validation is prepared. Public key infrastructures (PKI) are popular security infrastructures offering mainly authentication, digital signature and asymmetric encryption for users in domains where their authority is respected. PKIs are typically run in a specific domain providing services only for this domain, obtruding themselves as dedicated examples since the provided services are essential for stated secure collaboration.

A PKI is a representative example in current times for the frequently mentioned term "domain specific security set-up". PKI is an abbreviation for Public Key Infrastructure. The term already mentions the foundation of a PKI quite often lying in asymmetric or public key cryptography. But public key cryptography is not essential, a PKI is rather a infrastructure based on a trusted third party (TTP) vouching and certifying for identities of principals, e.g. employees in a company, by binding public keys to principals. A PKI consists in simplified terms of the following components:

- a root of trust, the topmost certification authority (CA)
- a sequence of CAs inheriting trust from the root along a certification path
- certificates being the formal expression of trust assigned to a principal containing evidence of the issuing CA
- principals, the entities to whom certificates are assigned to

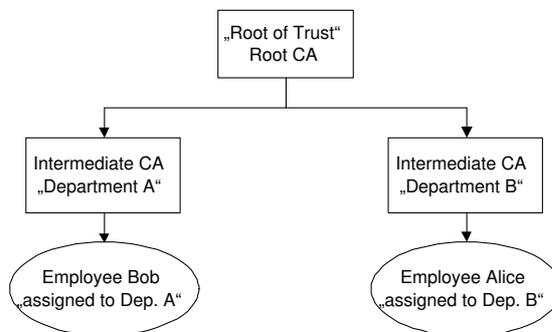


Fig. 1. simplified PKI structure

A CAs duty, the issuing of certificates is also based on public key cryptography. A CA owns a private/public key pair. Certification involves digital signature using the private key. The public key being embedded in a digital certificate, is attached as evidence for later signature verification to issued certificates. Since the root of trust has

to be protected as the component of utmost criticality, direct issuing of certificates to principals is the task of one or more CAs inheriting trust from the root. Inheriting is also based on digital certificates issued to them from the root. An inheriting CA's certificate also contains certificate information from the sequence of issuing CAs leading to the formation of a so-called certification path or chain. This high-level description of a PKI is not very detailed, nor exhaustive, but sufficient for the understanding of the following sections. More detailed information can be acquired at [5].

It was already mentioned that PKIs are frequently deployed by larger companies, mainly for authentication purposes. Such an installation then encompasses several complex computer systems assigned to different company intranet network segments and being tied into other integral IT parts such as corporate directories, e.g. based on LDAP. Directories are used to store the user's certificates containing the public key among other properties. The isolated management of PKIs does not necessarily imply that a PKI has to be unaware of another. Mechanisms like cross-certification exists where an CA C in a PKI P is able to sign a CA D's certificate being part of another PKI structure O. Cross-certification may be bi- or unidirectional, but the cross-certifying CA C is expressing a certain trust in the PKI hierarchy O. In reality, this mechanism is rarely used on an operational level among companies, since it is tedious to manage, involving a lot of careful negotiation on an inter-domain level and enforcing their outcome on an IT level in authorisations and access control [6].

Having mentioned certificates frequently, it is now time for a closer look on their structure. Certificates bind properties to principals, generally users. A common standard for digital certificates is X.509 [7]. A sample certificate structure in text form would look like the following:

```
* Certificate
  o Version
  o Serial Number
  o Algorithm ID
  o Issuer
  o Validity
    + Not Before
    + Not After

  o Subject
  o Subject Public Key Info
    + Public Key Algorithm
    + Subject Public Key
  o Issuer Unique Identifier
  o Subject Unique Identifier
  o Basic Constraints
  o Extensions
    + ...
* Certificate Signature Algorithm
* Certificate Signature
```

Basically, it contains information about

- the issuing CA
- the certification path
- the time period of validity
- the principal, the subject to whom it was issued
- the certificate's basic constraints enumerating its intended usage, such as authentication, encryption and (digital) signature

3 A B2B example in a Virtual Organization Set-Up

Virtual Organizations are considered as an important focus in recent European research conducted in the E-Business area. VOs are perceived as a way to model dynamically collaborating and rapidly evolving B2B scenarios. This section intends to commence a real-life example for domain-specific security setup-ups based on a PKI which inherently part of most existing VOs. More concretely, the chosen example is situated in an aerospace collaborative engineering context. Whenever a tender for manufacturing a new plane is announced, one organization or company alone is not able to perform all required tasks. The actors of the VO encompass all organizations dynamically, as fast as possible joining together to meet the tender's requirements. Figure 2 depicts a snapshot of the overall VO context, the actors meeting the immediate first objective of a plane design, other objectives encompass the actual manufacturing and maintenance which are not considered here.

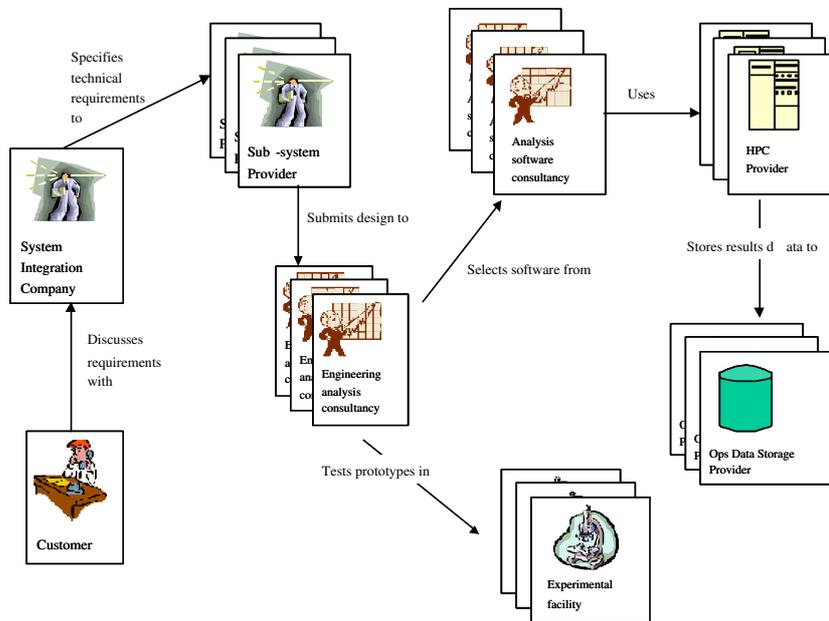


Fig. 2. VO example - collaborative engineering design (Source: TrustCoM project)

All depicted organizations are in general independent organizations and companies having their own agendas and are protecting their assets. For the latter, domain specific security set-ups such as PKIs are in place which have to interact in the dynamically evolving VO context [8]. In the following approach, the example focusses on the interaction of two partners, the Analysis Software Company and the High Performance Computing provider (HPC), in the following called Organizations A and B. All other

interaction involving three and more partners can be traced back to this basic example. We assume two employees, Smith working for A and Anderson working for B. A runs a CA A issuing certificates to its employees and B runs CA B.

Design and later manufacturing of planes is imposing a complex set of security requirements since the market competition is high or the tender issuer, e.g. the military already introduces these from the beginning. In the following, two simple security requirements are modeled:

- trust between A and B based on a list of trusted CA certificates
- confidentiality, based on the basic constraint encryption

Trust criteria are hereby rather simplified and static, this contribution doesn't intend to enter the highly speculative and vast arena of defining trust. Each organizations manages a list of CAs it trusts, if A trusts B this is represented by entering the certificate of CA B to A's list. Confidentiality means that information is only accessible to principals to whom it is intended to be accessible. A typical example is the set up of a confidential channel e.g. by the means of asymmetric cryptography using certificates (and corresponding private keys). The intended usage "Encryption" has therefore to be specified in the certificate's basic constraints field. In a real-life but more complex scenario, mechanisms like SSL would be used for confidential channels, using less resource consuming symmetric encryption for bulk data.

In summary, this chapter described a process beginning with trust establishment based on information offered by the respective domain specific PKIs. The process then continues asserting trust to meet e.g. business objectives. How this will be done, by modeling an ontology, will be described in the following chapter. In the end, the business objective is met by finally exploiting trust in conducting collaborative business.

The simplified basic example is now depicted in Figure 3. Typical use case for VOs is to establish a collaborative business process between employees, here coming from the organizations Orga A and Orga B. Both organizations have specified different security properties which not necessarily match to each other. We will show in the following section how to use an ontology to (i) describe both security properties declaratively and (ii) enable the automatic derivation of the fact whether employee Smith from Orga A is allowed to perform a collaborative business process with employee Anderson from Orga B.

4 Ontology for Domain-specific Security Setups

4.1 Introduction to Ontologies

In recent years ontologies have become a topic of interest in computer science. There are different 'definitions' in the literature of what an ontology should be, the most prominent being published by Tom Gruber [9]: *"An ontology is an explicit specification of a conceptualization. The term is borrowed from philosophy, where an Ontology is a systematic account of Existence. For AI systems, what 'exists' is that which can be represented."*

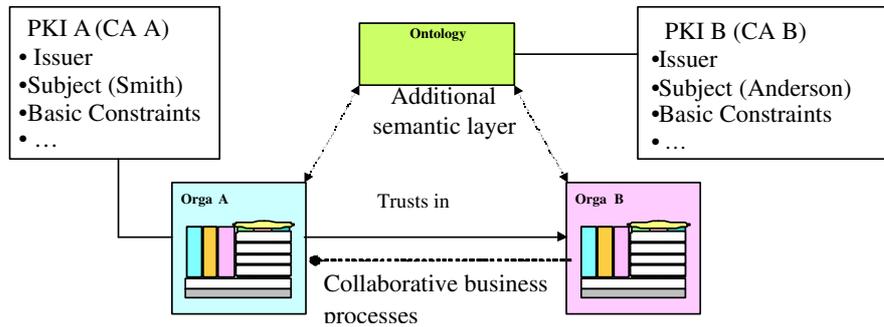


Fig. 3. Adding a semantic layer

A conceptualization refers to an abstract model of some phenomenon in the world by identifying the relevant concept of that phenomenon. Explicit means that the types of concepts used and the constraints on their use are explicitly defined. This definition is often extended by three additional conditions: “An ontology is an explicit, formal specification of a shared conceptualization of a domain of interest.” Formal refers to the fact that the ontology should be machine readable (which excludes for instance natural language). Shared reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted as a group. The reference to a domain of interest indicates that for domain ontologies one is not interested in modelling the whole world, but rather in modelling just the parts which are relevant to the task at hand. Methods and tools already exist to support engineering of ontologies (e.g. [10, 11]).

The informed reader might be aware that within the layered W3C² Semantic Web language stack currently the “logic” layer on top of the recommendation OWL [12] is being addressed. Thus, there might be a feasible solution to combine OWL with rules, but this is work to be done in future. Therefore we rely in the following example on F-Logic as representation language (*cf.* [13]), basically due to the fact that we want to model rules as mechanism for declaratively specifying business logic.

We will now introduce subsequently the building blocks of an example security ontology, namely concepts, relations, instances and rules. Such an ontology reflects the domain specific entities and allows for description of security logic supporting business processes in form of rules. Such an ontology enables users to declaratively describe their domain-specific security properties in a machine processable manner leading to improved automation. Thus, by using an inference engine such as Ontobroker [14] these security properties can be applied during runtime on the fly.

² <http://www.w3c.org/>

4.2 Concepts and Relations

Goal of this and the following subsections is to show the feasibility of formalizing the previously mentioned example scenario (see Figure 3). Please note that we will not describe all concepts and relationships etc., but rather the most relevant ones. Before we introduce the most relevant concepts we declare two namespaces, one for XML Schema and one as a default namespace.

```
ontons:xsd="http://www.w3.org/2001/XMLSchema"
ontons="http://x.y.z"
```

Core concept is a `Certificate` according to the X.509 standard (see Section 2) which has certain properties. These properties are modelled on the one hand as relations such as `version` or `serial Number` who have XML Schema datatypes such as `STRING` as range, and on the other hand as relations such as `issuer` who have another concept such as `Certification Authority` as range.

```
#Certificate[#version=>>xsd#STRING;
#serial_Number=>>xsd#STRING;
#algorithm_ID=>>xsd#STRING;
#issuer=>>#Certification_Authority;
#validity_not_before=>>xsd#STRING;
#validity_not_after=>>xsd#STRING;
#subject=>>#Principal;
#basic_constraints=>>#Basic_Constraints].
```

A `Certification Authority` typically is managed by an `Organization`. The concept `Principal` is typically be part of a role hierarchy which we will not elaborate on here. We will show later in the next section what kind of `Basic Constraints` exist.

```
#Certification_Authority[#has_X500_name=>>xsd#STRING;
#is_managed_by=>>#Organization].

#Principal[#has_X500_name=>>xsd#STRING].

#Basic_Constraints.
```

Each `Organization` typically has a list of `Trusted Root Certificates` which makes explicit in which `Certification Authority` an organization trusts in.

```
#Organization[#employs=>>#Employee;
#manages=>>#Certification_Authority;
#has_list_of=>>#Trusted_Root_Certificates].

#Trusted_Root_Certificates[#has_X500_name=>>xsd#STRING;
#trusts_in=>>#Certification_Authority].
```

Central for business activities are `Employees` who work for an `Organization` and have the role of a `Principal`. Each time they want to perform a `Task` which has a certain level of confidentiality with another partner in the VO (such as a collaborative business task), they will need a `Certificate`.

```
#Employee[#works_for=>>#Organization;
#has_role=>>#Principal;
#owns_certificate=>>#Certificate;
#performs=>>#Task].

#Task[#involves=>>#Employee;
#confidentiality=>>xsd#BOOLEAN].
```

4.3 Exemplary Instances

The ontology provides a schema for general purposes which can be reused by different applications. We now instantiate the given ontology e.g. with two organizations Orga A and Orga B, two employees Smith and Anderson who at the same time play the role of being a specific Principal, e.g. Principal A.

```
#Orga_A:#Organization.
#Orga_B:#Organization.

#Smith:#Employee[#works_for->>#Orga_A;
  #has_role->>#Principal_A;
  #performs->>#Business_Task_1].
#Anderson:#Employee[#works_for->>#Orga_B;
  #has_role->>#Principal_B].

#Principal_A:#Principal.
#Principal_B:#Principal.
```

Each employee owns a specific Certificate, both having different basic constraints such as Authentication, Encryption and Signature.

```
#Certificate_A:#Certificate[#subject->>#Principal_A;
  #basic_constraints->>#Authentication;
  #basic_constraints->>#Encryption;
  #basic_constraints->>#Signature].
#Certificate_B:#Certificate[#subject->>#Principal_B;
  #basic_constraints->>#Authentication;
  #basic_constraints->>#Encryption].

#Encryption:#Basic_Constraints.
#Authentication:#Basic_Constraints.
#Signature:#Basic_Constraints.
```

CA A is a specific Certification Authority which is managed by the organization Orga A who has a specific list of Trusted Root Certificates, namely TRC A. Furthermore, Orga A trusts himself and Orga B, but Orga B only trusts himself (and not yet or not in general Orga A).

```
#CA_A:#Certification_Authority[#is_managed_by->>#Orga_A].
#CA_B:#Certification_Authority[#is_managed_by->>#Orga_B].

#Orga_A[#has_list_of->>#TRC_A].

#TRC_A:#Trusted_Root_Certificates[#trusts_in->>#CA_A;
  #trusts_in->>#CA_B].
#TRC_B:#Trusted_Root_Certificates[#trusts_in->>#CA_B].
```

Last, but not least, the employee Anderson is involved in a specific business task, namely Business Task 1.

```
#Business_Task_1:#Task[#involves->>#Anderson].
```

4.4 Rules

To describe business logic declaratively we use so-called rules. Simple rules e.g. allow for a more complete knowledge, e.g. saying that *if “X employs Y” then “Y works for X” and vice versa.*

```
FORALL X,Y
  X[#employs->>Y] <-> Y[#works_for->>X].
```

A more complex rule is the following one which basically says if “*X is a specific employee who has the role of being a specific Principal Y*” and “*Y is a specific Certificate who has as subject the specific Principal Y*” then “*the specific employee X owns the specific certificate Y*”. Thus, this rule links different entities playing a role in the security properties, namely employees, principals and certificates.

```
FORALL X,Y,Z
  X[#owns_certificate->>Y]
  <-
  X:#Employee[#has_role->>Z:#Principal] AND
  Y:#Certificate[#subject->>Z].
```

Finally, the following rule allows us to derive the fact whether an employee is able to perform a security-enabled task with another employee coming from a different organization. The rule itself reads similar to the previous one.

```
FORALL X
  X[#confidentiality->>"true"]
  <-
  EXISTS A,B,C,D,F,G,H,I
  A:#Employee[#works_for->>B:#Organization] AND
  A[#owns_certificate->>C:#Certificate] AND
  B[#has_list_of->>D:#Trusted_Root_Certificates] AND
  D[#trusts_in->>F:#Certification_Authority] AND
  F[#is_managed_by->>G:#Organization] AND
  H:#Employee[#works_for->>G] AND
  H[#owns_certificate->>I:#Certificate] AND
  C[#basic_constraints->>#Encryption] AND
  I[#basic_constraints->>#Encryption] AND
  X:#Task[#involves->>A] AND
  X[#involves->>H].
```

To sum up, in a practical setting each organization makes its security properties explicitly available as an ontology. Additional rules make existing knowledge more complete and allow for linking of existing security properties in a previously unforeseen manner to support the dynamic environment of VOs.

As an afterthought, the presented work did not elaborate on the topic of deployment yet, who will host the ontology. It would be possible that the ontology is run and maintained, from a security perspective, by a trusted third party or each organisation runs its own instance. The latter would make the administration process regarding updates and corrections more difficult.

5 Related Work

Related work on the semantic layer as well as supporting technologies and mechanisms is conducted on different layers in similar fields, but not quite related to integrating domain specific security set-ups in B2B scenarios. KaOS [4] for instance is looking in the field of policy and domain services and how these may be connected on the semantic layer across domains. In the area of trust management and security, approaches based on a particular domain specific and authoritative PKI called SPKI/SDSI [3] using different forms of mediation are taken [15][16]. This work is still below the semantic layer

and would still need a semantic mapping of expressed domain specific security properties. In [17], similar work using properties stored in directory structures, specifically LDAP, was conducted. The work is therefore emphasizing on pure property mapping, but not relying on embedding these on a domain specific security set-up such as a PKI. Research from a trust perspective in VOs and similar concepts, even on logic level, is conducted in [18]. This work is laying a foundation for security and trust requirements towards the semantic layer. On the level of supporting technologies, particularly the (web) service layer, adding semantics to services is a recent research topic [19]. Such development will influence the standardization conducted in the WS-I gremium, being responsible for most security related web service (WS-) standards as well, such as WS-Security, WS-Trust, WS-Federation or WS-(Secure)Conversation. Mentioned standardization can not be considered mature at this stage, not all standards are yet fully specified. On a deeper technological level, protocols like SAML which is standardized in the OASIS body, may be used to provide interoperability of security related statements, such as assertions. The presented semantic solution would be supported on the transport layer by such mechanisms.

6 Conclusion

To conclude, we demonstrated the feasibility of using the semantic layer namely an ontology to span administrative domains. The emphasis was put on domain specific security properties which should be known, understood and employed to meet security requirements which extend a domain horizon and become cross-domain properties. The approach was successfully exercised in a concrete example, a VO context in collaborative engineering. However, the approach was not without hurdles. In the security community especially in trust management, research is conducted up to a service layer – but not further. Solutions requiring property mapping and translation are expected from the semantic layer – and rightfully though. The introduced example validated this expectation. But what is not considered in such expectations is the administrative effort to initialize the required ontology until it is able to perform its duties. This work raised interesting questions for future work, e.g. how easy will it be, to port an ontology “instance” like the one derived here and port it to another setting or how much of the rule set and other concepts can be re-used? How much effort is it in average to link a new organization to an existing VO? We expect further work in this area especially since the upper layers of the Semantic Web pyramid which are to be solved include proof and trust.

Acknowledgments

Research reported in this paper has been partially financed by the EU in the IST projects TrustCoM (IST-2003-01945)³ and SEKT (IST-2003-506826)⁴. We would like to thank our colleagues for fruitful discussions.

³ <http://www.eu-trustcom.com/>

⁴ <http://sekt.semanticweb.org>

References

1. Herbert, J.: Introducing security to the small business enterprise. GIAC (2003)
2. Robinson, P., Haller, J.: Revisiting the firewall abolition act. HICCS-36 (2002)
3. SPKI-WorkingGroup: SPKI/SDSI. IETF RFC 2692 and 2693 (1996) available at <http://theworld.com/~cme/html/spki.html#1-SPKI/SDSI>.
4. Uszok, A., Bradshaw, J., Jeffers, R., Suri, N., Hayes, P., Breedy, M., Bunch, L., Johnson, M., Kulkarni, S., Lott, J.: KAOs policy and domain services: Toward a description-logic approach to policy representation, deconfliction, and enforcement. Workshop on Web Services and Agent-based Engineering (2003)
5. Gutman, P.: Everything you never wanted to know about pki but were forced to find out. (2001) available at <http://www.cs.auckland.ac.nz/~pgut001/pubs/pkitutorial.pdf>.
6. Turnbull, J.: Cross-certification and pki policy networking. Entrust (2000)
7. Brickley, D., Guha, R.V.: X.509. IETF RFC 2459 (2004) available at <http://www.ietf.org/html.charters/pkix-charter.html>.
8. Strader, T., Lin, F., Shaw, M.: Information structure for electronic virtual organization management, decision support systems. (1998) 75–94
9. Gruber, T.: Towards principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies **43** (1995)
10. Sure, Y., Angele, J., Staab, S.: OntoEdit: Multifaceted inferencing for ontology engineering. Journal on Data Semantics LNCS (2003) 128–152
11. Pinto, H., Tempich, C., Staab, S., Sure, Y.: DILIGENT: Towards a fine-grained methodology for Distributed, Loosely-controlled and evolvinG Engineering of oNTologies. In: Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004), August 22nd - 27th, 2004, Valencia, Spain. (2004)
12. Smith, M.K., Welty, C., McGuinness, D.: OWL Web Ontology Language Guide (2004) W3C Recommendation 10 February 2004, available at <http://www.w3.org/TR/owl-guide/>.
13. Kifer, M., Lausen, G., Wu, J.: Logical foundations of object-oriented and frame-based languages. Journal of the ACM **42** (1995) 741–843
14. Decker, S., Erdmann, M., Fensel, D., Studer, R. In: Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information. Kluwer Academic Publisher (1999) 351–369
15. Biskup, J., Karabulut, Y.: A hybrid pki model with an application for secure mediation. 16th Annual IFIP WG 11.3 Working Conference on Data and Application Security (2002) 271–282
16. Karabulut, Y.: Towards a next-generation trust management infrastructure for open computing systems. SPPC: Workshop on Security and Privacy in Pervasive Computing (2004)
17. Ahmedi, L., Marron, P.J., Lausen, G.: Ldap-based ontology for information integration. 9. Fachtagung Datenbanksysteme in Buero, Technik und Wissenschaft (2001)
18. Josang, A.: Logic for uncertain probabilities. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (2001) 279–311 Available at: <http://security.dstc.edu.au/papers/logunprob.pdf>.
19. McIlraith, S.A., Son, T.C., Zeng, H.: Semantic web services. IEEE Intelligent Systems **16** (2001) 46–53