Using Context- and Content-Based Trust Policies on the Semantic Web

Christian Bizer
Freie Universität Berlin
Institut für Produktion, Wirtschaftsinformatik und OR
Garystr. 21, D-14195 Berlin, Germany
+49 30 838 54057
bizer@wiwiss.fu-berlin.de

Radoslaw Oldakowski
Freie Universität Berlin
Institut für Produktion, Wirtschaftsinformatik und OR
Garystr. 21, D-14195 Berlin, Germany
+49 30 838 52760
cax@wiwiss.fu-berlin.de

ABSTRACT

The current discussion about a future Semantic Web trust architecture is focused on reputational trust mechanisms based on explicit trust ratings. What is often overlooked is the fact that, besides of ratings, huge parts of the application-specific data published on the Semantic Web are also trust relevant and therefore can be used for flexible, fine-grained trust evaluations. In this poster we propose the usage of context- and content-based trust mechanisms and outline a trust architecture which allows the formulation of subjective and task-specific trust policies as a combination of reputation-, context- and content-based trust mechanisms.

Categories and Subject Descriptors

I.2.4 [Artificial Intelligence]: Knowledge Representation Formalisms and Methods – semantic networks.

General Terms

Reliability, Security, Human Factors

Keywords

Semantic Web, Trust Mechanisms, Trust Policies, Named Graphs

1. INTRODUCTION

The Semantic Web will be an open, dynamic network of independent information providers all having different views of the world, different levels of knowledge, and different intentions. Thus, statements published on the Semantic Web have to be seen as *claims* rather than as facts. The central enabling factor in realising the vision of the Semantic Web, as an open information sharing architecture, is the question whether it is possible to develop a pragmatic trust architecture which allows information consumers to decide which claims are trustworthy.

2. TRUST MECHANISMS AND POLICIES

A trust policy is a subjective procedure used for evaluating the trustworthiness of information in a specific situation. In everyday life, we use a wide range of trust policies. These policies depend on the specific situation, our subjective preferences, our past experiences and the trust relevant information available: We might trust Andy on restaurants but not on computers, trust professors on their research field, believe foreign news only when they are reported by several independent sources and buy only from sellers on eBay who have more than 100 positive ratings.

The future Semantic Web is supposed to be a dense mesh of interrelated information, similar to the information perception situation we face in the offline world. Thus, we argue, a trust architecture can support a similarly wide range of trust policies as used offline. Figure 1 shows an abstract view of the trust situation on the Semantic Web. All information which could be used in trust evaluations is shaded grey.

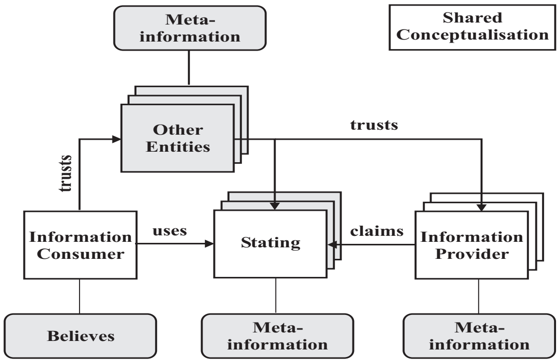

Figure 1: Trust Situation on the Semantic Web

Three general trust mechanism build on this information:

1. *Reputation-Based Trust Mechanisms* include rating systems like the one used by eBay and Web-Of-Trust mechanisms. All trust architectures proposed for the Semantic Web so far fall into this category [1,2,3]. The general problem with these approaches is that they require explicit and topic-specific trust ratings and that providing such ratings and keeping them up-to-date puts an unrealistically heavy burden on information consumers.

2. *Context-Based Trust Mechanisms* use metainformation about the circumstances in which information has been claimed, e.g. who said what, when and why. They include role-based trust mechanisms, using the author's role or his membership in a specific group, for trust decisions. Example policies from this category are: "Prefer product descriptions published by the manufacturer over descriptions published by a vendor" or "Distrust everything a vendor says about its competitor." An example policy using the statement context is "Distrust all product ratings that are older than a year."

3. *Content-Based Trust Mechanisms:* These approaches do not use metadata about information, but rules and axioms together with the information content itself and related information about the same topic published by other authors [4]. Example policies following this approach are "Believe information which has been

WWW 2004, May 17-22, 2004, New York, NY USA.
ACM 1-58113-912-8/04/0005.

stated by at least 5 independent sources." or "Distrust product prices that are more than 50% below the average price."

Context- and content-based trust mechanisms do not require explicit ratings, but rely on the availability of a dense mesh of background information. On the Semantic Web such a mesh will be available and therefore can be used for trust decisions.

3. TRUST ARCHIECTURE

The Semantic Web requires an open trust architecture without central trusted third parties. The trustworthiness of information should be subjectively evaluated by each information consumer. The trust architecture should not exclude information providers which have not been rated or do not publish trust relevant information in a specific way, e.g. sign their information. On the other hand, the system should be able to use all trust relevant information (signatures, context information, related information and ratings) published or generated during the information gathering process (source URL, crawling date). Users have different subjective preferences for specific trust mechanisms and – even in the same situation – different trust requirements. As a consequence an architecture should allow users to formulate subjective and task-specific trust policies combining different trust mechanisms. The key factor for building trust is the user's understanding of the information and the metrics used in trust evaluations. Thus an architecture should be able to justify its trust decisions and support something like Tim Berners-Lee's "Oh yeah?"-button [5], meaning that the user can click on every piece of information within an application and get explanations why she should trust the information.

We are prototyping a trust architecture following the principles stated above. Our architecture can be logically divided into four layers: The *Information Integration Layer* handles the aggregation of information from different sources and adds provenance metadata to the information. If information is digitally signed and the signature can be verified, the information is marked as "FromVerifiedOrigin" The *Repository Layer* stores the aggregated information. The *Query and Trust Evaluation Layer* handles the actual trust decisions using query-specific trust policies. The *Application and Explanation Layer* on which the retrieved information is used within an application context and which provides functionality to browse through explanations why data should be trusted.

For storing the aggregated data we use Named Graphs [6], an extension to RDF which allows avoiding the usage of reification when attaching provenance information to graphs. For querying the aggregated data we use *TriQL.P*, a query language extending TriQL [7]. TriQL is similar to RDQL but uses graph patterns instead of triple patterns for querying named graphs. TriQL.P allows the expression of trust-policies within queries and returns *justification trees* together with the query results. It supports set operations and different ranking mechanisms like Web-of-Trusts. The example TriQL.P query below retrieves all persons with the skill "Programming", based only on claims by people who have an affiliation to at least 3 projects involving programming. The variable ?b refers to the names of all graphs which contain information about persons with the skill "Programming". The patterns in the WHERE-clause are transformed into a pattern tree during query execution.

```
SELECT ?a
WHERE ?b (?a <km:skill> <km:Programming>.
          ?a <rdf:type> <km:Person>)
         (?b <swp:assertedBy> ?c.
          ?c <swp:authority> ?d)
         (?d <km:affiliation> ?e)
         (?e <rdf:type> <km:Project>.
          ?e <km:topic> <km:Programming>)
AND COUNT(?e) > 2
```

TriQL.P returns variable bindings together with a justification tree for each set of bindings. A justification tree contains the matching bindings for each pattern in the pattern tree. Applications can use justification trees to explain why retrieved information fulfils the trust requirements formulated within a query. In our example, the justification tree would contain information about the authors and their projects. A justification tree attached to a binding returned by a query, which uses a reputation-based trust mechanism, would include all known ratings for the selected object. Compared to [8], our concept of justification trees focuses on explaining the primary data which has been used in trust decisions, while their approach focuses on the explanation of distributed proof traces. More information about our trust architecture, example queries and justification trees are found at: http://www.wiwiss.fu-berlin.de/suhl/bizer/TriQLP.

4. CONCLUSION

A Semantic Web trust architecture should not be based exclusively on explicit trust ratings but use all trust relevant information available. It should allow users to formulate subjective and task-specific trust polices as a combination of different trust mechanisms. We think that the usage of context- and content-based trust mechanisms within Semantic Web applications presents a promising path for future research.

5. REFERENCES

[1] R. Agrawal, P. Domingos, and M. Richardson. Trust Management for the Semantic Web. In Proceedings of the 2nd International Semantic Web Conference, ISWC2003, 2003.

[2] J. Golbeck, B. Parsia, and J. Hendler. Trust Networks on the Semantic Web. In Proceedings of the 7th International Workshop on Cooperative Intelligent Agents, CIA2003, 2003.

[3] C. Bizer. Semantic Web Trust and Security Resource Guide, 2003. http://www.wiwiss.fu-berlin.de/suhl/bizer/SWTSGuide

[4] J. M. Reagle Jr. Finding Bacon's Key - Does Google Show How the Semantic Web Could Replace Public Key Infrastructure?, 2002. http://www.w3.org/2002/03/key-freetrust.html

[5] T. Berners-Lee. Cleaning up the User Interface, 2003. http://www.w3.org/DesignIssues/UI.html

[6] J. Carroll, C. Bizer, P. Hayes, P. Strickler. Named Graphs, Provenance and Trust. Technical Report HPL-2004-57, Hewlett Packard Labs, 2004.

[7] C. Bizer. TriQL Specification, 2004. http://www.wiwiss.fu-berlin.de/suhl/bizer/TriQL/Spec.htm

[8] D. McGuinness and P. Pinheiro da Silva. Infrastructure for Web Explanations. In Proceedings of the 2nd International Semantic Web Conference, ISWC2003, 2003.