# A Trust and Reputation Framework *

Francisco Moyano

Department of Computer Science, University of Malaga, 29071, Málaga, Spain
moyano@lcc.uma.es

**Abstract.** The Future Internet is posing new security challenges as their scenarios are bringing together a huge amount of stakeholders and devices that must interact under unforeseeable conditions. In addition, in these scenarios we cannot expect entities to know each other beforehand, and therefore, they must be involved in risky and uncertain collaborations. In order to minimize threats and security breaches, it is required that a well-informed decision-making process is in place, and it is here where trust and reputation can play a crucial role. Unfortunately, services and applications developers are often unarmed to address trust and reputation requirements in these scenarios. To overcome this limitation, we propose a trust and reputation framework that allows developers to create trust- and reputation-aware applications.

**Supervisors:** Carmen Fernandez-Gago and Javier Lopez

## 1 Introduction: Problem and Motivation

Future Internet (FI) scenarios bring together multiple entities, namely stakeholders and devices, that need to collaborate in order to reach their goals. Should these entities knew each other beforehand, upfront mechanisms could be in place at design-time in order to ensure that these collaborations have a successful ending for all parties. However, this cannot be assumed. Therefore, it is required to guarantee a successful ending even under risky and uncertain conditions, which generally involves making good decisions. These conditions present a breeding ground for trust.

Even when the concept of trust is not standardized, it is agreed that it can be a valuable tool to leverage decision-making processes. The concept and implications of trust are embodied in trust models, which define the rules to process trust in an automatic or semi-automatic way within a computational setting. For the last twenty years, many models have been proposed, each one targeting different contexts and purposes, and with their own particularities.

*The research leading to these results have received funding from the European Community's Seventh Framework Programme FP7/2007-2013 as part of the Network of Excellence NESSoS (www.nessos-project.eu) under grant agreement number 256980. The first author is funded by the Spanish Ministry of Education through the National F.P.U. Program.

One issue with trust models is that they are usually built on top of an existing application in an ad-hoc manner in order to match the specific needs of the application and its environment, limiting the models' re-usability. Furthermore, most models do not distinguish explicitly between trust and reputation, nor do they provide guidelines to combine these notions to yield more solid results.

We believe that this approach is not adequate and that developers should be provided with some mechanisms to systematically incorporate trust and reputation models into their services and applications.

The rest of the paper is organized as follows. Section 2 describes the goals that we are pursuing. In Section 3 we explain the research methodology that is being followed and discuss the work that has been carried out up to now. Finally, the conclusions and some lines for future research are presented in Section 4.

## 2 Aims and Goals

Our main goal is the specification, design and implementation of a development framework that allows developers to implement trust- and reputation-aware applications. The framework must expose an Application Programming Interface (API) in order to make its functionalities accessible, and it must also provide hot spots where trust models can be customized to fit the application needs.

An important sub-goal that is derived from the main expected contributions is the provision of insight into trust and reputation. It is often the case that these concepts are considered as being synonyms or are used interchangeably, however they are quite different notions that need to be considered separately. Building a development framework requires performing a domain analysis in the framework targeted area, in this case, trust and reputation. Not only can this domain analysis shed light on concepts such as trust or reputation, but also on the trust models internal workings.

The main expected contribution of this research to the field of Engineering Secure Software and Systems is two-fold: on the one hand, by providing developers with a tool like a trust and reputation framework, we foster thinking over trust and reputation requirements from the very beginning. On the other hand, as applications are developed by using the framework, trust and reputation models are naturally incorporated within the application itself, and not as patches added after-the-fact, as it is the standard nowadays. Thus, trust models can use all the information available to the application in a more efficient way.

## 3 Research Methodology

This section summarizes the research methodology that is being followed. It is divided into six phases, each one further elaborated in its own section. For each phase, we state whether it is completed or further work needs to be done, and we also outline their main findings and results.

### 3.1 Phase 1: Literature Review

Surveys, such as the one by Jøsang, Ismail and Boyd [5] or the one by Ruohomaa and Kutvonen [11] are the best starting point to obtain a solid knowledge of the work carried out in trust and reputation over a period of time, and they constitute the main source for the next phase: the domain analysis. Other interesting contributions include those that provide assistance to developers with creating trust and reputation implementations. In this direction, we conducted research on architectural styles [12], frameworks [7] [2] and middlewares [6] [4] where trust and reputation are the core concept.

Some drawn conclusions are that most works do not provide enough margin of customization and lack of a framework-oriented approach. In addition, no existing contributions differentiate between the notions of trust and reputation, as they tend to focus on just one of them, usually reputation.

Even though this task was already finished, it is required to continuously check out new interesting papers.

### 3.2 Phase 2: Domain Analysis

A domain analysis is of paramount importance when building a development framework [3], and for this analysis to be complete, it may be required to look up new literature that helps to fill some gaps that may have arisen.

The main contribution is a conceptual framework that gathers and relates the most important concepts in trust and reputation models. This framework is represented in the form of UML diagrams, like the one depicted in Figure 1. As explained in an earlier contribution [8], this conceptual framework also serves as a comparison framework under which different trust and reputation models can be compared.

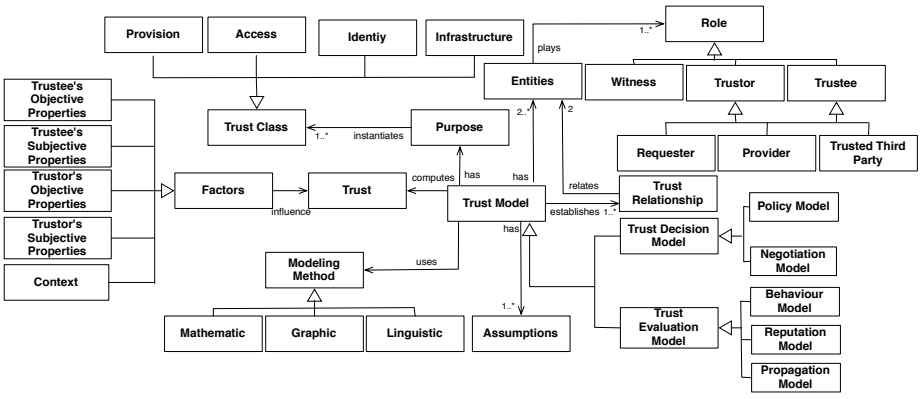


Fig. 1: Common Concepts for Trust Models

Also, in our earlier work [8], we classified trust models into two types: decision models, tightly related to the authorization problem, and evaluation models, where the evaluation of trust according to several influencing factors is the most important consideration.

Even though this phase is also completed, the analysis should be refined as new relevant papers arise.

### 3.3 Phase 3: Requirements Elicitation

The previous phase conducted an exhaustive analysis on trust and reputation. This analysis assisted in determining the requirements that a trust and reputation framework must fulfil. Since accommodating all possible trust models in a single framework may be a daunting task, we decided to focus on evaluation models. A list of requirements can be found in an earlier work [10].

Even though this phase is finalized, some new requirements may arise as a consequence of new relevant literature or due to the architecture and design phases. One of our findings is that evaluation models are centred around the notion of trust metric. Trust metric uses a computation engine to yield a trust or reputation value given a set of variables. The difference between many evaluation models stems from (i) the variables used in the computation and (ii) the computation engine used to aggregate these variables into a simple value or a tuple of values. Therefore, one of the most important requirements for a trust and reputation framework is to allow developers to define their own metrics. Time and uncertainty are two important factors as well, and developers should be allowed to include them. The former may be used to consider freshness in trust values, whereas the latter refers to how reliable a trust value is.

### 3.4 Phase 4: Architecture

This phase consists of sketching the high-level software structure that supports the requirements elicited in the previous phase. A half-way technical and conceptual architecture was discussed in earlier works [10] [9]. A recent work[1] provides greater insight into the technical details of a possible architecture, and even guidelines are given for implementation of some of the components and their communication mechanisms.

At the architectural level, building a trust and reputation framework requires planning in two fronts. On the one hand, we need to carefully design an easy yet flexible API that allows connecting any application to a trust server. On the other hand, the framework must provide enough hot spots to support the customization of the trust server behaviour at runtime in order to accommodate new trust and reputation models.

The type of application that we want to build by using the framework determines the design of the aforementioned factors: API and hot spots. In this

[1] We cannot provide the reference as the work is currently under review.

sense, we think of two types of applications that follow two different architectural styles: client-server applications and peer-to-peer applications.

The first one requires the developer to define the interactions between an application server and a trust server, and the trust server holds information about the whole system. In the second approach, each peer holds an instance of the trust server, which holds only partial information about the whole system.

The architecture proposed in our recent work[1] was originally designed to support the client-server approach, even though we think it can be tailored in order to support the peer-to-peer architectural style.

### 3.5 Phase 5: Design and Implementation

This phase elaborates on the architecture in order to refine the components into sub-components and modules. Inner data structures are also detailed and the database schemas and tables are fully specified. This refinement goes on until the implementation of each module is made easy. This phase remains unfinished.

### 3.6 Phase 6: Validation

The last phase consists of validating the framework implementation by developing a trust-aware application in the scope of e-Health and/or SmartGrid, which have been identified as the two main NESSoS[2] scenarios [1].

It is likely that we observe certain deficiencies and limitations of the framework in a real application. Actually, any framework requires iterations in order to be able to accommodate a wide range of applications. Therefore, the output of this phase could help to improve the architecture and design of the framework.

## 4 Conclusions and Future Work

New Future Internet applications will need support from trust and reputation services for their successful adoption. Yet these services have been laid aside and are very often considered once an application is already deployed and running. At that moment, adding trust and reputation features may be hard, and may lead to poor and, above all, barely reusable solutions.

We propose a trust framework that assists developers in the task of creating services and applications that need trust and reputation models. Examples of such applications are those proposed in the NESSoS project, and validation is to be done in their scope.

As future work, we are planning to research on how reconfiguration mechanisms can leverage trust models during the service or application lifetime. The trend in Software Engineering is towards adapting the software at runtime to new requirements or new environmental conditions, changing the architecture itself without the need for re-implementation. We would like to obtain insight into how the trust framework could exploit advances in this direction in order to support self-adapting trust models.

[2] www.nessos-project.eu

## References


1. Selection and Documentation of the Two Major Application Case Studies. NESSoS Deliverable 11.2, October 2011.
2. Vinny Cahill, Elizabeth Gray, Jean-Marc Seigneur, Christian D. Jensen, Yong Chen, Brian Shand, Nathan Dimmock, Andy Twigg, Jean Bacon, Colin English, Waleed Wagealla, Sotirios Terzis, Paddy Nixon, Giovanna di Marzo Serugendo, Ciaran Bryce, Marco Carbone, Karl Krukow, and Mogens Nielsen. Using Trust for Secure Collaboration in Uncertain Environments. *IEEE Pervasive Computing*, 2(3):52–61, July 2003.
3. Mohamed E.Fayad, Douglas C.Schmidt, and Ralph E.Johnson. *Building Application Frameworks: Object-Oriented Foundations of Framework Design*. Wiley, Septembre 1999.
4. Chern Har Yew. *Architecture Supporting Computational Trust Formation*. PhD thesis, University of Western Ontario, London, Ontario, 2011.
5. Audun Jøsang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, March 2007.
6. Rolf Kiefhaber, Florian Siefert, Gerrit Anders, Theo Ungerer, and Wolfgang Reif. The Trust-Enabling Middleware: Introduction and Application. Technical Report 2011-10, Universitätsbibliothek der Universität Augsburg, Universitätsstr. 22, 86159 Augsburg, 2011. http://opus.bibliothek.uni-augsburg.de/volltexte/2011/1733/.
7. Adam J. Lee, Marianne Winslett, and Kenneth J. Perano. TrustBuilder2: A Reconfigurable Framework for Trust Negotiation. In Elena Ferrari, Ninghui Li, Elisa Bertino, and YÃ$\frac{1}{4}$cel Karabulut, editors, *IFIPTM*, volume 300 of *IFIP Conference Proceedings*, pages 176–195. Springer, 2009.
8. Francisco Moyano, Carmen Fernandez-Gago, and Javier Lopez. A conceptual framework for trust models. In Simone Fischer-Hübner, Sokratis Katsikas, and Gerald Quirchmayr, editors, *9th International Conference on Trust, Privacy & Security in Digital Business (TrustBus 2012)*, volume 7449 of *Lectures Notes in Computer Science*, pages 93–104, Vienna, Sep 2012 2012. Springer Verlag, Springer Verlag.
9. Francisco Moyano, Carmen Fernandez-Gago, and Javier Lopez. Implementing trust and reputation systems: A framework for developers' usage. In *International Workshop on Quantitative Aspects in Security Assurance*, Pisa, Sep 2012 2012.
10. Francisco Moyano, Carmen Fernandez-Gago, and Javier Lopez. Building trust and reputation in: A development framework for trust models implementation. In *8th International Workshop on Security and Trust Management (STM 2012)*, Pisa, In Press.
11. Sini Ruohomaa and Lea Kutvonen. Trust management survey. In *Proceedings of the Third international conference on Trust Management*, iTrust'05, pages 77–92, Berlin, Heidelberg, 2005. Springer-Verlag.
12. Girish Suryanarayana, Mamadou H. Diallo, Justin R. Erenkrantz, and Richard N. Taylor. Architectural Support for Trust Models in Decentralized Applications. In *Proceeding of the 28th international conference*, pages 52–61, New York, New York, USA, 2006. ACM Press.