A Peer-to-Peer Multidimensional Trust Model for Digital Ecosystems

Mihaela Ion*, Andrea Danzi*, Hristo Koshutanski† and Luigi Telesca*
*CREATE-NET, Via Solteri 38, Trento, Italy
{mihaela.ion, andrea.danzi, luigi.telesca}@create-net.org
†Computer Science Department, University of Malaga, Spain
hristo@lcc.uma.es

***Abstract*— The aim of the Digital Ecosystem (DE) initiative[1] is to encourage Small and Medium-sized Enterprises (SMEs) to use the Internet and to adopt ICT technologies that would make them more innovative and competitive in the market. In order for a DE to take-off, specific solutions are required that are practical and easy to adopt, and that address the organizational and infrastructural particularities of the networked communities.**

This paper proposes a new trust model for DEs which has several innovative features. The model is based on the concept of social networks and addresses trust at different levels: user, data, service and node. The model allows fast bootstrapping of trust by importing existing trust relationships from outside DE systems and by relying on certificates issued by trusted authorities external to the DE. Furthermore, trust can be measured in a variety of contexts by using user-defined tags – folksonomy[2]. The model abstracts from specific reputation algorithms by providing necessary interfaces for plugging-in those on one's own choice.

I. Introduction

A digital ecosystem (DE) [1] is composed of heterogeneous and autonomous users, companies and resources which interact in a complex, distributed and dynamic environment. The complexity of interactions between different institutions is increased by the fact that institutions sometimes compete against each other and other times collaborate with each other and form stable and unstable federations. Digital ecosystems are interconnected by a network to form a complex and dynamic environment.

Trust represents the basis of every human interaction and without it there would be no collaboration and no society [2]. In a DE, actors decide to interact or not based on the mutual trust they have in each other.

A digital ecosystem, similar to a natural ecosystem, involves not only users or software that act on behalf of users, known as agents, but also the environment in which they interact. A trust model for DEs is not complete if it does not consider several levels of trust like: trust in agents (users), data or knowledge, services, nodes, infrastructure, and the DE as a whole.

Entities in a DE evolve and adapt to constantly changing conditions as well as to the appearance of new members, and so trust relations between entities form and adapt. The trust entities have in each other can change with each interaction or transaction. Because of that, the model relies on each transaction outcome to adjust trust values. Moreover, users and companies in a DE are involved in different business relationships outside the system. It is important to use these relations inside the DE especially because this will allow newcomers to create trust relations more easily.

Users have different levels of expertise in different domains, as well as different parameters for measuring the quality of services (QoS). A comprehensive trust model needs to be able to accommodate the different domains on which trust statements are expressed.

A. Ecosystem-oriented Architectures

Today users and organizations employ a broad set of digital components, such as software products, business services, knowledge (documents, e-mails, portals, wikis) and data structure representing business objects. An Ecosystem Oriented Architecture (EOA) [3] can be defined as a meta-level architecture for DE, allowing for the description of digital components and processes that are involved. The idea behind EOA is the extension of the classical Service Oriented Architecture (SOA) in a distributed and semantic rich architecture designed to support the interoperability and the integration of the different processes that characterize a DE. In an EOA all the components interact together, crossing organizations' boundaries and forming a DE that connects different systems, and exchange information using common data representations, like XML and other standard formats. All the EOA services are deployed on a distributed, peer-to-peer platform and described by business and functional models, using Unified Modeling Language (UML), adding in this way semantic to the service description. The decentralized architecture defines a topology and a replication schema that depend on a set of collaborative peer nodes. A peer-to-peer network supports this topology and the data replication across the network is guaranteed by a Distributed Knowledge Base (DKB) that stores and retrieves contents in a smart way. The final picture is a peer-to-peer and service oriented architecture with high integration capabilities offered by the adoption of open standards where the gap between business abstraction and software implementation is bridged by the adoption of model driven methodologies.

[1]http://www.digital-ecosystems.org
[2]http://en.wikipedia.org/wiki/Folksonomy

B. Paper Contribution

The paper proposes a new trust model for DEs based on calculation of trust that leads agents to interact or not with each other. The model has the following innovative features:

- It provides trust evaluation on a variety of DE layers: ranging from users and data exchanged among users to services and nodes (at platform level) providing services to users.
- Trust values can be calculated in a variety of contexts by using user-defined tags (folksonomy) that best represent a DE configuration.
- It abstracts from specific reputation algorithms by providing necessary interfaces for plugging-in those on one's choice.
- The model allows fast bootstrapping of trust evaluation by importing existing social networks of trust from outside DE systems and by relying on certificates issued by trusted authorities.

The rest of the paper is structured as follows. Section II defines trust and reputation in the context of DEs. Section III gives an overview of related trust management approaches. Section IV describes in details the reputation-based trust model. Section V shows how trust is provided at different levels of DEs. Section VI presents the social institutional trust approach. Section VII concludes the paper.

II. Trust and Reputation in DEs

Before describing the trust model in details, we will define trust and reputation in the context of DEs as understood and modeled in our approach.

We define *trust* as the confidence an agent has that another agent will behave certainly in a given situation. Trust is thus bilateral and subjective and represents the opinion of an agent about another one. Other agents will most probably have different opinions about the same agent. *Reputation* is an expectation about an agent's behavior based on past actions. Reputation is thus multilateral and represents how much an agent in a certain network or system is trusted by other entities. This implies a common view of an agent's trustworthiness. Agents with good reputation are trusted by other agents which means that trust is built through reputation, as well as, reputation is built from bilateral trust relations between agents.

Trust and reputation values are used by agents to evaluate with whom to interact. The opinion an agent has about another agent, based on past experiences (outside or inside the system) is made available to other agents through recommendations. In a DE, agents are autonomous entities that decide freely which actions to perform and with whom to interact. Agents are free to ignore or weight the trust statements of other agents. Moreover, agents organize themselves in social networks and trust more the opinions of known peers than those of unknown. Agents are diverse and have their own opinions based on own experience and social network.

Figure 1 summarizes different trust and reputation approaches based on the above definitions in different locality contexts. The reputation of a peer could be computed and accepted globally by all peers in the system or could be relative to one domain. The trust an agent has in some entity (e.g. agent, service) can be computed by taking into consideration the opinions of all agents in the system weighted based on some criteria dependent on the agent; or trust can be computed locally by taking into consideration the reputation the entity has in the agent's social network. As shown in the figure, the latter approach becomes more suitable for DEs. Hence, the model we propose computes trust based on local values specific to the agent and its social network.

	Bilateral	**Multilateral**
	Trust ⟵ ⟶ *Reputation*	
Local	Agent and social network dependent *DE suitable*	Neighborhood dependent Local & domain specific Generally accepted in the domain
Global	Agent and system dependent	System dependent & Neighborhood independent Globally accepted value

Fig. 1. Trust and reputation approaches

We consider trust to be multidimensional. The trust relations between agents take place in a context that is modeled as a multidimensional space described by a set of parameters. Users have different levels of expertise in different domains, a Service Provider can provide N services with different levels of quality, services can be trustworthy or not at the same time based on the parameters chosen to evaluate them (e.g. response time, availability, accuracy of results).

III. Trust Management Approaches

Broadly speaking, there are two main approaches to trust management as researched in the context of agents. With the first approach, agents use trust models to reason about the reliability or honesty of their counterparts. With the second approach, agents calculate the amount of trust they can place in their interaction partners, and the likelihood for an agent to be selected as an interaction partner depends on the calculated trust. Either of the models aims at guiding agents to decide on how, when and with whom to interact.

We review trust management approaches in the context of reputation-based trust, certificate-based trust and social institutional trust. Those concepts, though aim at establishing trust relationships among agents/users, differ in underlying principles which inspired our model.

A. Reputation-based Trust

Reputation-based trust models initially require agents to gather some knowledge about their counterpart's characteristics. A presumption drawn from the agent's own experience, as in [4], [5], defines a model where trust in an agent is calculated based on its performance in past interactions. Information gathered from other agents as advocated in [6], [7] draws trust indirectly from recommendations provided by others. Since recommendations could be unreliable, agents must be

capable to reason about the recommendations gathered from other agents.

Most peer-to-peer reputation models tackle only one level of trust or two at most. They could be classified as person/agent-based (e.g. PeerTrust, PRIDE), resource-based (e.g. Epinions), and person/agent and resource-based (e.g. Xrep). A person/agent-based model has the focus on modeling the reputation of people or agents that act on behalf of people. Usually, implementations of this model calculate the reputation in a customized and personalized way, visible only to the people/agent that does the calculation. For example, an IT consultant can have good and bad reputations in different IT domains at the same time.

Resource-based systems are focused on the reputation of resources like products (documents, media-files, etc.) or services. The model can serve as a guide for people or agents to select resources, and the reputation is often represented by a value that should be easily understood by different users. For example, a service provider with good recommendations can speed the users' adoption of its services.

B. Credential-based Trust

Advances in communication and networking brought distributed systems and applications to forefront of academic and industrial research. Those systems and applications faced the dilemma of how to establish trust in identities and attributes they possess.

As a result of that need, efforts have been launched for establishing public-key infrastructures [8], [9] (see also http://www.europki.org) that attest users' identities and attributes in a secure and trustworthy way.

The purpose of a public-key infrastructure (PKI) is to manage keys and certificates. By managing keys and certificates through a PKI, an organization establishes and maintains a trustworthy networking environment. A PKI enables the use of encryption and digital signature services across a wide variety of applications.

The term credential has become widely used for expressing digital access rights in a distributed environment and management of credentials emerged as a key issue for establishing trust among distributed entities [10], [11], [12]. Two distinct approaches have been proposed defining how trust is propagated and managed among entities – hierarchically or web-of-trust, so leading to two distinct proposals called X.509 [8] and SPKI [9].

Either of them faces the problem of how to establish trusted interdomain relationships prior to validating certificates. It is mainly because different domains have different security policies for certificate management and issuance, and so most of the proposed models such as cross-certification (part of CMP standardization) [13] or bridge certificate authority [14] providing solution to the problem pay the price of high complexity of system maintenance.

Our model is inspired by the fact that current security models are concentrated, most of the time, on network security aspects excluding from analysis the social behavior of users when they deal with digital institutions.

C. Institutional Trust and Beyond

Part of the contribution of this paper is that it involves the already existing network of institutional trust [15] that serves as a bootstrapping mechanism to any reputation-based trust model, especially when direct trust relationships do not exist or are difficult to establish.

Inspired by [15], our model provides the initials of evolutionary trust as characterized in [16]. The evolution process is supported by use of *meta-data* certificates of organizations that reflect the constantly evolving relationships of those organizations with their partners.

A meta-data certificate is a digitally signed XML document that describes an institution's relations with other (social) institutions in an ontology language like OWL[3]. The certificate is institution-related (self-signed by the institution) and each institution decides on its own what institutional relations should be included in the certificate. Through this meta-data certificate, other partners will have a wider information of the institution's relations.

The important aspect here is that *social trust*[4] can leverage joining to an online environment by examining partners' institutional relations as specified in their meta-data certificates rather than only examining trust between entities already (known) in the network. In this way, when institutional trust relationships evolve over time, they will be reflected by the meta-data certificate and adequately facilitate future collaborations.

IV. The Reputation-based Trust Model

The model we propose provides trust at different dimensions ranging from trust values in users, data or knowledge, services and nodes and social inter-institutional trust.

Let us examine the basic settings of our reputation-based trust model for peer-to-peer communications. Every user or agent keeps lists of opinions about other users, data, services and nodes, as shown in Figure 2. These values are made available to trusted agents and updated after every interaction or transaction. The model assumes the existence of the Distributed Knowledge Base (DKB) which allows searching and updating these lists. The lists are made available to contact users and interfaces allow making specific queries on the data. When computing the trust value for an unknown entity (user, data, service or node), agents ask their contacts for opinions who can further forward the query to their contacts. Since agents take into considerations the opinions of their contacts in computing trust values, each agent additionally keeps a list of opinions about the capacity of a contact to provide reliable recommendations.

[3] http://www.w3.org/2004/OWL

[4] In public or private organizations, government bodies or institutions.

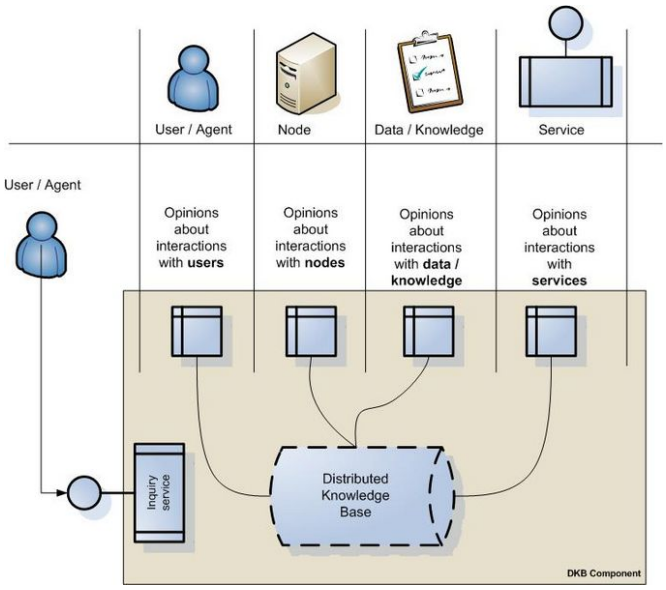

Fig. 2. List of opinions

A. Contact Lists

We will base our trust model on the fact that agents in a DE are *social* actors which dynamically create *social networks*. The social networks are created based on mutual trust relations between agents. The social networks are represented through contacts lists maintained by each agent. The contacts can be known either from the DE or from outside. By adding a contact to the list, a user states to trust that contact. A contact is added only with his agreement and the contact lists are symmetric (i.e. if A is a contact of B, then B is a contact of A too).

Different levels of trust are assigned to each contact. These values represent the trust a user has to receive accurate recommendation from specific contacts. In Figure 3, A has a list of opinions about the capacity of her contacts to provide good recommendations (in general or in certain contexts).

A	C_1	*Construction*	0.3
A	C_2	*Construction*	0.7
A	C_3	*Automotive*	0.6
A	C_4	*Automotive*	0.4
A	..	....	...
A	..	....	...
A	C_{n-1}	*Energy*	0.2
A	C_n	*Energy*	0.8

A trust (value 0.4) C_4 about **recommendation** (Automotive context)

Fig. 3. Trust in the contacts' recommendations

The relations between agents are complex and involve both cooperation and competition. Because of that we chose to make the list of contacts private and only accessible to direct contacts.

In our model, we consider trust to be a bilateral relation between two agents which is primary established on personal experiences. For unknown users, trust in providing reliable services is computed based on the social network. For example, in order for A to compute a trust value for an agent B with whom A has not interacted before, A uses the reputation B has among the agents belonging to her social network. The trust value that A computes for B is specific both to A and to her social network. A weights the opinions received from her contacts based on her own trust in the capacity of her contacts to provide good recommendations.

B. List of Opinions

Each agent keeps on his private space a history of (recent) experiences (interactions, transactions) with other agents, services, nodes, and data. Based on these, each agent computes an opinion which is added to the list of opinions kept private by the agent and made available only to contacts. Opinions are 4-tuples composed by subject, object, keyword and value as shown in Figure 4.

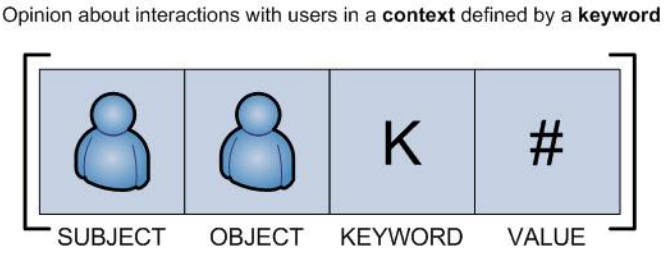

Fig. 4. Opinion Data Model

The subject is an entity (user) that gives an opinion, the object is an entity target of the opinion, the keyword is a string field used for identifying the context, and the value field is the trust value rating. Keywords can be a list of predefined user-specific tags (folksonomy) from which the subject can choose or simply a free text given by the subject. Users known from outside the system, with whom no inside contact has been made, are also added to the list of opinions. Typical peer-to-peer reputation systems do not allow this feature, but instead rely only on transaction reporting. By using this approach, the model allows building trust relationships more easily, and users perceive this relationships as stronger because of the personal contact.

Since users interact in different contexts and the expertise of a user could be greater in one domain than another, we distinguish between opinions in different contexts. Each agent has a list of opinions about other agents' expertise in different contexts. The agents in the list of opinions are not necessarily contacts as well. The list of opinions simply represents an agent's view about the capacity of other agents to provide good services and is based on personal experiences.

Trust ratings will be represented as probabilistic values from 0 to 1. Zero means no trust, and 1 mean complete trust. Users compute and assign trust values in an autonomous and independent way. Each user is free to choose the criteria and algorithm for computing these values.

For example, if the value is computed based on transaction experience, criteria to take into account could be: level of

satisfaction with each transaction, size of transactions (fraud on a $100 transaction counts more than on a $1 transaction), number of transactions, and time at which transactions occurred (recent transactions weight more). If transaction experiences in a given context T are recorded as (Entity ID, Transaction ID, data/time, size, context, rating), for a history of transactions of user A with user B:

$$(B, Tr_1, t_1, s_1, T, r_1)$$
$$(B, Tr_2, t_2, s_2, T, r_2)$$
$$...$$
$$(B, Tr_n, t_n, s_n, T, r_n)$$

the following equation could be used for computing A's opinion about B in context T weighted by transaction sizes s_i and a time function $f(t_i)$:

$$O(A, B, T) = \frac{\sum_{i=1}^{n} f(t_i) \cdot s_i \cdot r_i}{\sum_{i=1}^{n} f(t_i) \cdot s_i}$$

where $f(t_i)$ is a weighted function that gives more importance to recent transactions.

C. Trust Value Computation

When agent A wants to interact with agent B with whom has no previous experience, A can compute a local trust value for B in the following way: A asks her contacts the opinions about B in context T, and based on their recommendations, weighted by different factors, computes a trust value for B. Based on this value, A decides whether to interact with B or not.

In case a contact does not know B, the contact can ask its contacts. In this case, opinions are propagated by using the trust value as a multiplicative factor. The depth of the tree can vary and certain thresholds can be set for propagating trust values. Figure 5 shows the propagation scenario.

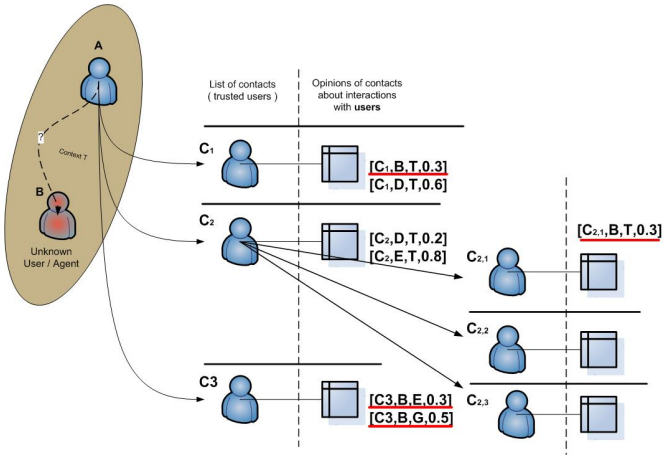

Fig. 5. Propagation of trust values (recommendations)

A will distinguish between values coming directly from a contact and values propagated through a chain of trust of a specific length. For example, [C1,2, B, T, 0.4] is an opinion from contact C1 with chain length 2. Opinions with a certain chain length will be weighted differently when averaging trust values.

Our model allows for plugging-in any reputation algorithm such as those in [17], [7], [18]. The basis of each reputation algorithm is defined by taking as parameter the querying peer A, the unknown peer B, the context T, and the chain length l. Thus, an application will retrieve from the DKB a graph G of peers and their trust values. This graph will be used by a reputation algorithm to compute a trust value for the unknown peer B. We note that this value is particular to the peer A making the inquiry and depends on A's social network and its trust in the other peers.

The two functions below show the possible functional interfaces necessary for a reputation algorithm to be plugged-in.

Graph getGraph(PeerId A, PeerId B, Context T, int l)

float computeTrust(Graph G, PeerId A, PeerId B, float thresh)

The graph representation may vary from one algorithm to another one depending on particular implementations.

The model also allows different peers to use different algorithms (and implementations) which best suit their needs or even to keep private their algorithms.

D. Updating Trust Values

After each interaction, the two lists need to be updated. Let's assume A interacted with B in context T and rated the transaction with value O(A, B, T). The opinion of A about B in context T is updated based on some algorithm which takes into consideration several factors such as number of transactions, time, and size of transaction. The user inserts manually the level of satisfaction after a transaction, but the system takes care of updating the trust value automatically. Based on value O(A, B, T), A can judge the recommendations provided by contacts with different values O(A, Ci, Recommendation) and update the trust she has in her contacts. Once A rates the transaction, the system can automatically update the trust level of contacts.

E. Generalization

The opinion data model described above can be generalized taking into consideration that the space underlying the context keywords can be a more generic context space with one ore more dimensions. A hierarchical classification of entities (products, processes, documents, human groups) of interest of an enterprise, organization or administration is an example of multidimensional context space that can be used for defining opinions. Figure 6 shows the essence of the generalization scheme.

The definition of the context space depends from the modeling needs and from the granularity expected. A typical example is the context of industrial products, which can be modeled as a hierarchical three-dimensional space with Category, Brand, Product:

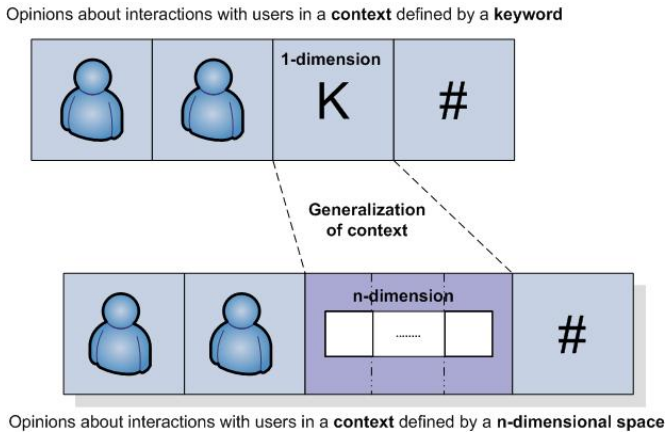

Fig. 6. Generalized Opinion Data Model

Category	Brand	Product
BabyCare	BabySan	N345
BabyCare	BabySan	Q579
BabyCare	BabySan	B2080
FirstAid	MyPatch	P250
FirstAid	MyPatch	P500
FirstAid	NeoAid	A350
DentalCare	BestTooth	T2000
DentalCare	BestTooth	T3000

In this example, each opinion has a triplet to identify the product context. The triplet's entries have dependencies between them: Product depends on Brand and Brand depends on Category. With this hierarchical data model we are able to compute the trust value grouping, for example, all the opinions under the same Brand. In this case, the opinion is reported using a list of the form: *[Subject S, Object O, Category C, Brand B, Product P, Value V]*.

Example: 3-dimensional context trust

[Andrea, Paolo, BabyCare, BabySan, N345, 0.5]

[Andrea, Paolo, FirstAid, MyPatch, P250, 0.4]

[Andrea, Paolo, FirstAid, NeoAid, A350, 0.3]

[Andrea, Paolo, DentalCare, BestTooth, T2000, 0.7]

How much does Andrea trust Paolo about providing FirstAid products?

In this case, the computing of trust values will be extended as follows:

Graph getGraph(PeerId A, PeerId B, Category C, Brand B, Product P, int l)

Double computeTrust(Graph G, PeerId A, PeerId B, float thresh)

V. Trust at Different Levels

Sometimes it could be useful to compute trust based on the relations between users and the components of a DE at different levels: front end objects, support services or low level infrastructure components. For example, a user interacts with different peers, accesses contents and uses services defining in this way many relations. The trust mechanism can group opinions of users about other entities following an aggregation path derived by the relations between users and the other entities.

For example, if there are relations between users and nodes in a DE then we can aggregate opinions of all users related to a specific node to form an *indirect opinion* of that node about another node the users have opinions about.

Figure 7 shows an example of a node opinion (recommendation) for another (unknown) node based on users' opinions.

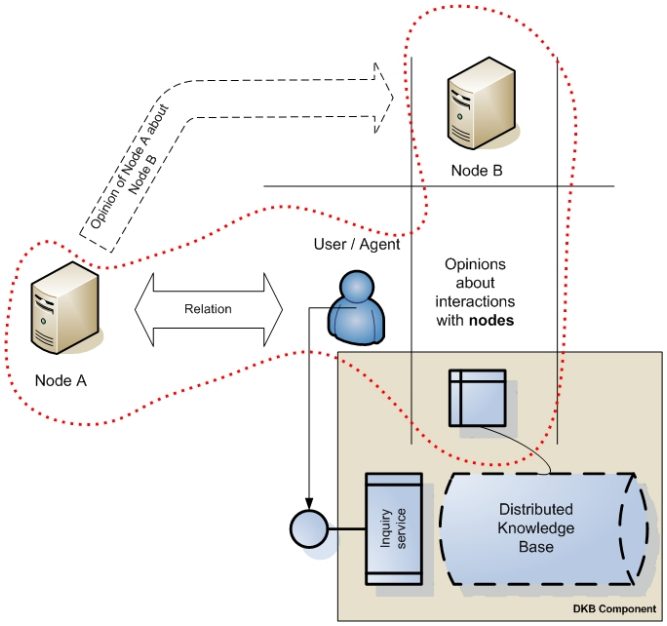

Fig. 7. Indirect opinions

Node A has an indirect opinion about node B. The opinion is derived from the relationship between node A and a specific User, which in turn has its own opinion about his interactions with node B. The relationship between node A and the User could be, for example, defined by the fact that the User has registered to a DE with node A.

VI. Bootstrapping Social Networks of Institutional Trust

This section provides the motivations and foundations of our social institutional trust model. Particularly, we show how institutional trust and socioeconomic trust (knowledge) bootstrap the reputation-based trust model.

A prerequisite to the trust model is an identity management model that allows entities identification in order to keep track of their actions in a system. The trust model we present in this paper relies on an identity management mechanism that scales to distributed systems, which underlies the nature of DEs.

We adopted the model in [19] for providing user authentication and identification. According to this model, each Service Provider (SP) (part of a peer-to-peer DE network) has a trusted Credential Provider (CP) to which users are forwarded for authentication (by using single sign-on[5]). Since

[5]https://opensso.dev.java.net

we are in a peer-to-peer setting, each CP establishes trust relationships with other CPs on a hierarchical or web-of-trust basis depending on its own policy. Likely, each CP has its format and standard for credentials representing identity information. Furthermore, there exists a number of standard solutions for identity representations but not all of them are compatible. Based on these relations, a user not known in a DE can get authenticated based on a translation of credentials from an external CP to the one trusted by the SP. The identity framework in [19] provides an easy-to-integrate solution to an existing service management system by requiring the adoption of the SAML standard [20].

A. Initial Trust Value Setup

When a user accesses a web site for registering to the system, the user gets forwarded to the trusted CP where it provides registration information such as name, organization, phone, age, e-mail address etc. The user can also provide any certificates obtained from certification authorities (CAs) outside the system, for re-use of already certified user information. If these CAs are in the list of trusted CAs of that CP with trust values assigned for each of CAs (possibly in different contexts), the user will be assigned from the part of the CP a trust value greater than 0. The system uses values from 0 to 1, where 0 means no trust and 1 complete trust. By default, each registered user starts with 0 trust value if it does not present any trusted certificates. A CP can assign initial trust values in different contexts depending on what information the user supplies. The CP will be considered always trustworthy, and for this reason it is assumed that users add their CPs to the contacts list with trust value 1.

There are two main reasons for assigning initial reputation values based on certificates issued by reliable (external) organizations: agents which have reliable credentials from trusted CAs outside the system are accepted in the system not with low-level trust but with a higher one, making them trustworthy for transactions. On the other side, newcomers with a good initial value are easy to be discovered (through the CPs) by other agents wishing to establish (business) communications even though they are not yet known as reliable agents.

If a user is invited by another user of the system, the invited user gets assigned a trust value manually by the inviting user.

B. Social Network of Trust

We consider recommendations of an institution for other institutions an important aspect in our model. Each institution could serve as a reference point for other institutions or authorities when dealing with trust decisions. To form the ”spine” of inter-organizational relationships we need to represent their references and recommendations in our model.

We do that by allowing an institution to rate and recommend other known institutions with values based on its own experience and business relations. Since the model between agents must be intuitive and flexible, we apply the same reasoning when an institution models its relationships, that is, by using multidimensional trust values.

In this way, an institution can express opinions using the same dimensional model as for user recommendations (possibly with different taxonomy) and, therefore, any reputation algorithm used for computing user recommendations can also be used for computing inter-institutional recommendations.

To form the spine of institutional networked relations, we create a special database in the Distributed Knowledge Base (DKB) network that lists those organizations willing to serve as institutional references.

Each institution has a meta-data certificate on a network end-point where the institution stores its recommendations and references with other institutions. When an institution wishes to register as a (trusted) reference point in DKB, it supplies the location pointing to its meta data certificate together with a description about the institution and its social status.

Thus, the more institutions registered, richer the database becomes and, hence, more accurate recommendations are formed.

In the model each certificate authority maintains a meta-data certificate and can be itself an institution serving as a pointer for institutional trust.

The meta-data certificate is digitally signed by the CA and is public accessible via a network connection to ensure its authenticity, validity and availability.

An important issue here is privacy of a company’s relations and recommendations. To this extend, we have two possibilities: first a company makes all its relations in the meta-data certificate public to anybody wishing to obtain such information or, second, a company provides an access control process to its document defining what types of organizations can access and under what conditions (like day time, domain restrictions, etc). Additionally, a sticky-policy model can be used to convey restrictions on data usage (like disclosure to third parties or removal if certain time expires) by end entities.

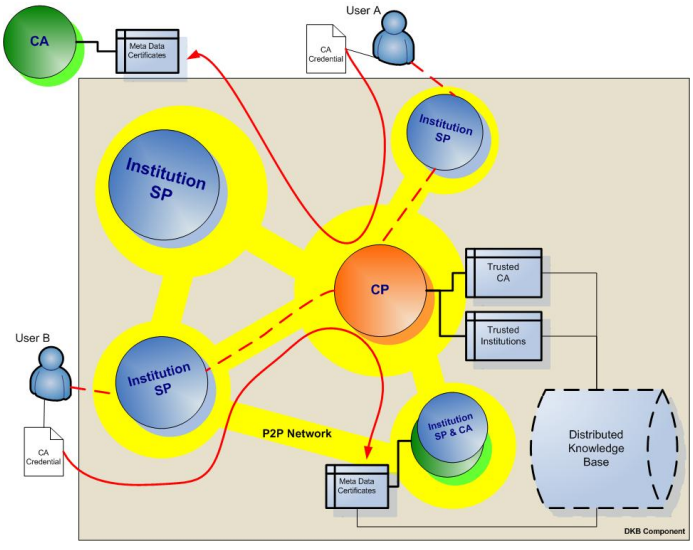

Fig. 8. Social institutional trust

To comply with the model, each CA includes in the end-user certificates, it issues, a link to its meta-data certificate.

This is done either via an extension field in a certificate (as adopted by X.509 standard) or by inserting an additional field in the certificate identifying the location of such (used by SPKI standard).

When a CP verifies a certificate issued by a CA, it can allocate the meta-data certificate and get additional information on CA's institutional network connections and analyze how trustworthy the CA is from a social network point of view.

Each CP has a list of trusted CAs and a list of social institutions that the CP considers reliable (trusted) for recommendations. To evaluate a certificate, a CP first checks if there is a direct trust link between the CA and the CP's list of trusted CAs. If such is found, then the CP computes a trust value[6] for that CA obtained from the recommendations available (retrieved) from DKB social network database and its list of trusted social institutions. The trusted institutions serve as initial trust value points in the newly formed graph of recommendations. Based on the computed trust value, the CP assigns a respective trust value to the user possessing the certificate issued by that CA. We assume that each CP registers in the DKB database those institutions which it considers reliable for recommendations including its list of social institutions.

If in the DKB social database there is not enough information about the CA, then CP looks at CA's meta-data certificate and extracts the institutions the CA claims it has relations with, along with the locations to their meta-data certificates. In this way, the CP may recognize some institutions it knows from its social knowledge (which currently are not reflected in its list of trusted social institutions nor in the DKB database) and can look for those institutions' recommendations.

Now, if the CP evaluates those recommendations positively, it rates the user having that certificate accordingly and registers the recognized institutions in its list of social institutions and in the DKB.

In this way, a DE evolves over time by using social knowledge and by including on-the-fly new institutions in its institution network database.

Figure 8 shows the social institutional trust model. There are two cases shown: when a user A belongs to an institution that is certified (its users have certificates) by uknown CA; and when a user B belongs to an institution that is certified by a known (trusted) CA.

VII. Conclusions and future work

We have proposed a trust framework for DEs which brings together the notion of social networks and peer-to-peer reputation systems. The model allows importing outside contacts in the system which allows fast bootstrapping and improves the accuracy of trust values.

The model provides multidimensional trust considering different levels of users experience with other users, data, services and nodes. The model can be generalized to any N-dimensional system requirements configuration without restricting it to a specific reputation algorithm.

Future work is divided in tho directions. One for experimental assessments on suitability of the model when using existing algorithms and implementations, such as [17], [7], over different dimensions and contexts. An open research question here is: how does using different contexts and factors improve the accuracy of trust values?

The second direction is to provide a practical implementation of the trust model that is suitable for majority of SMEs.

[6]A CP obtains a trust value computed over a reputation data set, as described in Section IV, based on a reputation algorithm and a threshold value.

Acknowledgment

Hristo Koshutanski is supported by 038978 EU-MarieCurie-EIF-iAccess fellowship. Mihaela Ion, Andrea Danzi and Luigi Telesca are supported by projects: 034744 EU-INFSO-IST ONE and 034824 EU-INFSO-IST OPAALS.

References

[1] P. Dini, *Digital Business Ecosystems*. European Commission, 2007, ch. A scientific Foundation for Digital Ecosystems, www.digital-ecosystems.org/book/de-book2007.html.

[2] J. Preece, *Online Communities: Designing Usability, Supporting Sociability*. John Wiley & Sons, Chichester, UK, 2000.

[3] P. Ferronato, "Architecture for digital ecosystems, beyond service oriented architecture," in *Proceedings of the 1st IEEE Conference on Digital EcoSystems and Technologies (DEST'07)*, 2007.

[4] M. Witkowski, A. Aritikis, and J. Pitt, "Experiments in building experiential trust in a society of objective-trust based agents," *Trust in Cyper-societies*, pp. 111–132, 2001.

[5] J. Sabater and C. Sierra, "REGRET: a reputation model for gregarious societies," in *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agents Systems*, 2002.

[6] A. Abdul-Rahman and S. Hailes, "Using recommendations for managing trust in distributed systems," in *Proceedings IEEE Malaysia International Conference on Communication*, 1997.

[7] D. Sepandar, T. M. Kamvar, and H. Garcia-Molina, "The EigenTrust algorithm for reputation management in p2p networks," in *Proceedings of the Twelfth International World Wide Web Conference*, 2003.

[8] X.509, "The directory: Public-key and attribute certificate frameworks," 2005, ITU-T Recommendation X.509:2005 | ISO/IEC 9594-8:2005.

[9] SPKI, "SPKI certificate theory," Internet Engineering Task Force (IETF), 1999, iETF RFC 2693.

[10] H. Rifà-Pous and J. Herrera-Joancomartí, "An interdomain PKI model based on trust lists," in *Proceedings of the 4th European PKI Workshop: Theory and Practice (EuroPKI 2007)*, ser. Lecture Notes in Computer Science, vol. 4582. Springer, 2007, pp. 49–64.

[11] J. Marchesini and S. Smith, "Modeling public key infrastructures in the real world," in *Proceedings of the 4th European PKI Workshop: Theory and Practice (EuroPKI 2005)*, ser. Lecture Notes in Computer Science, vol. 3545. Springer, 2005, pp. 118–134.

[12] K. Zeng, "Pseudonymous PKI for ubiquitous computing," in *Proceedings of the 4th European PKI Workshop: Theory and Practice (EuroPKI 2006)*, ser. Lecture Notes in Computer Science, vol. 4043. Springer, 2006, pp. 207–222.

[13] C. Adams, S. Farrell, T. Kause, and T. Mononen, "Internet X.509 public key infrastructure certificate management protocol (CMP)," IETF RFC 4210, 2005, www.ietf.org/rfc/rfc4210.txt.

[14] D. Blanchard, "I-CIDM bridge to bridge interoperations," in *5th Annual PKI R&D Workshop "Making PKI Easy to Use"*, April 2006.

[15] P. A. Pavlou, Y.-H. Tan, and D. Gefen, "The transitional role of institutional trust in online interorganizational relationships," in *Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03)*. IEEE Press, January 2003.

[16] L. Telesca and H. Koshutanski, *Digital Business Ecosystems*. European Commission, 2007, ch. A Trusted Negotiation Environment for Digital Ecosystems, www.digital-ecosystems.org/book/de-book2007.html.

[17] P. Massa and P. Avesani, "Trust metrics on controversial users: balancing between tyranny of the majority and echo chambers," *International Journal on Semantic Web and Information Systems (IJSWIS)*, 2007.

[18] J. de la Rosa, "Opinion-based filtering through trust," in *EXYSTENCE Topical Workshop on Trust-Based Network and Robustness in Organisations*, May 2006.

[19] H. Koshutanski, M. Ion, and L. Telesca, "A distributed identity management model for digital ecosystems," in *Proceedings of International Conference on Emerging Security Information, Systems and Technologies (SECURWARE'07)*. Valencia, Spain: IEEE press, October 2007.

[20] SAML, "Security Assertion Markup Language (SAML)," OASIS Security Services TC, 2005, www.oasis-open.org/committees/security.