OPAALS PROJECT

Contract n° IST-034824

WP 4: Distributed Accountability, Identity, and Trust

Deliverable 4.3 - Trust Model for the DE

Project funded by the European Community under the “Information Society Technology” Programme

Contract Number: 034824

Project Acronym: OPAALS

Title: Open Philosophies for Associative Autopoietic Digital Ecosystems

Deliverable N°: D4.3

Due dates: 9/2007

Delivery Date: 12/2007

Short Description:

This deliverable proposes a new trust model for DE. The deliverable first reviews both the notion of trust in different disciplines and different trust management strategies. We take a multidisciplinary approach and combine ideas from computer science with ideas from sociology, economy, and biology. We take, however, mainly a computer science approach for providing an evolutionary trust model for the DE. The deliverable also includes a survey of current peer-to-peer reputation systems and approaches.

We then propose a new trust model for DE that has four main components: distributed identity management, peer-to-peer reputation, trusted rating agencies, and learning, which address different security, social and evolutionary aspects of the DE.

We conclude with future directions and suggestions for evaluating and validating the model.

Authors: Mihaela Ion, Luigi Telesca (CREATE-NET), Jimmy McGibney, Dmitri Botvich (WIT)

Partners contributed: WIT

Made available to: Public

VERSIONING		
VERSION	DATE	AUTHOR, ORGANISATION
1.0	**30/11/2007**	**MIHAELA ION, LUIGI TELESCA (CN), JIMMY MCGIBNEY, DMITRI BOTVICH (WIT)**

Quality check:

1st Internal Reviewer : Brendan Jennings, WIT

2nd Internal Reviewer: Antonella Passani, T6

Table of Contents

1 Introduction

Digital Ecosystems (DE) have emerged as the new multidisciplinary paradigm for the evolution of Networked Organisations. Different definitions of Digital Ecosystems are available. This is mostly due to the complexity of the concept and to the disciplinary domain of reference. Anyway, although we are simplifying the concept, we can define the Digital Ecosystems as an open decentralised information infrastructure where different networked agents, such as enterprises (especially SMEs), intermediate actors, public bodies and end users, cooperate and compete enabling the creation of new complex structures. In Digital Ecosystems, the actors, their products and services can be seen like different organisms and species that are able to evolve dynamically. Those organisms or species through incessant interactions, alliances, adaptation and composition could experience new evolutionary paths in order to better adapt and survive to changing market conditions. In this context organisms are able to migrate from one ecosystem to another, can experience a mutation or even die because of adverse conditions and digital selection.

The dynamic nature of Digital Ecosystems poses many challenges for identity management and reputation-based trust systems. Existing technologies and models do not fully integrate the needs of SMEs and are not suitable for DEs. We propose a new multidisciplinary model for DEs based on current security technologies and reputation mechanisms. Though we take a computer science approach, in order to solve concrete issues related to the open distributed platform of the DE, we also rely on concepts and methods from ecology, economy and sociology.

We introduce a new Distributed Identity Management model which constitutes the basis of our trust model. The Identity Management model enables creating a reputation framework by providing ways for securely identifying entities. The reputation framework has two main components: a Peer-to-peer Reputation model and a decentralised Trusted Rating Agencies model. DEs evolve in time in order to respond to changing conditions. To better accommodate the nature of DEs, the model uses evolutionary trust, a novel research concept that reflects the constantly evolving social relations.

2 Trust

In the real world, as in the digital one, trust is one of the key factors that permit the growth of social and business networks. As identified by Jenny Preece [58] only when there is trust between people the relations can flourish, without trust there is no cooperation and therefore no society. Trust in online communities is even more important, than in real world, due to the difficulty of a physical interaction and to the lack of unintentional emotional states that could be expressed through body language and implicit behaviour. Trust is therefore central and is also a basic requirement for a functioning Digital Ecosystem (DE). The concept of trust has been researched in different disciplines and due to its multidisciplinary dimensions it has been open to multiple interpretations. In this section, we introduce different aspects of the notion and models of trust and we try to understand how trust can be measured and used in a computational manner to make Digital Ecosystems work.

2.1 Background

When people interact with each other in the real world and provide and consume services, well-evolved social and commercial standards and structures help to provide assurance of orderly behaviour. Our senses are well tuned to the subtleties of real personal contact. Furthermore, in dealing with service providers like banks or shops, physical structures and contact help to convince us that we are really dealing with the service provider (and not an impostor), that the service provider appears to be backed by some assets and will still be there into the future, that the transaction is genuine, that the communication is confidential, and so on. Previous experience is also a major factor – for example, a bank is more likely to extend a loan to a customer with a reliable track record. Previous good experience with a service provider assures a consumer of the quality of future transactions. Laws of the land provide penalties that reduce the incentive to behave dishonestly.

Reference is also made to third parties where appropriate. Credit card and other financial transactions need to be correctly authenticated and authorised. Various respected bodies issue credentials to people to help with verification of identification, nationality, permission to drive a car, access to restricted locations, and so on. Service providers may also be certified as to professional competence, adherence to health standards, and so on. More fuzzy personal recommendations and reviews are also of value.

The digital world provides opportunities for similar (and in some ways more sophisticated) social and commercial interaction. A great benefit of the Internet is its openness and lack of centralised control. Anyone can provide a service with the minimum of fuss and invite others to make use of it. As well as in the strictly Internet world, mobile telecommunications networks are facilitating more open service provision and consumption. The increasing proliferation of wireless and ad-hoc networks using unlicensed radio spectrum is serving to further loosen control.

Anyway this digital world presents a wide variety of risks. Anyone can set up a web site that looks just like your bank's site and use various tricks to get you there; online communications can be eavesdropped upon, or even modified; someone else may impersonate you; incorrect or misleading information may be provided. Legal protection is made difficult by the inter-jurisdictional nature of these networks. A major factor is that we can no longer rely on physical structures, social skills or intuition to provide assurance of security, and thus there is a much greater need for reference to third parties for identification of entities and verification of credentials. All those factors can

increase the risk of online interactions and as a result limit people's participation in online communities.

Trust is therefore a basic requirement for these interactions to succeed over time. A successful online communication or transaction requires that the parties involved sufficiently trust each other and the infrastructure that connects them. Consider for example the case where I use a web browser to navigate to an online retailer's site to purchase a product or service. For this to work, I need to trust several entities: that my computer hardware and software is acting in my interest (trojaned web browser software could direct me to a bogus location); that the website is actually that of the vendor; that the vendor and any other parties collaborating with the vendor (e.g. for payments) will act responsibly with my personal information; that the vendor will deliver the product or service as expected.

I may also need to trust, to some extent at least, other parts of the infrastructure. Where there is no strong means to authenticate the other party (e.g. an email correspondent or a non-secured website), I may need to trust that the domain name system (DNS) directs me to the correct IP address and that my Internet service provider and other infrastructure providers correctly route data packets.

The current approach to trust establishment and management on computer networks works in some situations, to some extent, but has significant weaknesses that limit its potential, especially in enabling rich peer-to-peer interactions and transactions. In those environments trust and identity can not be centrally controlled and therefore current approaches fail.

Note that the value of building trust is not limited to commercial interaction. Even if we consider knowledge that is shared for free and without restriction, there are still threats, which can be reduced by having measures of trust, such as:

- The knowledge provided could be deliberately false. For example, free software could contain a Trojan horse or other malicious code.
- The knowledge provided could be erroneous or subject to misinterpretation, due to limitations of its creator or editor.
- The consumer of the knowledge could waste a lot of time on a facet of low value. It takes time to read a document. Software takes time to install. There may also be a learning curve, meaning that significant time and energy needs to be invested in its adoption, making it painful to rollback if it turns out not to be fit for its intended purpose.
- Some knowledge may be undesirable and unwanted, especially if the consumer does not specifically solicit or request it. Spam is an example of bad 'knowledge'.

2.2 The meaning(s) of trust

The social concept of trust is very complex and sophisticated, perhaps deceptively so. Trust is closely related to many other social concepts such as belief, confidence, dependence, risk, motivation, intention, competence and reliability [27]. It is also interwoven with the areas of accountability [28] and identity [29].

Trust management has become a fairly well-studied area in recent years, and several recent surveys summarise the state of the art [30][31][32][33]. Much of the work on trust in the computer science domain attempts to provide computational measures of trust so that it can be used in making decisions related to electronic transactions, in a sense mimicking people's well-evolved forms of social interaction.

One of the difficulties of modelling trust computationally is that social trust is based on quite an intuitive and personalised subjective assessment. As trust is quite an overloaded term, most

attempts to model trust start by defining what is meant by trust (at least for the purposes of that particular model). Thus there are currently several different definitions and interpretations of what trust means and how it can be used. Here we describe some contributions and definitions about trust coming from Social Sciences, Economics and Biology. This will help us in order to better understand the complexity of the topic and to define our multidisciplinary approach toward trust in DE research.

2.2.1 Definitions

Trust refers to a unidirectional relationship between a **trustor** and a **trustee**. A trustor is an autonomous entity that is capable of making an assessment. A trustee can be anything.

A frequently cited definition of trust is by Gambetta as [34]:

> *"a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action and in a context in which it affects his own action"*

This definition formulates trust as:

(i) a probabilistic measure; i.e. implying that trust can be modelled as a single value between zero (complete distrust) and one (complete trust).
(ii) defined by the subject; each *trustor* may have a different view of the same *trustee*
(iii) relating to a particular action – i.e. a particular service offered by the trustor; you would trust your bank more for financial advise and your doctor more for medical advice.
(iv) an *a priori* measure; trust relates to incomplete information and is thus an estimate.
(v) relating to context; trust depends on the viewpoint of the trustor on how it might affect his or her action

Jøsang et al. in [30] adapt the work of McKnight and Chervany [35] to define **decision trust** as:

> *"the extent to which one party is willing to depend on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible"*

In some cases, "to trust" is taken to mean making the decision itself. "I trust you" means that I believe that you will act in a particular way.

Trustworthiness is a term closely related to trust, and sometimes used interchangeably with it. Solhang et al. [36] have made a useful distinction between trust and trustworthiness, by accepting Gambetta's definition of trust as a *subjective* probability and defining trustworthiness as an *objective* value:

> *"the objective probability by which the trustee performs a given action on which the welfare of the trustor depends".*

Reputation is also related to trust, though differing views exist on precisely how they are linked. A common view is that reputation is one of the sources used by a trustor in making a trust assessment. Other sources of information for a trustor, besides reputation, typically include verifiable credentials of the trustee as well as any direct experience the trustor has had of the trustee. One simple definition of reputation is [37]:

> *"perception that an agent creates through past actions about its intentions and norms"*

For reputation to be meaningful (distinct from the trustor's own experience), it must be the collective perception of a community of observers that is somehow made available to the trustor.

2.2.2 Trust in social science, economics and organisational studies

The concept of trust in economics is always linked with the irrational aspects of human life and its contraposition with the classical theory of rationality *(homo economicus)*[59][60]. In J.S. Mills work the man (*homo economicus or sapiens*) is a rational and self interested agent or actor that acts with the objective to obtain the highest possible level of well-being, at minimal cost, considering all the opportunities and constrains that could affect him directly or indirectly. The rationality of the *"homo economicus"* is represented by its objective to maximise his utility function (well-being representation) considering all available opportunities. This means that first of all an agent cares only about his or her own material payoff and is indifferent about the payoffs of others; he or she is also aware of all the possible options. In this context trust represent a paradox [61].

Through a game theoretic approach Hollis [61] explain that being rational, while acting together with another actor, could lead to a situation where the overall utility is far less than hoped. Hollis affirms that *"Self-interest, even enlightened self-interest, turns out to be self-defeating. [...] The sum of [actor's] choices can be suboptimal for both of them. When they see an inferior outcome in prospect, they can try to head it off by agreeing to cooperate. But words are cheap talk since each keeps an agreement only when breaking it pays less, all things considered. So [agents] do what they promised only if they would have done it anyway. That is why, at this stage of the story, two ideally rational agents [...] have as yet no hope of reaching "The Triumph of Reason"".* Hollis also sustains that *"Social life depends on trust, especially on trust that promises will kept. […] the stronger the bond of trust, the more a society can progress; the more it progresses, the more rational its members become and hence the more instrumental in their dealings with one another; the more instrumental their relations, the less trustworthy they are. So the progress of reason erodes the bond which made it possible and which it continues to need. [...]".*

The only way to escape from this paradox is therefore to recognise that agents experience limits in defining and solving complex problems (Bounded Rationality)[62] and that some sort of altruistic behaviour is possible. Here risk is therefore central in the economic concept of trust and goes beyond the calculative self interest. Trust is therefore seen as a multidimensional concept where rational and social or ethical aspects could be taken in consideration. Luhmann [63] says that *"The truster sees in his own vulnerability the instrument whereby a trust relationship may be created*". In this phrase we can capture two main things. The first relates the strong relation between risk and vulnerability; the second considers the instrumental use of vulnerability as a medium to build a reliable network of trusted links. Trust is fundamental in organisations and is the main driver for a successful collaboration.

Cummings and Bromiley [64] with their Organisational and Trust Inventory (OTI) model study trust in the organisations with such multidimensional approach. The authors, in fact, define trust *"[...] as an individual's belief or a common belief among a group of individuals that another individual or a group, makes good faith efforts to behave in accordance with any commitment both explicit or implicit, is honest in whatever negotiations preceded such commitment and does not take advantage even when the opportunity is available".* They not only identify clearly the three main components of trust (the ability to fulfil commitments, the willingness to negotiate honestly and the intention to avoid taking any sort of excessive advantage from unforeseen opportunities at the expenses of the others), they also declare that trust is linked with the belief and is therefore distributed among three main dimensions (the affective state, the cognition and the intended behaviour). The result of the study reinforces the idea of trust as a multidimensional concept based on the combination of all the above mentioned elements and underlines the strategic and tactical dimension of trust in order to build a sustainable working environment.

The effective protection of complex systems, especially those that are made up of loosely coupled autonomous entities, requires the development of appropriate strategy. A strategy must of course

have an objective. In our case, the objective is to enhance collaboration between the organisms of the digital ecosystem (especially SMEs) while reducing the opportunities for corruption of the system caused by attacks of malevolent entities, possibly in collusion with each other.

These interactions between members of the ecosystem and those intent on undermining the community and the system could potentially be modelled as a game. Several authors have already taken a game theoretic approach to trust management (e.g. [50]). In this approach, trust models are based on providing incentives to members to participate actively in the system and reducing free riding. The benefit in participating in the system needs greater than some threshold to make useful contributions worthwhile.

2.2.3 Biological relationship to trust

At it has been said many times in literature, trust is at the basis of any human interaction and collaboration. Our goal is to create a multi-disciplinary trust model inspired from social sciences, biology and computer science. Examples of collaboration and trust can frequently be found in the biological ecosystems. Our model can get inspiration from the way relations between entities in biological ecosystems are formed and evolve in time. Though it is not always clear whether cooperation in biological ecosystems it is due to trust relations, it is, however, worth analysing these aspects as well.

Examples of cooperation for mutual benefit are evident in biology, at several different levels, from cells up to complete ecosystems.

At the cellular level, ‘quorum sensing’ is a biological phenomenon that is manifested through synchronised behaviour of cells. For our purposes, quorum sensing is interpreted as similar to voting: an entity updates its own state by using some sort of averaging procedure for node states in its neighbourhood. At the level of organisms, an example of mutual cooperation is in how ants find the best path between two points, and are able to adapt to changes in the environment to discover a new path.

It has been argued that cooperative behaviour is a product of biological evolution [43]. By Darwinian processes, signals that predicted what one individual was going to do, and techniques for responding to this, would have become mutually beneficial. Also, quick interpretation of the actions of familiar individuals would have been useful.

Three evolutionary explanations have been identified in the literature [43] for social cooperation between animals in natural ecosystems:

1. Individuals cooperate because they are closely related (e.g. parental care)
2. The surviving character is the property of many individuals: group survival. The outcome of the joint action of individuals could become a character. This applies for example to symbiotic partnership.
3. Two individuals who are not necessarily related mutually benefit from the cooperation and are more likely to survive.

Cooperation and trust relations in a natural ecosystem adapt to changing environment conditions or to the appearance of new members, and evolve in time. We find these characteristics to be important for digital ecosystems as well. We will borrow from these concepts and model adaptive and context-aware trust.

2.3 Trust Types in DBEs

In the Digital Business Ecosystem Project some work has been done in the study of trust for Digital Business Ecosystems (Deliverable 32.2). In particular, in the above mentioned deliverable, authors provide a taxonomy that is comprised of three major dimensions: trust types (X, Y and Z), building blocks of regulatory trust (privacy, consumer protection, e-signatures and security, and jurisdiction) and operational perspectives (DBE relationships, actors and software lifecycles).
The trust types were identified on the base of regulatory trust in the DBE environment, where the regulatory aspect was considered *"central to building trust relationships between partners"* (Deliverable 32.2). Several levels of trust can be distinguished in a DE:

- **Trust type X** refers to the *trust in the system*. The trust companies and users have in the technical architecture and services of the system determines them either to join or not the system.
- **Trust type Y** refers to the *expectations* established members have about *joining users*.
- **Trust type Z** refers to the *trust relationships between participants* of the system. A higher mutual trust results in a higher number of transactions and collaborations between participants.

The authors used a further breakdown on those values based on the DBE layers suggested by Nachira (2002) and on the base of type of business transaction business to business (B2B) and business to consumer (B2C). The layer identified by Nachira can be seen as follow:

- **Trust in services and in technological solutions** in terms confidence about platform security and reliability.
- **Trust in business activities** in terms of agreement towards practices and procedures for specific sectors and local contexts.
- **Trust in knowledge** in terms of symmetric vs. asymmetric access to information.

In the deliverable a table representing the classification of regulatory issues based on trust types has been provided and has been inserted below for completeness.

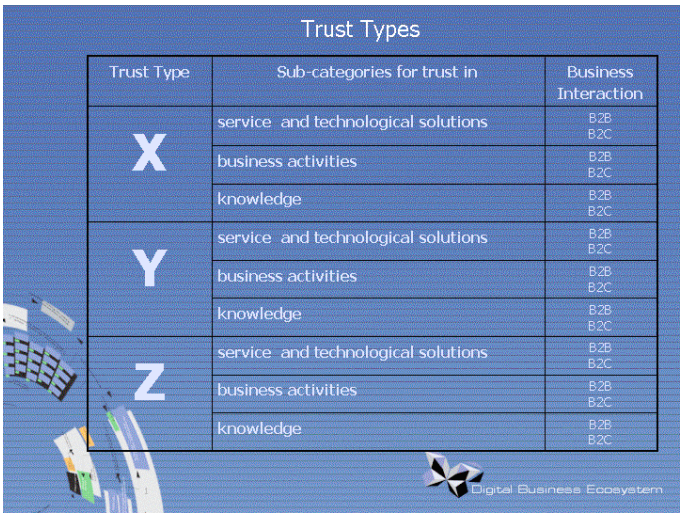

Figure 1: Trust Model Components

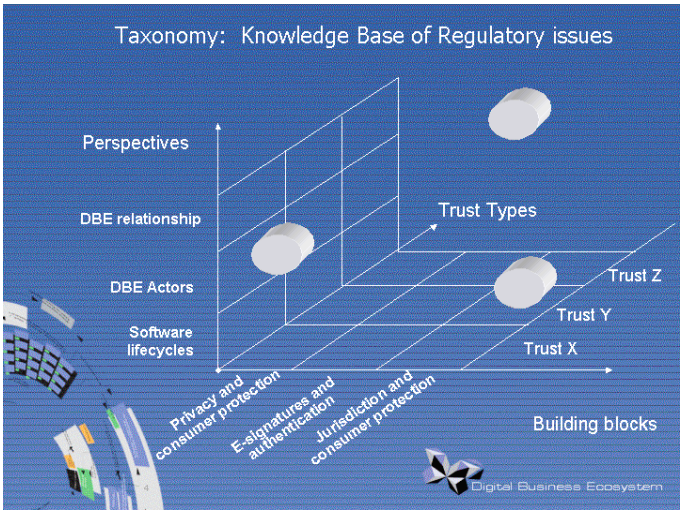

Figure 2: Trust Model Components

Those trust types values have been then portrayed in a three dimensional matrix with the building blocks of regulatory issues and the operational perspectives. The authors wanted to provide a clear framework to analyse the impact of any identified regulatory issue on any element of the classification adopted. In this way it would be easy to identify and analyse the issue and also to predict its influence on all the other elements.

2.4 Trust management

The trust management problem was first described in 1996 by Blaze et al. [38] as an approach to protecting open distributed systems. They specified trust management as based on the following principles, as implemented in their *PolicyMaker* system:

- Having a unified mechanism wherein policies, credentials and trust relationships are represented using a common language.
- Flexible specification of trust relationships.
- Local control; i.e. each entity makes independent trust-based decisions.
- Separation of mechanism from policy; how credentials are verified does not depend on the credentials themselves.

Two major categories of trust management approach have since then emerged, namely credential-based trust management and experience-based trust management. These are sometimes called hard trust and soft trust respectively, and it is of course possible to have hybrid systems that combine the two approaches [39].

2.4.1 Credential-based trust management

Properties of trusted entities are attested to by credentials that are issued by a third party that has some authority. Credentials can be verified with reference to the authority. In practice, credentials are normally digitally signed and can be verified cryptographically. The PolicyMaker system mentioned above is largely credential-based.

Credentials are often used in *trust negotiation*. This is the process of building mutual trust between entities that have no prior knowledge of each other. This is done by iteratively exchanging credentials between the entities. Each credential is used by the recipient to verify certain properties of the holder [40].

2.4.2 Evidence-based trust management

Following initialisation at some default level, trust in an entity is adjusted as experience is gained of that entity's behaviour. This may be direct experience, or third party experience of the entity in question (i.e. reputation information), that is reported to the trustor.

With evidence-based trust management, there may or may not be a reliance on centralised references. Such a reliance on centralised authorities does not sit well with rich peer to peer interaction in a digital ecosystem. In a truly peer to peer network, there is no way to store trust information centrally. In any case, it can be argued that trust by its nature is subjective and thus best managed by the entity doing the trusting. It is perhaps worth noting though that in practice, as yet, and with the exception of Pretty Good Privacy (PGP) for email, there are few widely used truly distributed trust systems. Several online marketplaces, social networks and review websites use reputation and ratings systems to give a measure of trust, but these mostly depend on some centralised storage and management.

Modelling this kind of trust management system is a non-trivial task, and there are many issues to consider. We can identify several requirements for effective evidence-based trust management strategies, particularly:

- *Strategy for exchanging trust information*
  Algorithms and protocols are required for the dissemination of evolving trust information between entities, in possibly large scale distributed environments, taking into consideration the impact on performance and scalability.
- *Strategy for updating trust*
  Algorithms are required for updating trust at a particular node on receipt of new information. These may take into account the level of trust in the referrer (trust transitivity), among other things, and may be based on averaging, thresholds, time decays, and so on.
- *Convergence*
  Just like in social systems, trust should in general be allowed to evolve gradually with time and experience. A distributed trust system should have stable dynamics. Responsiveness to threats needs to be offset against a desire for stability.
- *Incentive to participate*
  Strategies are needed that allow each entity to act in its own self-interest while contributing to the common good.
- *Different trust decision per service*
  Each distinct service will have its own specific trust requirements. It should be possible for entities to earn trust based on usage of low-risk services, and possibly eventually use this increased trust level to carry out more sensitive actions.
- *Resistance to "ballot stuffing" attack by colluding malicious entities*
  There is a risk of colluding malicious entities attempting to artificially raise each other's trust or artificially lowering the trust of good entities by supplying false data to corrupt the system.
- *Default trust and identity changes*
  Trust strategies need to specify how much trust, if any, is assigned to previously unknown entities. Subsequent behaviour may cause this trust level to either rise or fall. The motivation for this approach is to distinguish known bad entities from the simply unknown. In many social systems, there is default trust. Unknown people are trusted more, for certain activities anyway, than those who are known to be untrustworthy. A balance needs to be struck between having a default trust level of zero and having a sufficiently high initial assignment of trust that some services are accessible by new arrivals right away. A risk with having a non-zero default trust

level is the *Sybil attack* [29]. Having a default trust level helps with trust bootstrapping. If we were to start off with no trust between any pairs of entities, then no services would be allowed and there would be little opportunity for the new entities to get any information on which to base decisions.

- *Normalisation*
  Some entities have a bias towards positive referrals while others have a negative bias. It is desirable to have some kind of normalisation strategy that removes most of the effects of such a bias. Kamvar et al. [41] have proposed some useful normalisation strategies.
- *Time decay*
  Usually, the more recently experience with another party is, the more confidence we have in our trust assessment of them. Thus, a distributed trust system should apply some mechanism to allow trust values to decay over time – i.e. so that events that happened a long time ago have less weight than those that happened fairly recently.

2.5 Trust metrics and representations

There is no consensus on how trust measures are modelled. Adoption of Gambetta's definition of trust [30] as a probabilistic measure indicating confidence in a certain type of behaviour usually implies that trust is to be modelled as a single number in the (0/1) range. Note that Castelfranchi has cautioned against a non-reductionist approach of trust [42].

If trust is taken to mean the decision itself, this lends itself to a binary model meaning that an entity is either trusted or untrusted.

Table 1, adapted from that maintained by J.-M. Seigneur at http://www.trustcomp.org/, summarises the diverse range of trust representations that have been proposed in the literature. The first column "Format" indicates in summary the way that trust is represented, the second column provides more detail on this, and the third column gives an example. Trust is thus variously represented using integers, real numbers, logical decision outcomes, fuzzy categories, or vectors. Vector representation of trust allows for encapsulation of a rich set of measures.

Format	Representation	Examples
Fuzzy	undefined/marginal/complete, don't know/untrustworthy/marginal/full	PGP's direct trust
Fuzzy	very trustworthy/trustworthy/ untrustworthy/very untrustworthy	Ruth/Xu/Bhargava/Regnier's RT0
General	Lattice	Moreton/Twigg
General	Binary (yes, no) + poss. supplementary info	Trust management (Blaze/Feigenbaum's PolicyMaker and KeyNot
Integer	Virtual currency {0..infinity}	Trustos in ubiquitous computing (http://www.trustos.com); maybe the Nuglets in mobile ad-hoc networks
Integer	{-infinity..infinity}	Slashdot.org's Karma; Ebay's feedback rating
Integer	1-10	FOAF trust module - http://trust.mindswap.org/trustOnt.shtml
Integer	{0..infinity}	Chimaera (Tarah, Huitema); Free Haven (www.freehaven.net, Dingledine) direct and meta trust; Li Gong system and direct trust; PGP's parameters (number of agents); Advogato's trust metric
Integer	{1..infinity}	Ueli Maurer's deterministic trust
Integer	{-1..4}	Abdul-Rahman/Hailes; Poblano (www.jxta.org/docs/trust.pdf)
Logical	Binary (true, false)	Cryptographic logics (Burrows/Abadi/Needham' BAN, Gong/Needham/Yahalom's GNY, Syverson/van Oorschot's SvO); Freenet (www.freenetproject.org); Lars Rasmusson/Sverker Jansso P. Venkat Rangan.; SPKI
Logical	trust(trustor,trustee,action,level) where level is in {-100;100}	Grandison/Sloman's SULTAN
Logical	trustBL(trustor,service,trustee,,depth,blacklist) where depth is in {0..infinity} + {*} (* for)	Giorgini/Massaci/Mylopoulos/Zannone
Logical	formulas	Demolombe/Jones logics; Walt Teh-Ming Yao's Fidelis
Matrix	Table	Yao-Hua Tan
Real	[0;1]	Ueli Maurer's probabilistic trust; Neurogrid (www.neurogrid.net, Samuel Joseph); Sierra's (sierra.openprivacy.org, OpenPrivacy) System trust; Castelfranchi/Falcone/Pezzulo's trustfulness
Real	[0;1[	BBK (Beth, Borchedring, Klein) direct trust
Real	]0;1[	IST (Jamil, Sadri, Shiri)
Real	[-1;1]	Marsh (http://citeseer.ist.psu.edu/marsh94formalising.html); Sierra (sierra.openprivacy.org, OpenPrivacy) interpersonal and self trust
Vector	Tuple (each component in [0;1])	Josang's opinions (belief/disbelief/uncertainty/atomicity); Narenda Shankar/William A. Arbaugh; Golbeck/Hendler/Parsia (minimum, maximum, and weighted average trust); SECURE (support, inconclusive, contradict)
Vector	Emotional Bank Account (EBA) Tuple of 3 fields in a given virtual currency (amount of money paid by the trustee in terms of actions; money lost due to these actions; some money, which is not yet known to be gained or lost)	Trustee's EBA(100,12,4): the trustor has gained 100 trustos from th trustee, lost 12 trustos and 4 trustos are still not processed (http://www.cs.tcd.ie/publications/tech-reports/reports.04/TCD-CS 2004-37.pdf) ; the term Emotional Bank Account is from Stephen Covey's 7 habits of highly effective people

Table 1: Some trust representations from the literature (adapted from http://www.trustcomp.org/)

2.6 Summary and conclusion

OPAALS is a multi-disciplinary project that attempts to build a foundation for digital ecosystems based on investigations in the natural sciences, social science and computer science. Trust is one of the key project concepts that have meaning in several of these areas. This section has examined the

various meaning that trust (as well as related concepts like reputation) has in these areas. We have also looked at how trust is interpreted in existing research on digital ecosystems (specifically the trust types as defined and used in the DBE project). As our main objective is to apply trust in a computational manner to make digital ecosystems work, we next examine trust management strategies (both credential-based and evidence-based) and identify requirements for effective trust management. This section concludes with a list of possible computational trust representation formats.

It is probably worth concluding with a statement of the definition of trust that we (in OPAALS) choose for our DE model. Specifically, we choose to adopt that of a social scientist, Gambetta (see section 2.2.1) of trust as a subjective probability with which a participant assesses that another participant (or the infrastructure) will perform a particular action. Decisions are then made based on a level of trust – i.e. depending on the action to be performed, a trust-based decision by a participant will be based on a comparison between the participant's trust in another participant (or the infrastructure) and a trust threshold that the participant has for that action.

3 Identity, reputation and their impact on trust

A Digital Ecosystem is a distributed environment in which there is no big-brother or central authority. A Digital Ecosystem functions as a peer-to-peer network in which all peers are equal and free to join or leave the system, interact with each other mainly for business transactions, and form stable or unstable coalitions. In such a system, peers behave like autonomous agents. Each agent is free to reason and decide about the trustworthiness of other agents and decide with whom to transact.

Using a reputation-based trust model, agents can cooperate with each other in order to assess the trustworthiness of unknown peers based on the experiences of others. Thus, the reputation a peer has in a Digital Ecosystem is very important. From an economical point of view, reputation is an asset. Reputable peers are trusted for transactions which helps them make more profit. Sellers with a high reputation can increase prices while sellers with a bad reputation will not be able to attract any buyers and will eventually be eliminated form the system. From a sociological perspective, reputation helps building mutual trust between peers which is the basis for every interaction and cooperation such as a single transaction or a large coalition. From a computer science point of view, assigning reputation values to peers increases the security of the system. In this way, peers can prevent defected transactions, malicious attacks on peers or on the system as a whole.

In order to assign reputation values to entities, each entity that takes part in a DE needs to have a digital identity which is used when interacting with other entities. Peers do not interact with each other directly, but online through the system which increases mistrust. Dellarocas [9] shows that the more separated in time and space the entities are, the greater the risks. In Digital Ecosystems, identity management becomes a bottleneck especially when entities from different administrative domains interact with each other. Identity management is an important part of any reputation system, either centralised or decentralised.

The last aspect we have to consider is the evolutionary nature of the ecosystem. Relationships evolve during the lifetime of an ecosystem and peers change the way they interact with each other. Peers can only evolve in time and change their behaviour if a learning mechanism is used. In computing the reputation of a peer, information from outside the system can be additionally used like for example existing trust relationships and institutional trust or inter-ecosystem trust.

In the following we will analyse current solutions for the main components of a trust model that we identified: identity management, and reputation systems, both centralised and decentralised.

3.1 Identity management

There are a number of industrial approaches offering identity management solutions such as OASIS SAML[1], Liberty Alliance[2] and WS-Federation[3]. The key idea behind these systems is to enable a multilateral federation of partners sharing the same domain (circle) of trust. Each federation supports multiple identity providers and within a federation (circle of trust) a user may traverse all involved partners' services with a single authentication.

However, as we described in [47], a proper identity management model that scales to the DE nature should go beyond a federation-based concept and rather provide:

[1] OASIS Security Assertion Markup Language: http://www.oasis-open.org/committees/security
[2] http://www.projectliberty.org
[3] http://www.ibm.com/developerworks/webservices/library/ws-fed

- user-centric identity management: each entity will be the sole holder of its identity information,
- peer-to-peer or a hybrid (partially hierarchical/federated) model of trust relationships between identity providers (authorities issuing certificates and identity tokens),
- brokering trust of identities and authentication information between different DEs.

When joining the system, agents are assigned a unique identifier which they use to interact with other peers. Peers desire to have a certain level of *privacy*, so the identifier is usually a pseudonym. Pseudonyms are then used to identify parties when negotiating with different ecosystem domains. Pseudonyms can be used to achieve different levels of anonymity. Other properties that might be desired from identities are *spoof-resistance* and *unforgeability*. Spoof-resistant identifiers prevent malicious peers from impersonating others. This can be achieved by using public/private key pairs and nonces to prevent replay attacks. Unforgeable identities prevent peers from having several identities in the system. It also prevents whitewashers from leaving and rejoining the system with a new identity in order to wash up a bad reputation. Achieving these levels of anonymity and security is a complex and costly task for an identity management system. By using certificates and a reputation system based on institutional trust and a dynamic learning component we aim to achieve both a high level of security and anonymity.

3.2 Reputation-based trust models for peer-to-peer systems

Trust mechanisms are needed to assess the trustworthiness of peers and data in a distributed environment. A trust mechanism helps to deter malicious behaviour. Reputation can help build trust between participants based on the assumption that peers who behaved well in the past will behave accordingly in the future. Hence, the higher the reputation of a peer, the more trustworthy the peer is considered to be.

A reputation system helps establishing mutual trust (or distrust) between peers by assigning reputation values to each. A reputation system collects trust scores about peers from other peers and based on these assigns a reputation value to each peer. The reputation values are available to all peers in the system to be used for assessing the trustworthiness of an unknown peer. Resnick et al. [45] identified the following challenges in building a reputation system:

1. Provide relevant information to peers such that they can distinguish between trustworthy and untrustworthy peers.
2. Provide incentives for peers to be trustworthy.
3. Discourage untrustworthy peers by taking actions against them.

In building a reputation system, several aspects or components need to be defined. Swamynathan [10] proposes a classification of these components similar to the following one:

Recommendation context and generation: Peers communicate trust information to each other through recommendations. The reputation system needs to define the context of reputation (peer, data, institution, user etc.), what kind of information a recommendation contains, and who and when generates recommendations.

The reputation-based trust model: The trust model defines how individual ratings are converted to a reputation value. The model defines the sources of information to be used and by whom the reputation values should be computed (e.g. individual peers when needed or the system as a whole). The reputation values could change with every transaction. The model defines how the system handles this to update the reputation values. The model also defines how to handle newcomers and if there are any feedback incentives.

Storage of recommendations and reputation values: Recommendations and reputation values are stored in the peer-to-peer network. The system needs to define what kind of data is stored, how and where.

Recommendation and reputation exchange protocol: This component describes the exchange of information (recommendations and reputation values) between peers: how the information is exchanged and between which peers.

Taking actions: This component describes how the system handles misbehaved peers as discovered by the trust model. Once these peers are identified, different actions could be taken against them.

Security: This component assures the integrity of the data exchanged and stored between peers. It also analyses possible vulnerabilities of the system to different attacks. For example, it provides means from protecting against unfair ratings and colluding parties.

The aspects described above are addressed in current peer-to-peer reputation systems. In the following we will provide an overview of how these issues are handled in a representative set of reputation models and systems. However, the model we propose for DE introduces an additional dynamic component: *learning,* based on the notion of evolutionary trust.

The following reputation systems and models will be analysed based on the criteria described above:

- **TrustMe** [12]: randomly assigns to each peer a Trust-Holding Agent which collects trust ratings from other peers. It emphasises on anonymity and security. It relies on broadcasting for querying and reporting trust values which makes it too expensive for large and unstructured networks.
- **PRIDE** [13]: Each provider stores locally its recommendations. A recommendation is an SPKI[4] certificate signed by the party making the recommendation. A positive recommendation increases the reputation by one and a negative recommendation decreases the reputation by one.
- **XRep** [14]: reputation sharing protocol proposed for Gnutella.
- **PeerTrust** [19]: is designed for quantifying the trustworthiness of peers in p2p eCommerce communities. PeerTrust relies on a transaction-based feedback system to evaluate the trustworthiness of a peer. The feedback system collects the feedback in terms of amount of satisfaction from peers after a transaction occurs. The model takes into consideration five main factors: the feedback (amount of satisfaction) a peer obtains from transactions with others, the number of transactions the peer has performed with other peers, the credibility of the peers who submitted the feedback, a transaction context factor and a community context factor.
- **EigenTrust** [18]: assigns to each peer a unique global trust value which reflects the experiences of all peers in the network with the particular peer. Each peer keeps a local trust value (sum of all individual transactions) for every peer it has interacted with. Each peer has several score managers assigned using a DHT which compute and store the global trust value of the peer. The global trust vector which contains the global trust values of all peers is computed by all peers collectively and is the stationary distribution of the Markov chain defined by the normalised local trust matrix.

[4] SPKI certificate theory, 1999. IETF RFC 2693: http://www.ietf.org/rfc/rfc2693.txt

- **PowerTrust** [16], [17]: uses a similar approach to EigenTrust. Global trust values are computed iteratively be all peers and values are updated by using power nodes (peers which transacted with a very big number of peers).
- **Poblano** [20]: was designed for JXTA and relies on the concept of transitive trust. A chain of trust is found from the peer making the query to the unknown peer. The trust value of a node in the chain is the opinion of the node prior to it. Poblano simply averages these values which makes it vulnerable to attacks.

3.2.1 Recommendation context and generation

Based on an identity scheme, the reputation system gathers information on agents' behaviour and computes their reputation. The system needs to define what kind of information is being shared, how is being shared and which are the sources of information. Trust information is communicated through a recommendation.

First of all, the system needs to define the context of the recommendation: peer, data, institution, user, etc. Current reputation models could be classified as: *peer-based* (e.g. PeerTrust, PRIDE), *resource-based*, and *peer and resource-based* (e.g. Xrep).

There are two kinds of approaches for reporting trust information: the *transaction-based* approach in which each peer provides feedback after each transaction, and the *user-based* approach (e.g. TrustMe, EigenTrust, PRIDE) in which peers give a reputation rating based on the whole history of transactions with a particular peer.

Another aspect to consider is whether the recommendation is referring to the quality of the service provided, or to the quality of reputation ratings provided. Reputation systems usually assume that peers who provide reliable services also provide reliable trust information (e.g. EigenTrust). Swamynathan et al. [7] developed a model that assigns to peers two sets of reputation ratings: an aggregated service rating (*s-rating*) and an aggregated feedback rating (*f-rating*). Additionally, the system maintains for each peer a set of peers that have rated it and the ratings the peer made. At the end of a transaction, a peer sends feedback about the quality of the service and also sends feedback about the opinions that were expressed about the peer. Opinions are weighted by the system based on the credibility of peers. This approach prevents against peers who provide good services and badmouth targeted nodes (e.g. competitors). The simulations performed by the authors showed that using a decoupled approach increases the accuracy of reputations and results in fewer malicious transactions, false positives and false negatives. Buchegger, S. and Boudec, J.L. [8] follow a similar approach in a reputation system for peer-to-peer and mobile ad-hoc networks. Each peer maintains a reputation rating and a trust rating about all peers it has interacted with. A Bayesian approach is used to tackle the problem of false ratings.

The last issue that needs to be defined is what kind of information is contained in the recommendation and how is the trust rating represented (e.g. binary, discrete values, interval) and what is the meaning of the rating. Poblano uses discrete levels of trust with the following meanings: -1 Distrust, 0 Ignore, 1 Minimal trust, 2 Average trust, 3 Good trust, and 4 Complete trust. EigenTrust uses normalised values between 0 and 1. In eBay, feedback consists of a positive, negative, or neutral rating, and a short comment.

3.2.2 The reputation-based trust model

Using established network identities, a reputation system protocol gathers information about a peer's behaviour in past transactions to determine its reputation. The trust model describes how a peer's reputation is computed based on the information collected about that peer from one or

multiple sources. Current trust models take different approaches about the sources of information used and how the information is used in computing the reputation value.

3.2.2.1 Sources of information

Information collection can be done in two ways: *reactive* when each peer gathers information individually, or *proactive* when all peers put together their experiences.

Marti et al. [11] classifies the sources of information in the following way:

- **Local information** Users record their experiences with other peers and use this information to decide whether a peer is trustworthy or not. Users do not trust other peers to provide information and rely solely on their own experiences. Local information is considered to be the most reliable one, but it may not always be sufficient. When the user interacts with unknown peers, it has to rely on additional sources of information.
- **A priori trust relationship** Users might know other users or institutions from outside the system and consider them reliable to provide trust information about other peers. In this case, the user relies on local information and information from users or institutions with whom there exists an a priori trust relationship outside the system.
- **Chain of trust** If local information about a peer is not available, the user can ask the opinions of trusted peers. These peers could further ask peers whom they trust until someone who directly interacted with the unknown peer is found. This creates a transitive chain of trust. This information may be more reliable than asking random peers. However, depending on the length of the chain, even this information could be unreliable. Additionally, depending on the number of chains found, this method might be computationally demanding.
- **Global reputation value** Reputation systems collect information about all peers from all peers. The previous approaches are decentralised, while global history systems could be also centralised. eBay [21] uses a single trusted entity that collects all transaction reports and rates each user. Other peer-to-peers systems use a distributed approach: TrustMe uses a centralised server to assign unforgeable identities, but reputation is handled distributed among peers. EigenTrust and PowerTrust use fully decentralised solutions. However, the systems are vulnerable to whitewashing, peers being able to leave and rejoin the system with different identities.

O'Hara et al. [22] give another taxonomy for the Semantic Web based on how agents react when they deal with peers which whom they have no personal experience. They identified five strategies:

- **Optimism**: assume all agents are trustworthy unless proven otherwise. An agent will trust another agent even if its trustworthiness is unknown. Trust is the default attitude. This approach in used if the benefits of cooperation are considerably greater than the costs of betrayal.
- **Pessimism**: assume all agents are untrustworthy unless proven otherwise. This approach restricts interactions with agents unless there is a reason to trust them. Trust is established via personal acquaintances in offline world which is the basic model of trust.
- **Centralised**: trust information is managed and obtained from centralised institutions like Trusted Rating Agencies. The assumption is that everybody trusts the institution and certificates issued by it.
- **Investigation**: check and evaluate agents to determine their trustworthiness. In this approach, trust is a response to uncertainty.

- **Transitivity**: networks of agents cooperate to determine the trustworthiness of a peer. An agent sends out a message to trusted agents about the trustworthiness of another agent. The agents send back an opinion based on personal experience with the agent, or send the query to their acquaintances which are unknown to the agent who sent the message.

3.2.2.2 Computing the reputation value

The next problem to consider is how to use the collected information in order to compute the reputation value. All opinions could be considered equal, or reputation can be used to weigh the collected opinions. Systems that weigh ratings based on the trustworthiness of the source perform better against colluding adversaries and front peers which promote each other and denigrate well-behaved peers. The opinions of a friend would weigh more than the opinion of an untrustworthy peer. The opinion of a peer could be weighed by their previously determined reputation (e.g. EigenTrust, PowerTrust). In case of transitive trust, the reputation of a peer could be the value assigned to this peer by the predecessor. This approach is used by Poblano.

Global history reputation systems also apply reputation-based weighing to the scores. For example, EigenTrust and PageRank [23] use similar distributed algorithms to compute a global reputation value for each peer using transaction reports sent by the other peers and weighed by the rating of the peer sending it.

To improve the accuracy of reports some systems require proof of interaction. TrustMe requires that both parties sign a transaction certificate which is presented when reporting transaction feedback. This solution prevents adversaries from submitting fraudulent reports about peers with whom they have not interacted with the purpose of lowering their reputation. However, it does not prevent peers from lying in their reports.

The reputation value of a peer can be computed by an agent interested in making a transaction with the peer, by a centralised system (eBay) or by all peers collectively as in EigenTrust and PowerTrust.

Including the value of the transaction in the computation of the reputation score is a good approach. A misbehave in a €1 transaction should be weighed different than a misbehave in a €100 transaction. Malicious peers could behave well in several small transactions and then misbehave in one large transaction. This kind of behaviour should be reflected in the reputation value.

The number of transactions should be included in the calculation of the reputation value. Ebay simply adds the ratings (+1, 0, -1) without considering the number of transactions. This allows peers to increase their reputation value by increasing the number of transactions. In this way peers could misbehave in many transactions without having this reflected in their reputation value.

Another aspect that might be interesting to take into account is the time when the transaction took place. Recent transaction information might be more useful than older information. In this case, transaction information would need to be weighted based on the time at which occurred. Ebay, for example, only considers transactions that took place in the last six month.

Hence, when computing a reputation value, it is important to consider the number of transactions, the value of the transaction, the time at which the transaction occurred, and the credibility of the source. PeerTrust introduces additional optional criteria like the context of the transaction and the community.

Systems that use global trust values need to update these values (e.g. PowerTrust).

3.2.2.3 Stranger policy

Another issue that the trust model should consider is how to handle an agent whose trustworthiness is unknown to the system. Feldman et al. analysed the problem of stranger policies and whitewashing in peer-to-peer systems [24], [25]. They suggest a "stranger adaptive" strategy in which first-time interactions with strangers are aggregated together by all peers in the system. Using a "generosity" metric based on recent stranger transactions, an agent can estimate the probability of being cheated in the next transaction with a stranger.

3.2.3 Storage of recommendations and reputation values

Reputation information is stored distributed in the network. The storage needs to be efficient such that the information can be retrieved with little communication overhead.

Storing feedback about every transaction that took place would create a considerable overhead. To prevent that, transaction ratings of one peer for another could be aggregated and updated after every transaction such that a single value is stored. For example, after peer A buys a service from peer B, it simply updates the trust value it has for peer B instead of storing another rating. Moreover, old transaction ratings could be replaced by new ones once. Ebay replaces all transaction ratings older than 6 month.

Reputation data is stored among participants. One approach is to have every peer store its own ratings (i.e. received from the peers with whom it has interacted). Another approach is to assign to each peer one or several peers who store the ratings. The third approach is to have each peer storing own opinions about other peers and make them available on request.

In PRIDE, each peer maintains its own trust scores digitally signed by those who issued them such that they cannot be modified.

In PeerTrust, recommendations about a peer are stored at designated peers located by hashing the unique ID of the peer. EigenTrust and PowerTrust use a similar approach. Each peer's reputation information is stored by several score managers designated by using different hash functions. Peers are not aware for whom they are storing and computing the trust values. TrustMe randomly assigns to each peer a Trust-Holding Agent. This is a DHT-based structured way of storing the recommendations. Data integrity is assured through redundancy (several score managers), anonymity, and digital signatures on recommendations.

XRep uses an unstructured approach. Each peer maintains an experience repository of resources and peers with whom it has interacted. Before downloading a file, a peer sends a request and all peers which interacted with the unknown peer respond with their ratings. Ratings are validated by sending another poll to the peers who responded. Validated ratings are then used to determine the trustworthiness of the peer. Data integrity is assured by using digital signatures on recommendations. This approach is not scalable because of the big number of messages being sent. Another disadvantage is that it only considers ratings of online peers.

3.2.4 Recommendation and reputation exchange protocol

The exchange of reputation information happens in two stages: before the transaction occurs the requester searches for reputation information to decide which service to buy, and after the transaction occurs the requester rates the service provider. The exchange protocol needs to ensure the integrity of the reputation information travelling through the network.

XRep uses broadcasting on the Gnutella network. Poll messages are implemented on top of the ordinary Query messages. To ensure the integrity and confidentiality of the poll responses, the poll

request includes a public key (generated ad hoc) with which the response is encrypted. PRIDE uses digital signature to protect the recommendations.

In DHT-based approaches, a peer is responsible for storing the reputation information of several peers and also maintains a routing table for reaching other peers. If it receives a request for a key (peer) that is not responsible for, it forwards the request according to the table. EigenTrust and PowerTrust use this approach.

3.2.5 Taking actions against misbehaved peers

Reputation systems can take different actions when discovering misbehaved peers. The first thing that can be done by the reputation system is to inform agents about peers who are likely to defect on a transaction. In this way agents will avoid malicious peers who will not be able to cheat others any more. Furthermore, overlay network neighbours can disconnect from the malicious peers, ejecting them from the network. The adversary could be kicked out from the network for a period of time or permanently banned. Finally, systems attached to a financial institution for monetary payments could fine the malicious peers for each verified act of misbehaviour.

3.2.6 Security

One of the most important problems that the security component needs to overcome is the issue of dynamic peer personalities. Peers are honest sometimes and dishonest other times. Weighing recent feedback more than older feedback would force peers to maintain an honest behaviour. This kind of approach also allows peers with a bad reputation to build a good one by behaving well, encouraging in the same time newcomers to enter the system.

Dishonest peers not only provide bad services, but also provide false ratings. There are two kinds of threats to consider here: false positives (i.e. colluding peers cooperate to promote each other) and false negatives (i.e. malicious peers give bad recommendations to well-behaved peers to damage them). This can be prevented by incorporating the credibility of the source in calculating the global trust value. Several systems use this approach. PeerTrust, XRep, EigenTrust and PowerTrust are a few examples.

In a decentralised network, participants can have several identities, and leave and rejoin the system with a new identity to wash up a bad reputation. Giving newcomers little power discourages whitewashers, but it can damage legitimate newcomers. A participant could create a large number of identities and use them to give false recommendations to one of its identities. This kind of attack is called liar farm. PRIDE implements an IP Based Safeguard method that defines a security zone and averages all recommendations received by a provider from identities places in one zone. This is possible because PRIDE uses self generated certificates which not only bind an identity to the public key, but also other information like the range of IP addresses from which the identity can be used. XRep also checks IP addresses to prevent attackers from using multiple identities.

Another kind of attack concerns colluding peers. Dellarocas [9] identified two kinds of collusion attacks: *ballot stuffing* and *bad-mouthing*. In ballot stuffing, colluding peers increase the reputation of a peer such that it can be used afterwards for malicious purposes. In bad-mouthing, colluding peers give unfair ratings to other peers in the network to damage their reputation. EigenTrust aims at preventing this by using pre-trusted peers while PeerTrust uses a feedback similarity function that computes the similarity of ratings of two peers over a common set of peers with whom they interacted.

Reputation data is stored by peers in the network. Malicious peers could manipulate these values. To ensure integrity of stored reputation data, several approaches can be used. Redundancy of data

by using several peers for keeping the information is used by EigenTrust and PowerTrust. TrustMe uses digital signatures to ensure the integrity of recommendations.

Reputation information is exchanged by peers according to a protocol. Exchanged data might be changed by malicious peers. To ensure integrity of exchanged data, cryptographic techniques can be used. For example, XRep uses encryption to protect exchanged messages.

Table 2 summarises the current approaches in building a peer-to-peer reputation system.

Recommendation context and generation	**Who/what is being rated**	**What aspect to rate**	**Metrics**
	Resource based Peer based Peer & resource based	Quality of service Quality of recommendation	Binary Continuous range Discrete values
The reputation-based trust model	**Source of information**	**Criteria**	**Stranger policy**
	Own experiences Recommendations from a chain of trust All experiences of the system A-priori trust relationships	Credibility of the source (weighed opinions) Value of transaction Number of transactions Recent vs. older information	Trust all Trust none Estimate whether to trust

Storage of recommendations and reputation values	**What to store**	**Where to store**	**Data authenticity**	**Data credibility**
	Store individual transactions vs. aggregate values	Each peer stores received ratings Each peer has a ratings keeper (using hashing) Each peer send out ratings when requested	Signing of ratings	Proof of interaction

Recommendation and reputation exchange protocol	**Exchange mechanism**	**Integrity & confidentiality of messages**
	Broadcasting DHT approaches	Public key Signature

Actions against misbehaved peers	**Community awareness**	**Architectural action**	**Banning**	**Financial action**
	Inform peers about cheaters	Overlay networks exclude misbehaved peers	Permanent or temporary banning from the system	Fine misbehaved peers

Security threats	**Peer behaviour**	**Ratings correctness**	**Identity uniqueness**	**Peer coalitions**
	Dynamic personalities: sometimes peers lie and other times tell the truth	False positives False negatives	Multiple identities: liar farm Whitewashers	Collusion: ballot stuffing and bad-mouthing

Table 2: Peer-to-peer reputation systems approaches

Current reputation systems only consider a few of the aspects discussed in this chapter.

3.3 Credit rating agencies

A peer-to-peer reputation system is a decentralised solution for assigning reputation values to peers. The centralised approach is using a rating agency that gathers information and transaction feedback from the system to compute ratings for peers. Rating agencies for peer-to-peer reputations are a new research field and we will base our model on Credit Rating Agencies (CRA).

CRAs [52] [53][54] have been first introduced in 1906 in US. CRAs are institutions that assess and publish credit opinions, research and ratings on fixed-income securities, issuers of securities and other credit obligations. Rating agencies differ from public and private credit registries (or credit bureaus) because together with the *"assessment of how likely an issuer is to make timely payments on a financial obligation"*[55], covering information asymmetry between issuers and buyers, they also provide information related current and prospective factors that may also affect credit risk in the future.

The most recognised rating agencies worldwide are Standard & Poor's (S&P), Moody's and Fitch Ratings. These leading rating agencies established specific methodology of drawing up the rating reports. Since the importance of those ratings for the investors CRAs' methodology and processes, usually centrally controlled and managed, have been mostly standardised and the activities and all the necessary information on rating decisions are available on the Internet.

Although heavy standardisations in the rating market some differences in the evaluation procedures could be still highlighted even because the exact processes used for rating may be proprietary [55]. In particular some CRAs (including many of the larger CRAs) count on a process where multiple analysts create an assessment, based on qualitative and quantitative indicators, that is then reported to a rating committee for a final judgment. Ratings committees are generally formed to start, update or withdraw a rating and are not permanent. Committees are composed by lead analysts, managing directors and junior analytical staff. The opinion about the likelihood of an issuer to repay its obligation could be made upon a simple majority vote.

Other CRAs rely more on a mechanical process to arrive at a rating (statistical analysis of all available information and all financial disclosures about an issuer).

3.4 Digital ecosystems and reputation systems

A Digital Ecosystem consists of diverse actors which, like animals in a natural ecosystem compete against each other, collaborate, form stable or unstable federations and generate niches. Companies can cooperate some times and be competitors other times. Digital Ecosystems are interconnected by an ICT network to form a complex and dynamic environment.

Entities in such an environment are selfish and act in their own interest. However, they know they will be evaluated by the quality of services provided by a reputation system and that the success of their business in the ecosystem depends on the reputation they have.

Furthermore, there is no central coordinator and no central database storing reputation values. Reputation ratings are shared and computed among participants in a distributed fashion.

Transactions occur between participants and are atomic in nature. They have to complete in order to be rated. Each individual will rate the outcome of the transaction subjectively. Individuals might lie about outcomes to damage a competitor or to promote a friend. Different individuals might rate differently the same outcome because they have different expectations or rating systems.

In such a dynamic environment in which there is no central authority controlling the system, individuals could be able to create multiple identities or change their identifier. An identity management model suitable for DEs makes the system more secure.

Natural ecosystems are self-organising, self-managing, scalable, and able to provide complex solutions and an automated composition of solutions [5]. Agents in a DE are like agents in a natural ecosystem: they reproduce, vary, interact, move and die. Levin describes ecosystems as complex adaptive systems [6]. Ecosystems are comprised of diverse components which interact locally and are subject to selection. This explains the self-organising nature of ecosystems which results when interactions among agents and their environment gives rise to complex non-linear behaviour.

In order to build a trust model for Digital Ecosystems, a reliable identity system based on standards and a robust reputation system are needed. For building a reputation system, two approaches can be taken: a purely decentralised one such as a peer-to-peer reputation system or a centralised approach such as rating agencies. In our model, we accommodate the notion of rating agencies to the distributed and dynamic nature of DE and propose a hybrid approach which uses both centralised and decentralised techniques. To further accommodate the nature of DEs, we will introduce a dynamic component based on the notion of evolutionary trust.

4 An Adaptive trust model for DE

In Chapters 2 and 3 we analysed trust definitions, representations and characteristics in different disciplines such as sociology, biology and computer science. The reason for doing so is that the DE philosophy is inspired by different sciences. In this chapter we conclude our multidisciplinary research on trust with a trust definition suitable for DEs. The trust properties that we describe suit different characteristics and needs of the DEs. Furthermore, we propose an adaptive trust model which has several components.

4.1 The notion of trust in OPAALS

Before describing the model we are proposing, we summarise the trust properties we are modelling, inspired from the study of trust in different disciplines. We define trust as the confidence an agent has that another agent will behave well in a certain situation. Thus, trust is a unidirectional relation between a *truster* and a *trustee*. In a DE, a truster can be a user, an agent, an organisation and a trustee can additionally be a service, data, or infrastructure. Trust can also be defined at DE level and expressed as inter-ecosystem trust. Trust has the following properties:

- **subjective**: different agents have different opinions about the same agent, based on their own experiences and perceptions.
- **context-dependent**: trust is not defined in general, but in a certain context, i.e. A trusts B to behave well in context T (e.g. automotive)
- **present at several levels**: trust is expressed at different levels in the digital ecosystem like user, data (knowledge), services, infrastructure, and ecosystem.
- **multi-dimensional**: trust does not always have to be represented as a simple number. Complex values that regard different dimensions of the trustee can be used. For example, a service can have different trust values for: availability, response time, accuracy of result, memory usage, etc.
- **dynamic**: trust values change permanently based on new experiences and changing environment conditions.

Hence, we can say that in a DE trust is **adaptive** and responds to context changes, new experiences, and environment changes that can occur at different levels.

Reputation is an expectation about an agent's behaviour based on past actions. Reputation is multilateral and represents how much an agent is trusted by the other agents of the system. This means that all agents share a common opinion on an agent's reputation.

Agents with a good reputation are considered trustworthy. This means that trust is built through reputation. Also, reputation is built from expressed trust relations between agents.

We can further extend the discussion about trust and reputation to different locality contexts. As summarised in Figure 3, reputation can be either globally computed and accepted by all agents or be computed only by agents in a specific neighbourhood. Trust needs to be computed individually by each agent, and in doing so, the opinions of all agents of the system could be taken into account, or only the opinions of a small trusted neighbourhood which could be seen as a social

network of agents. This last approach is the most suitable for digital ecosystems in general and it will be modelled in our approach. There are many reasons for preferring this approach. First of all, entities in a DE are autonomous and choose freely whom to trust, so a globally accepted reputation value is not suitable. Second, entities have unstable relations involving both cooperation and competition. For this reason, they will only share their trust opinions with their partners and the list of partners should be updated dynamically based on environment conditions.

	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

Figure 3: Trust and reputation properties with respect to locality

4.2 Multidisciplinary framework supporting trust in digital ecosystems

The trust model we propose for DE is based on a multidisciplinary framework first introduced by us in [47] and illustrated in Figure 4. The right side column represents possible technology platforms suitable for DE service execution management. The left side column represents the trusted environment that SMEs use to perform their business goals.

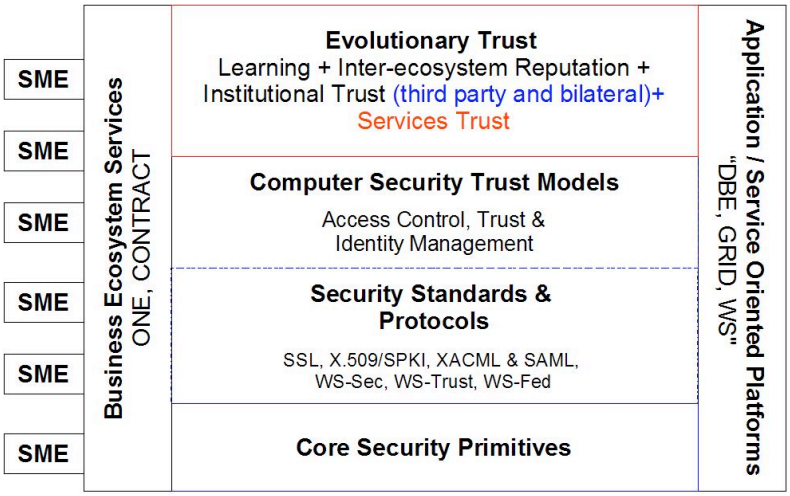

Figure 4: A Multidisciplinary Framework for Digital Ecosystems (Telesca L. and Koshutanski H.)

We target a model that would provide ecosystem trust at different levels:

- Infrastructure
- Services
- Knowledge (information or data)
- User

In order to provide trust at all these levels, we will monitor service execution using an accountability model. Using this together with a reputation model and a context-aware model would achieve trust at all different levels.

As motivated previously, for defining a trust model for Digital Ecosystems, we need to consider the following components:

1. Distributed identity management
2. Reputation
    a. Decentralised: Peer-to-peer reputation
    b. Centralised: rating agencies
3. Learning: evolutionary trust

Figure 5 shows these components and the relations between them.

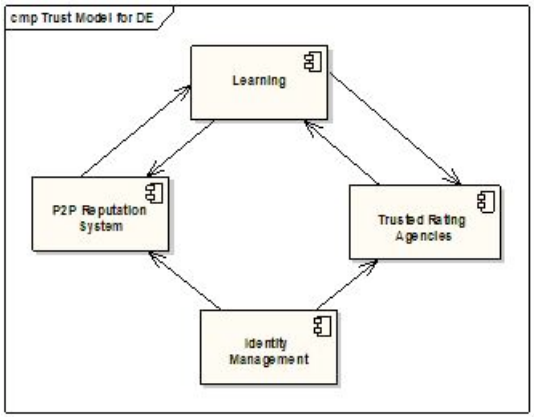

Figure 5: Trust Model Components

The Distributed identity management component already introduces a certain level of trust and security into the DE by assigning to each user an identity certified by other trusted entities in the system.

The Peer-to-peer reputation system and the Rating Agencies component rely on the Identity Management model and assign reputation values to DE identities. Users' reputations are built and evolve into the DE through transaction interactions with other users. To update the reputation values, the Peer-to-peer reputation system and the Trusted Rating Agencies run algorithms on transaction feedbacks.

Though the two components appear to have the same functionality, they actually address different aspects and needs of the DE and therefore they can be seen as complementary. The P2P reputation system views peers in the DE as equal and self-organising entities which cooperate with each other to compute the reputation values. Though any peer in the DE can be a rating agency and users are free to choose which agency to trust, the Rating Agencies model introduces a certain level of hierarchy which results in higher credibility of the rating certified by an agency. Peer-to-peer reputation is subjective while certificates issued by rating agencies are objective. Hence, the rating agencies model is suitable for commercial transactions with certain restrictions. The two models are complimentary and there is no required dependency between the two. Depending on the criteria a certain agency uses, reputation information could be taken into consideration when issuing a rating, but will not be the only criteria. Although any peer can be a rating agency, in order for a peer to act as a rating agency in a DE, high computational and administrative capabilities are needed. Therefore, the P2P reputation system is the most simple and cost-effective way to cover the basic needs of the DE. As soon as the system evolves and different ecosystem instances are running, in order to ensure the correct functionality of the system, rating agencies are needed. For example, the agencies will need to translate trust values from one ecosystem to another such that inter-ecosystem communication and transactions are possible.

To learn from past transactions and users' behaviour, historical data produced by the two systems can be input by a Learning component. This component can derive new rules and patterns used by the two systems to improve their rating algorithms. In this way, the trust model we propose evolves in time together with the DE.

4.3 Distributed identity management for DEs

The Distributed Identity Management model we present in this section was published in [46] as part of our research on Digital Ecosystems.

A DE is a dynamic and unstable environment composed of diverse institutions which sometimes compete and other times collaborate. As a result, stable and unstable federations are created which poses many challenges to managing identities. First of all, organisations use divers types of certificates and identity technologies (e.g. X.509[5], SPKI, Kerberos and OpenID[6]) which are not always compatible with each other. Secondly, users often need to access applications, services or a composition of services located on different administrative domains. Finally, because of the dynamic nature of the environment, federating and sharing of identities becomes a complex task.

WS-Policy[7], WS-Trust[8] and WS-Federation[9] offer identity federation solutions, but these are pure federating approaches are hence viable only when there is a stable federation. In DEs, federation does not scale up because of the unstable and ad-hoc coalitions. Moreover, these models are heavy to implement by SMEs, being more suitable for large enterprises. What DEs need is an identity management model able to provide:

- exchange of identity information between companies independent of the standards they use, and
- sharing user identity between different domains, either federated or with no direct trust.

Additionally, the model needs to be targeted, easy to understand and straightforward to implement by SMEs.

We propose a model that satisfies these requirements and automates the process of identification between DE partners.

The key entities in the model are:

- **User**: any entity in the network (peer or web browser user, institution or person)
- **Service Provider (SP)**: any entity that has one or more services or resources available to other entities.
- **Credential Provider (CP)**: any entity that is able to provide digitally signed credentials to other entities.

The model considers all users to be equal and assumes there is no hierarchy of DEs. Each peer can be a CP (thus issue identification and authorisation certificates to other peers) and each peer can be a SP (thus provide services and resources to other peers). A peer can be both a CP and a SP at the same time. Each user can freely choose what kind of certificates to accept. For that matter, each user and SP has a list of trusted CP and only certificates coming from a trusted CP are accepted. CPs can establish trust relations between them and accept certificates coming from a trusted CP.

[5] Public-key and attribute certificate frameworks, 2005. ITU-T Recommendation X.509:2005 | ISO/IEC 9594-8:2005.
[6] http://openid.net
[7] http://www-106.ibm.com/developerworks/library/specification/wspolfram
[8] http://www-106.ibm.com/developerworks/library/specification/ws-trust
[9] http://www-106.ibm.com/developerworks/webservices/library/wsfed

Each CP has a list of accepted security tokens (i.e. issued by a trusted CP) and issues certificates based on:

- secure tokens issued by the provider itself or
- trusted secure tokens (from CPs with whom it has trust relationships) or
- based on user registration information.

Thus users obtain a number of security tokens from different CPs that can belong to different DEs. The model also allows users to import certificates obtained outside the DE. This approach has several advantages. First of all, for legal and economic reasons, sometimes companies are required to use a certain kind of certificates (e.g. issued by a certain certification authority which is outside the system). Secondly, it allows building trust and networks of trust in the system by using trusted certificates and relations from outside the system. Moreover, it allows the system to take off fast and newcomers are accepted in transactions more easily. The lake of trust relations between peers can be a bottle neck to the growth of a new peer-to-peer system. By using this approach, our model overcomes this problem.

The model is based on the new SAML[10] (v2.0) standard for bridging different standards that companies might use (e.g. X.509, SPKI, Kerberos tickets, username/password etc) and for exchanging identification and authorisation information independent of the technologies used by different SMEs in the DE. To cope with a wide range of identity mechanisms we require for every CP to support by default SAML v2.0. This means that each SP adopts the identity standard best suiting its needs but its related CPs supports SAML. This means that any SME could preserve its existing identity management infrastructure but should enhance its trusted CP with the ability to understand SAML. Furthermore, each CP must be able to issue SAML assertions derived (transformed) from any of the standards the CP supports.

Every user has an identity profile distributed in the peer-to-peer network which can be accessed by the user from any peer in the system using unique username & password chosen at registration. This approach accommodates users which do not own a computer and provides ease of use of the system. A user profile contains a list of all available certificates and security tokens the user has. The profile is encrypted with a long master password known only by the user and never stored in the system. To make the profile available at all times and to provide a secure storage, we replicate the profile on trusted nodes. Each DE has a number of trusted nodes determined by the peer-to-peer reputation system.

The model is based on transformation of security tokens from one CP to another and is further described in Figure 6.

[10] Security Assertion Markup Language (SAML), 2005. www.oasis-open.org/committees/security.

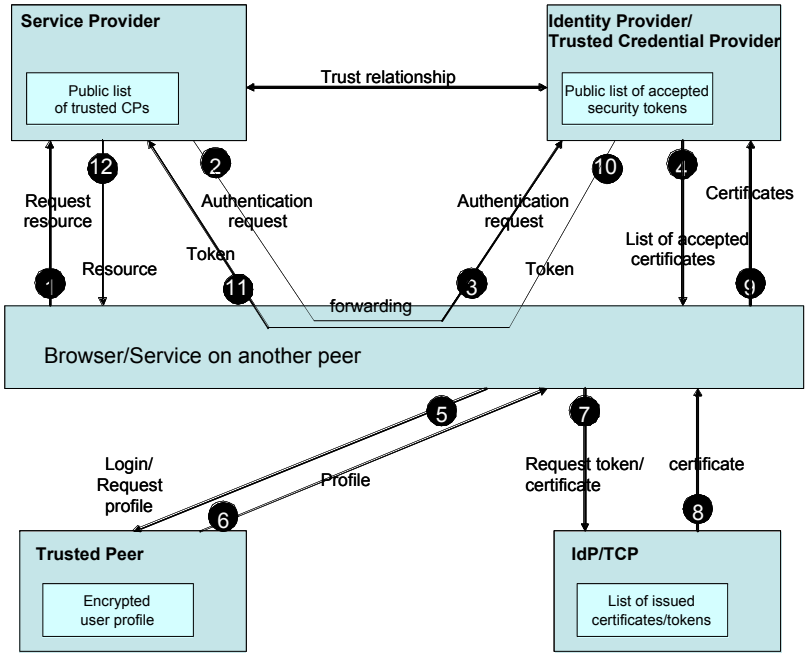

Figure 6: Basic Identity Management Model

When a user tries to access a service (1), the SP redirects it to the trusted CP (2-3). If the user already has a token from this CP, then the user can simply authenticate and be redirect to the SP. If the user is not known by the CP, the CP will provide a list of accepted security tokens (4). The user has his profile replicated on trusted peers and can login with username and password (5) and download his profile and decrypted on a secure memory with the master password (6). The user can check if there is a match between the requested tokens and the tokens in the profile. If a match is found, the user requests a token from the CP who issued it (7-8). The user provides this token to the CP (9) who will issue a new certificate that is understood and accepted by the SP. The CP might need to make a translation, i.e. issue another type of certificate that is understood by the SP.

Users wish to protect their privacy. For that matter, each CP has the responsibility to provide proper pseudonymity to end users by using a pseudonym, either chosen by the CP or by the user. The pseudonym is then certified in a trusted secure token to a SP. In case of misbehaviour, the SP explicitly asks a CP to reveal user identity. So, each CP maintains a database mapping user's pseudonymity with user's real identity.

Once we have defined a proper identity system, reputation systems can be built on top of it.

4.4 Peer-to-peer reputation system

For the purposes of this work, we adopt a two layer model for communications between peers. Peers can either interact for a transaction or to exchange trust information. For modelling purposes, each service usage interaction is a discrete event. A logically separate trust management layer handles threat notifications and other pertinent information.

We mimic social trust by setting a fuzzy trust level. Each different service can then make an appropriate decision based on this trust level – e.g. certain actions may be allowed and others not. In our system, we model trust as a vector. In the simplest case, at least if there is just one service, this can be viewed as a simple number in the range (0,1). Each peer may then maintain a local trust score relating to each other peer of which it is aware. If peer A's trust in peer B is 1, then peer A completely trusts peer B. A score of 0 indicates no trust. Note that this is consistent with

Gambetta's widely accepted definition of trust as *"a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action and in a context in which it affects his own action"* [30].

4.4.1 Trust representations

If we want to represent trust as a complex value such that it is more meaningful and conveys more information about the trustee, each dimension of trust will be represented as a probabilistic value from 0 to 1. For example, if agent A wants to express the trust it has about service S, it can use the following structure:

(A, S, availability, 0.9)
(A, S, response_time, 0.3)
(A, S, result_accuracy, 0.8)
(A, S, memory_usage, 0.2)
(A, S, security, 0.6)

Users or companies could be trusted in a certain context, but not in another one. For example, A can trust B about car fixing, but not about baby sitting. In order to express trust in a particular context, an agent can use the following structure: *(TrusterID, TrusteeID, Context, trust_value)*. Trust_value can either be simple or multidimensional as showed above. For defining contexts, we will allow users to build a folksonomy by defining own keywords for expressing contexts.

4.4.2 Architectural overview

We propose the overlay of a distributed trust management infrastructure on top of the service delivery infrastructure (which may itself contain multiple layers).

Figure 7 illustrates the relationship between underlying services and this new infrastructure. With our proposed trust overlay architecture, a trust management layer operates separately from the mechanics of the transactions themselves. Two message passing interfaces are defined between the transaction layer and the trust management layer and another between the trust managers of individual peers. The interfaces are as follows (Figure 7):

(1) Experience reports: Transaction engine □ Trust manager
(2) Trust recommendations: Trust manager □ Trust manager
(3) Policy updates: Trust manager □ Transaction engine

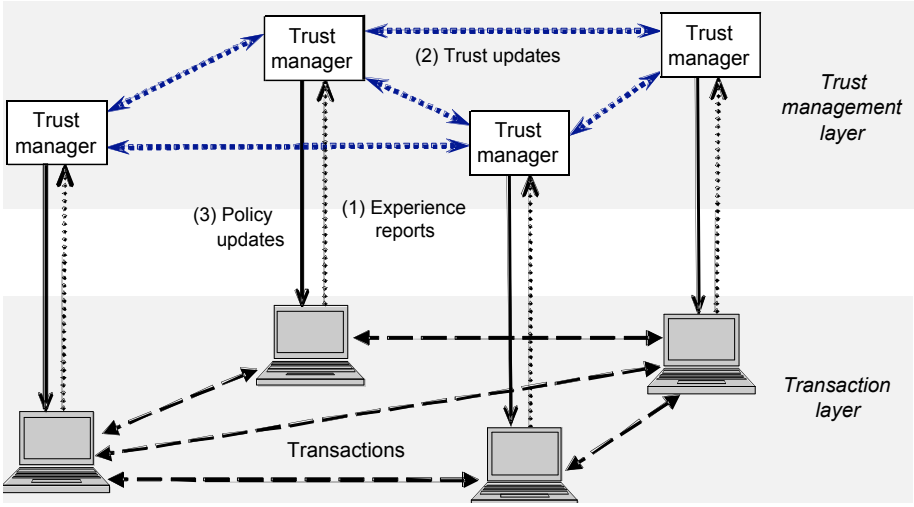

Figure 7: Trust overlay helps to secure transactions in a digital ecosystem.

Transactions are as normal, with the trust information just influencing protection mechanisms. The trust manager gathers transaction experience and uses this together with the experience of collaborators to inform its trust in other peers.

In the initial case, all trust values are set to a low value. Having initial values set to zero would prevent the so called *Sybil Attack* [29], whereby attackers can take advantage of a default trust level by maintaining multiple identities. This is impractical though as we need to have some low-risk services enabled in order to get experience of other peers.

We then need to have a system to build up trust as peers gain experience of each other or learn of each other's reputation. In our system, trust can be updated in two ways:

1) *Direct experience:* On completion of a transaction between two peers, each peer updates its trust in the other based on a measure of satisfaction with the transaction.
2) *Reputation:* Peer A notifies other peers in its *neighbourhood* of the trust score that it has for peer B. This will change significantly following a security-related event.

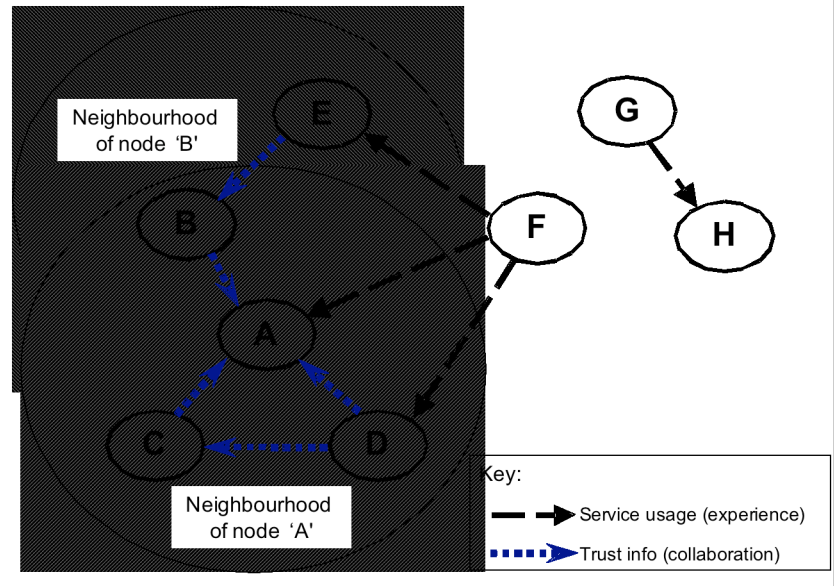

Figure 8: Trust information is shared with "neighbouring" peers

How this neighbourhood is defined is significant. The neighbourhood of a peer is the set of peers with which it can communicate or with which it is willing to interact. The choice of neighbourhood peers is up to each individual peer to decide, and can be viewed as the peer's social network. In reality, choice of neighbours may depend on physical geography, network topology, frequency of contract or even trust level.

The main benefit of this system is in using these trust scores to tune security settings. Trusted peers can be dynamically provided with more privileges than untrusted peers. In our system, as mentioned, we model all interaction between peers in terms of services. Each peer then sets a threshold trust level for access to each service in which it participates. If the trust score of a peer decreases, for example due to detected suspicious activity by that peer, services available to that peer are reduced.

4.4.3 Trust overlay protocol

This section describes a generalisation of a protocol already proposed by the authors in [48] to support trust-based email filtering, named TOPAS (Trust Overlay Protocol for Anti Spam). The TOPAS protocol allows for the collection of spam statistics to calculate trust, sharing this trust information between peers and feeding it back to mail hosts to allow them to more effectively filter spam.

The protocol executes using asynchronous message passing in the case of interfaces (1) and (2) of Figure 7. Interface (3) uses a simple synchronous request-reply technique. An underlying set of services is assumed, including message delivery, failure detection and timeout. It is assumed that each process receives queued messages of the form (*tag, Arg_1, ..., Arg_n*). This outline style of protocol specification is influenced by the Generic Aggregation Protocol (GAP, [49]).

(1) Experience report: Service host □ Trust manager
The service host has an associated access controller. Each transaction attempt is processed by the access controller, with the result that it is either accepted or blocked. Accepted transactions are then monitored (e.g. by an IDS, or a spam filter if the service is email) and determined whether the transaction was benign or malicious. The result of this assessment of the transaction needs to be made available to the trust manager. The following two messages, from the service host to the trust manager, provide this:

- (*singleTransaction*, *i, a*) may be sent from the local service host to the trust manager to report on a single transaction from peer *i*. The value of *a* is 1 if the service has triggered an alarm and 0 otherwise. *i* identifies the peer using the service.
- (*bulkTransactions*, *i*, *n*, *a*) may be sent from the local service host to the trust manager to report on a sequence of transactions involving peer *i*. *n* is the number of discrete transactions by peer *i* (since the last report) and *a* the number of these determined by the service host to be malicious.

Note that in both messages above, the peer identifier *i* could be an IP address, a hostname, a domain name or any other identifier supported by the service host. It is up to the trust manager to process this.

Sending a *bulkTransaction* message conveys the same information as would several *singleTransaction* messages. It is included in the protocol for performance reasons. Where transaction volumes are heavy, it is more efficient to send periodic *bulkTransaction* updates rather than burden the server with generating a *singleTransaction* message per transaction.

(2) Trust recommendation: Trust manager □ Trust manager
Peers collaborate to share trust information with one another. This is done by the trust managers, using the primitives outlined here. Trust managers may issue recommendations spontaneously (for example on a usage alarm) or in response to a request from another peer. Note that a request may be issued by one trust manager to another, but a reply is not guaranteed. As mentioned earlier, message passing is asynchronous and the protocol requires no state information to be maintained by the corresponding entities. The following five messages may be sent from one trust manager to another:

- (*getTrust*, *k*) allows peer *i* to ask another peer *j* to report its trust in peer *k*. This message should be interpreted as an indication from peer *i* of a desire to receive a recommendation regarding peer *k*. Peer *j* may respond with a trust report. This might be expected to be used as follows. On receipt of a transaction request from peer *k*, peer *i* could issue a series of *getTrust* messages to neighbouring peers requesting recommendations regarding peer *k*. Only those neighbours with some experience or knowledge of *k* might reply, with the remainder staying silent. Note though that peers are not obliged to respond, even if they do have knowledge or experience of *k.* Peers could have other reasons to not reply. (e.g. lack of trust in the requester, *i*) Peers that have no information are expected to stay silent.
- (*getTrustResponseRequested*, *k*) is a variation on *getTrust* that expects the recipient to respond (with a null value) even if it has no trust information on peer *k*. Again, there is no strict obligation to respond.
- (*getTrustAll*) allows one peer *i* to ask another peer *j* to report its trust in all known peers. This message should be interpreted as an indication from peer *i* of a desire to receive a recommendation regarding all peers of which the recipient is aware. There is no obligation on the recipient to respond. Peers that have no information on any peers should issue a null reply.
- (*setTrustReportingPreferences*, *t, c, r, f*) allows peer *j* to specify to peer *i* how spontaneous trust reports are sent to it (see discussion on "push" model later). This message also indicates a desire by peer *j* to receive trust advertisements from peer *i* (i.e. to be included in its neighbourhood).
- *t* is a list of zero or more trust level thresholds. Trust updates are requested whenever trust exceeds or falls below any of these threshold values.
- *c* is a confidence level threshold. Trust updates are only desired if the confidence level of the sender is at least *c*.
- *r* is a recency threshold. Trust updates are only desired if the trust information has been updated by the sender within the previous *r* time units.
- *f* indicates the maximum frequency of update.
- (*trustReport*, *k, T*) allows peer *j* to send a recommendation to another peer *i*, in relation to peer *k*. *T* is an object that encapsulates sender *j*'s trust in peer *k*. *T* is set to null in the case where the sender *j* has no trust information regarding peer *k*. Note that a *trustReport* may be issued either spontaneously or in response to a *getTrust* or *getTrustResponseRequested* message.
- (*bulkTrustReport*, *l*) allows peer *j* to send a recommendation to another peer *i*, in relation to a set of peers. Parameter *l* is a set of pairs (*k, T*) where *k* is the peer identifier and *T* is an object that encapsulates sender *j*'s trust in peer *k*. *l* is the empty set in cases where the local peer has no trust scores to share. Note that a *bulkTrustReport* may be issued either spontaneously or in response to a *getTrustAll* request.

(3) Policy update: Trust manager □ Service host
The third part of this collaboration architecture is responsible for closing the loop. Direct experience is recorded by peers and shared among them. The result of this experience and collaboration is then used to inform the service host to allow it to operate more effectively. Specifically, the service host, on a transaction attempt peer *i* needs to be able to access the current trust information that its trust manager has on peer *i*. Although the service host may be able to use local storage to, for example, cache trust values, we do not place any such requirements on it. Thus we need the ability for the service host to request trust information from the trust manager and receive a timely reply.

- (*getTrustLocal*, *i*) allows the service host to request the trust score for a particular peer, *i*.
- (*trustReportLocal*, *i, T*) allows the trust manager to respond to a *getTrustLocal* request. *T* is an object that encapsulates the trust manager's trust in peer *i*. *T* is set to null in the case where the trust manager has no information regarding peer *i*.

Note on using this protocol. The functions specified at interface (2) above allow for either a "pull" or a "push" model for sharing trust information. Messaging is asynchronous for experience reports and recommendations.

"pull" model: The trust manager receives a *getTrust*, *getTrustResponseRequested*, or *getTrustAll* request for trust information about another peer, or all peers, and subsequently replies with a *trustReport* or *bulkTrustTeport* message. The sender of the request will need to use a timeout mechanism. In the case of a *getTrust* message, there is no onus on the recipient to reply. With *getTrustResponseRequested* and *getTrustAll*, the recipient should issue a reply even if this contains no trust information.

"push" model: The trust manager may be configured to spontaneously issue trust updates, either to other peers or to its local service host. This is typically for reasons of performance and efficiency. It is wasteful for peers to repeatedly poll each other unless there is useful new information. In the "push" case, each peer decides when and to whom to issue trust advertisements.

4.4.4 Model description

4.4.4.1 Assumptions

In this section, we present a model for using trust information to enhance security through collaboration in a rich multi-service decentralised environment. One of the problems of modelling decentralised systems like ad hoc networks is that it is unrealistic to take a top-down "bird's eye" view. Thus we model the set of peers as those peers of which a specific peer *is aware*.

We also make several assumptions. We assume that neighbour discovery and routing services are in place. We also assume that some kind of service registration and discovery is available to allow peers to reach an understanding of the set of services available and their associated trust thresholds. It should be noted that the assumption of a common threshold across all peers that provide the same service is something of a simplification. Even having all peers share an understanding of the relative meaning of trust threshold values is non-trivial.

We also assume authentication of identity. This could be done, for example, by having peers exchange public keys on their first interaction (in fact the public keys could be used as unique peer identifiers). Further messages between those peers could then be signed by the sending peer's corresponding private key so that the recipient could at least be confident that the sender is the same entity as that for which it has been building up a trust profile.

4.4.4.2 General model

Topology: Let $V_i = \{1,...,N_i\}$ be the set of peers of which peer *i* is aware. Some of these peers will be *adjacent* to *i*, normally by reason of network topology. This can be modelled by an adjacency vector, $A_i = (a_{i,j})_{j=1,...,N_i}$, where $a_{i,j} = 1$ if pair (i,j) are neighbours and $a_{i,j} = 0$ otherwise.

Services: Let $S = \{S_1,...,S_M\}$ be a set of services that may be provided. Each peer *j* provides a set of services $S^j \subset S$. Some peers will just be service consumers, so S^j will in those cases be empty.

Trust thresholds: Each service S_x has an associated *trust threshold* t_x, where $0 \le t_x \le 1$.

Representing Trust: We denote the local trust that peer *i* has in peer *j* as $T_{i,j}$. Each other peer $k \in V_i$ will maintain its own local view of trust in *j*, which may, as we shall see, influence the local trust of peer *i* in *j*. Note that $T_{i,j}$ is a trust *vector*. In the simplest case, this can just be a number, but such a number may be associated with other attributes that relate to it, such as confidence in the trust score or recency, or the service(s) to which it relates.

Trust initialisation: In the case where *i* has no prior knowledge of *j*, we will have $T_{i,j} = x$ where *x* is the *default trust*.

Trust decision: (using trust in service protection)

When peer *j* attempts to use service S_x provided by peer *i*:

- Service use is permitted if $f_x(T_{i,j}) > t_x$, where the function f_x maps trust vector $T_{i,j}$ onto a scalar number in the range (0,1)
- Otherwise peer *j* is blocked from using service S_x

Trust update following transaction:

After a transaction, by peer *j* on peer *i*:

- positive outcome: $T_{i,j}$ is increased according to some algorithm
- negative outcome: $T_{i,j}$ is reduced according to some algorithm

In general, following a transaction, we update $T_{i,j}$ according to:

$$T_{i,j} := f_e(T_{i,j}, E) \quad (1)$$

where *E* is a vector of attributes related to the transaction and f_e is a function defining how trust is updated based on transaction experience.

Trust update following referrals by a *j*, $T_{k,j}$. In general, following such a third party recommendation, we update $T_{i,j}$

Peer *i* may receive a message from a third party peer, *k*, indicating a level of trust peer *j*. This can be modelled as peer *i* adopting some of peer *k*'s trust level in peer

third party: according to:

$$T_{i,j} := f_e\left(T_{i,j}, T_{i,k}, T_{k,j}\right) \quad (2)$$

where f_r is a function defining how trust is updated. This trust transitivity depends on $T_{i,k}$, as peer *i* can be expected attach more weight to a referral from a highly trusted peer.

4.4.4.3 Some algorithms for updating trust parameters

The way trust is updated based on both experience and recommendations has a profound impact on the usefulness of this kind of overlay system. Note that peers are autonomous in our model and each might implement a different algorithm for updating and using trust. It can also be expected that a peer's behaviour in terms of handling trust may change if it is hijacked. Potential trust update algorithms include:

- *Moving average*: Advanced moving averages are possible, where old data is "remembered" using data reduction and layered windowing techniques.
- *Exponential average:* Exponential averaging is a natural way to update trust as recent experience is given greater weight than old values, and no memory is required in the system, making it more attractive than using a moving average.
- *No forgiveness*: This in a draconian policy where a peer behaving badly has its trust set to zero forever. Even more extreme is where a peer that is reported by a third party as behaving badly has its trust set to zero forever. This could perhaps be used if a particularly sensitive service is misused.
- *Second chance (generally, n^{th} chance)*: Draconian policies are generally not a good idea. IDS and other security systems are prone to false alarms. A variation on the "no forgiveness" approach is to allow some bounded number of misdemeanours.
- *Hard to gain trust; easy to lose it*: To discourage collusion, there is a case for making trust hard to gain and easy to lose.
- *Use of corroboration*: To prevent an attack by up to *k* colluding bad peers, we could require positive recommendations from at least *k*+1 different peers.
- *Use of trust threshold for accepting recommendations*: It is possible to model the ability to issue a recommendation as a kind of service usage on that receiving peer. Thus the receiving peer can apply a trust threshold to decide whether to accept that recommendation in the same way as any transaction attempt is adjudicated.

For our initial implementation, we update trust using an *exponential average*. As mentioned above, this is a natural way to update trust, as least where trust is simply represented using a scalar number between 0 and 1. Using an exponential average, we can tune the responsiveness of a node's trust determinations to new information received. Setting a high value for the exponential average parameter (close to 1) means that trust is heavily influenced by new data (compared with long-term experience). Setting a low value for the exponential average parameter (closer to 0) means that trust is just slightly influenced by each new experience or referral.

More specifically, we ***update trust based on direct experience*** according to:

$$T_{i,j} := \alpha E + (1-\alpha) T_{i,j} \quad (3)$$

where exponential average parameter α can be viewed as the *rate of adoption of trust*, $0 \le \alpha \le 1$. *E* is a binary attribute related to how the transaction is perceived (i.e. positively or negatively). Note that having α set to 0 means that the trust value is unaffected by the experience. Having α set to 1

means that local trust is always defined by the latest experience and no memory is retained. The higher the value of α, the greater the influence of recent experience of a node on the trust value recorded for that node. Lower values of α encourage stability of the system. If a succession of experiences of node *j* return the same binary trust evaluation, *E*, then the trust value $T_{i,j}$ converges towards *E* (towards 0 for a sequence of bad activity or towards 1 for repeated normal behaviour).

Peer *i* may receive a message from a third party, *k*, indicating a level of trust in peer *j*. This can be modelled as *i* adopting some of *k*'s trust level in *j*. As well as introducing a new parameter □ indicating the level of influence of recommender trust on local trust, we also use $T_{i,k}$, how much peer *i* trusts peer *k*. Specifically, we ***update trust based on referrals by a third party*** according to:

$$T_{i,j} = \beta T_{i,k} T_{k,j} + (1 - \beta T_{i,k}) T_{i,j} \ , \qquad (4)$$

where □is a parameter indicating the level of influence that recommender trust has on local trust, 0 ≤ □≤ 1. Note that, the larger the value of $T_{i,k}$, (i.e. the more *i* trusts *k*), the greater the influence of *k*'s trust in *j* on the newly updated value of $T_{i,j}$. Note that, if $T_{i,k} = 0$, (i.e. *i* has no trust in *k*), this causes $T_{i,j}$ to be unchanged.

Note that parameters α and □ can be varied as required to implement strategies mentioned above, such as:

- *Hard to gain trust; easy to lose it:* here α and □ are set to a lower value when trust is to be increased than for a decrease.
- *Use of trust threshold for accepting recommendations:* here □ is set to zero where trust in the referring node is less than the threshold (say 0.5, for example).

4.5 Trusted rating agencies

A rating agency offers a centralised alternative to peer-to-peer reputation systems. We propose a hybrid model of decentralised trusted rating agencies which cooperate with each other. This approach would better suite the nature of DEs.

In our model, a rating agency is an organisation trusted in the ecosystem. Since a DE is a purely decentralised model in which peers are considered equal, any peer in the ecosystem can take the role of a rating agency. The selection of rating agencies used by the participants is done in a natural and dynamic way based on the relations between participants, the characteristics of the institution, and trust relations between participants. In order to receive a certificate from a rating agency, users need to register themselves with the agency. The rating agency then collects information about transactions and recommendations from peers. After a transaction occurs, the agency gathers a proof of transaction and an assessment of the quality of the service provided signed by both parties.

The rating agency collects recommendations from different peers and aggregates them in order to compute a reputation score for each peer. The model on which reputation values are computed is complex and takes into consideration many factors like user profile and ecosystem context. The model also has a dynamic component based on learning and institutional trust.

Users can request a certificate from the agency at anytime. The certificate, signed with the private key of the agency, is an SPKI or X.509 certificate which contains the user's pseudonym, the reputation information, and a validity field. When interacting with another peer, the user can present this certificate. Users can also request information about other users from the agency.

In order to register with an agency and provide transaction feedback, users need incentives. First of all, the rating agency needs to be a trusted one such that certificates issued by it would be

considered reliable by partners. Users would register with a trusted agency to increase their reputation. A higher reputation means more transactions with trustworthy peers which increases the profit. In order to encourage users to use Rating Agencies and to cooperate in gathering information, the rating agency should consider in computing the reputation value the number of agencies with which the user is registered and how cooperative and active the user is in providing recommendations about services or knowledge.

Each ecosystem has one or several trusted rating agencies. Rating agencies in different ecosystems might use different measures (binary, probability, discrete values etc.). For interacting with peers from other systems, we use the same approach as for identity management. The unknown peer presents a certificate from its trusted rating agency. The peer gets redirected to the trusted agency of the user. The agency needs to have a trust relationship with the agency from the other ecosystem. After verifying the validity of the certificate, the agency issues and signs a new certificate. If agencies use different measures, a transformation of values is done. For example, if one user accepts only values between -1 and 1, he can request its trusted rating agency to make a transformation which also incorporates an ecosystem factor.

Figure 9 shows three interconnected DE. Each DE has a trusted rating agency inside the ecosystem (i.e. R.A.1, R.A.2, and R.A.3). Peers in the same DE are registered with the local agency which stores users' profiles and transaction ratings which are used to compute reputation values. The certificates issued by the agency are accepted in the local DE. If peer A wants to transact with peer B which belongs to another DE and has a different trusted rating agency (R.A. 3), then a translation of reputation needs to be made by R.A.3. In step 1, A gets a certificate from R.A.2. In step 2, A contacts B. Since B does not trust R.A.2, it forwards A to R.A.3. In step 3, A shows the certificate from R.A.2 to R.A.3. R.A.3 has a pre-existing trust relationship with R.A.2 and accepts and translates the certificate. It then issues a new certificate that is accepted by B (step 4).

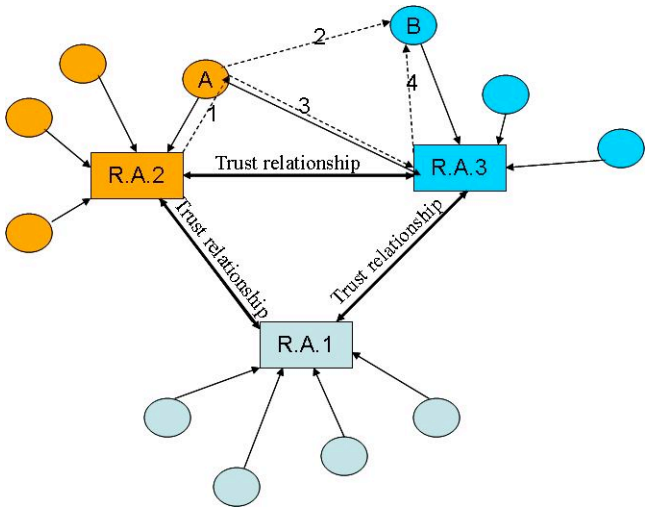

Figure 9: Trusted rating agencies model

Any peer can be a Rating Agency, a Credential Provider, a Server Provider, or a buyer. Peers can have one or several of these roles at the same time. It will often be the case that a Credential Provider is also a Rating Agency. It is up to the other peers in the system to decide which Credential Provider, Rating Agency, or Service Provider to trust and to use based on a reputation model.

The sustainability of the model is assured by the fact that the roles defined in the model are needed for the ecosystem to exist and function. Without CPs, users will not be able to interact in a secure way (e.g. authenticate each other) inside the DE. The lake of CPs will create bottlenecks in the

system and the users will not trust and use the system for business transactions. The trust model described is a democratic approach towards security. It is based on web2.0 and covers the dynamic needs of social and business networks. This democratic vision is assured by the fact that everyone can be a CP or RA and there will be no central authority. Business integrators and Regional Catalysts are the best candidates for assuming the roles of CP and RA. Both actors are in fact willing to invest in the approach to let the network grow and to provide new revenue streams in the region/cluster. In this way they will favour their own network and at the same time the overall ecosystem. Also Association of SMEs can cove this role on a voluntary base to facilitate business transactions in all different business fields. Some business actors can also provide these services and charge a fee from the users. Especially insurance companies and financial rating agencies could be interested in entering this new market in order to exploit new business opportunities in the online world.

In order to provide these services, actors need to invest in technological infrastructure and on internal organisation. The CP has the role of providing certificates to users. Though simple technological and infrastructural requirements are needed for issuing a certificate, to increase the security of the network, human verification of subjects might be necessary (i.e. users need to present themselves to the agency, requested documents need manual verification). The RAs are more complex since they need more ITC resources for running algorithms and storing data, and depend more on the human factor for verifying and issuing ratings.

A decentralised approach has the advantage that there is no single point of failure. The peer-to-peer network which interconnects the different actors of the DE implements a replication mechanism which makes resources available at all time, independent on the state of one single node. Moreover, users register themselves with different CPs and RAs from their ecosystem and from other ecosystems. To make their services and information available at all times, CPs and RAs replicate their data and services on different nodes such that at any point in time, a number of services will be up and running. If one RA or CP goes out of the DE permanently, the ecosystem will continue to function based on certificates issued by other CPs and RAs and the system.

5 Model evaluation, extension and future directions

5.1 Evolutionary trust

A natural ecosystem permanently evolves in state and time as a response to changing conditions without the intervention of human factors. In the same way, trust relations in a DE are established and evolve in a dynamic way. In particular, DE should provide a model for decentralised cross-domain trust relationship management.

When a new user or organisation joins the DE, a decentralised trust management model needs to facilitate the establishment of trust relationships with the new online community. Moreover, the model needs to provide support for organisations already active in a DE who take a role in another DE. Current security models for IT digital business only deal with trust relationships between entities already in the network and do not consider these cases.

In order to provide an evolutionary trust model, we will use the concept of *institutional trust* [51] and analyse user behaviour when dealing with digital institutions. New users can benefit from their affiliation to some institution when communicating with partners of the institution of with entities that trust the institution.

To provide evolution in the DE, a trust management model should additionally provide learning mechanisms based on institutional trust. An initial trust value can be obtained by examining trust relations between institutions to which the user belongs and institutions known by another peer. In this way, the platform is capable to adapt and evolve independently based on institutional trust. [48].

The evolutionary model that we propose combines learning mechanisms with reputation and social institutional trust. Some learning techniques can be found in [45, 15]. This new evolutionary approach can complement the concept of trust as already established in computer science literature.

5.2 Inter-ecosystem trust and learning

Based on history data and similarity functions, a Learning component could learn how to translate reputation values between institutions. For that matter, an analysis of trust scores obtained in different DEs by the same peer should be made. Rating agencies have a lot of historical data which can be used for learning how user ratings can be normalised.

Moreover, rating agencies could correlate user profile with behaviour. In this way it could be learnt that a certain type of user always gives smaller scores than another type, and that a certain type of user lies some times.

Learning can also be used to prevent malicious attacks. For example, the Learning component can discover patterns of collusion behaviour and other kinds of attacks from historical data and relate this with the user profile and user behaviour inside the system.

Strategies for frequency of exchange of trust information also need to be considered. Possible strategies include the following:

- Random destination and/or frequency for trust information passing
- On request
- Following a transaction
- On a significant change in trust (e.g. a reduction)

- Each node specified its own trust receiving preferences.

Inter-ecosystem trust just makes sense, of course, where the boundary of an ecosystem can be defined. It is outside the scope of this document to define how an ecosystem is bounded, but we can imagine that such a boundary might be based on such things as physical geography, common services or interests, or any characteristics that might define a community. It is reasonable to assume that these ecosystems might overlap in some ways, and that entities may participate in multiple ecosystems. Future development of our trust model must take this into account.

5.3 Relation to work on distributed accountability

It is planned to more closely integrate this work with parallel work in OPAALS on distributed accountability.

In particular, we plan to consider the subtle trust relationships that exist between service providers that come together provide a service. These service providers may have little in the way of formal relationships with one another, and may even be competitors in some markets. When a provider delivers a specialist service that forms part of a composed overall service, it expects the overall service to be supplied to the user at the appropriate quality level. Firstly, the user can only expect to pay if satisfied with the overall service (e.g. video is not much use without audio) and, secondly, the specialist service provider will want to protect its reputation, brand image, etc.

Thus, non-performing collaborating service providers can be seen as a threat. We plan to use our distributed trust approach to assist each service provider in making decisions to enter such collaborative arrangements. Each service provider will record its own experience of other service providers together with the shared experience of third party service providers, and use this. Rather than having the service provider make a binary yes/no decision, the trust system can dictate the price advertised for the service (in a competitive market, setting a really high price is of course equivalent to declining service). This approach is similar to the way a risk premium is assessed in finance – loans that are perceived as of higher risk are typically charged a higher interest rate.

5.4 Simulations

Experiments are required to explore the dynamics of the trust models proposed as they are refined. It is intended to evaluate convergence of trust from initial values to “true” values and stability of these values, as well as the performance and scalability of various approaches taken. Further experiments are required to explore the effects of peer mobility, changes in behaviour, and network topology. It is also necessary to evaluate robustness of the system in the face of attempts to corrupt it, including collusion between malicious entities (issuing false ratings about each other, for example).

6 Concluding remarks

In this deliverable, we have proposed an evolutionary trust model for Digital Ecosystems. The model is suitable for the decentralised and dynamic nature of DEs. We base our model on standards and consider scalability, availability and ease of putting into practice as the most important characteristics of the model. The model has four main components: distributed identity management, peer-to-peer reputation, trusted rating agencies, and learning. These components address different security, social and evolutionary aspects of the DE and create a complete model. In the context of the deliverable we are not able to cover all the elements of the new models. Some aspects, analysis and studies are still on going and will be available in an updated version of this deliverable. Experiments and simulations are also required to evaluate convergence of trust towards stability, and to test performance and scalability of various approaches taken. It is also necessary to evaluate robustness of the system in the face of attempts to corrupt it, including collusion between malicious entities (issuing false ratings about each other, for example). We are not sure to cover all those research activities in the next deliverable foreseen in December 2007. Some research activities could, in fact, be postponed in phase two of the NoE.

7 References

[1] Aitken, D., Bligh, J., Callanan, O., Corcoran, D. and Tobin, J., , "Peer-to-Peer Technologies and Protocols", (2001), at: http://ntrg.cs.tcd.ie/undergrad/4ba2.02/p2p/ (last access: 02/11/06).

[2] Aberer, K. and Hauswirth, M., "An Overview on Peer-to-Peer Information Systems", Workshop on Distributed Data and Structures (WDAS-2002), 2002 - minoas.di.uoa.gr

[3] Albert R., Jeong H., and Barabási, A., "Attak and tolerance in complex networks", Nature 406 378, 2000.

[4] Adamic L. A., "The small World Web." Third European Conference on Research and Advanced Technology for Digital Libraries (ECDL 99); 1999 September 22-24; Paris, France. NY: Springer Verlag; 1999; Lecture Notes in Computer Science; 1696: 443-45.

[5] Briscoe G., Sadelin S. and Paperin G. "Biology of Applied Digital Ecosystems".

[6] Levin S., "Ecosystems and the biosphere as complex adaptive systems", *Ecosystems*, vol. I, pp.431-436, 1998.

[7] Swamynathan G., Zhao B. Y. and Almeroth K. C., "Decoupling service and feedback trust in a peer-to-peer reputation system" (2005). Parallel and Distributed Processing and Applications - ISPA 2005 Workshops. 3759, pages. 82-90. Postprint available free at: http://repositories.cdlib.org/postprints/2083.

[8] Buchegger S., and Boudec J. L., "A robust reputation system for P2P and mobile ad-hoc networks". *In Proc. of the 2nd P2PEcon Worksho,* June 2004.

[9] Dellarocas C., "Immunizing online reputation reporting systems against unfair ratings and discriminatory behaviour". In EC '00: *Proceedings of the 2nd ACM conference on Electronic commerce*, p. 150-157. ACM Press, 2000.

[10] Swamynathan G., "Reputation Management in Decentralized Networks. Major Area Examination: Report". 2005. http://www.cs.ucsb.edu/~gayatri/MAE-Documents/mae-report.pdf.

[11] Marti S. and Garcia-Molina H., "Taxonomy of Trust: Categorizing P2P Reputation Systems," 2005.

[12] Singh A. and Liu L., "TrustMe: Anonymous Management of Trust Relationships in Decentralized P2P Systems", in: *IEEE 3rd International Conference on Peer-to-Peer Computing (P2P 2003)*, 2003.

[13] Dewan P. and Dasgupta P.. "PRIDE: Peer-to-peer Reputation Infrastructure for Decentralized Dnvironments". In *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, p. 480-481. ACM Press, 2004.

[14] Damiani E., D. C. di Vimercati S., Paraboschi S., Samarati P., and F. Violante. "A Reputation-Based Approach for Choosing reliable resources in peer-to-peer networks". In *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*, p. 207-216. ACM Press, 2002.

[15] Tan M., Multi-agent reinforcement learning: Independent vs. cooperative learning. In Huhns, M. N. and Singh, M. P. (eds), *Readings in Agents*, pages 487-494. Morgan Kaufmann, San Francisco, CA, USA, 1997.

[16] Zhou R. and Hwang K., "PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing", IEEE *Trans. on Parallel and Distributed Systems, Vol. 18, No.5*, 2006.

[17] Zhou R. and Hwang K., "Trusted Overlay Networks for Global Reputation Aggregation in P2P Grid Computing", the *20th IEEE International Parallel & Distributed Processing Symposium (IPDPS'06)*, Greece, April 25-29, 2006.

[18] Kamvar S.D., Schlosser M. T. and Garcia-Molina H.. (2002) "The EigenTrust Algorithm for Reputation Management in P2P Networks".

[19] Xiong L. and Liu L., "A Reputation-Based Trust Model for Peer-to-Peer eCommerce Communities". In *IEEE Conference on E-Commerce (CEC'03)*, Newport Beach, USA, June, 2003.

[20] Chen R. and Yeager W., Sun Microsystems. "Poblano: A Distributed Trust Model for Peer-to-Peer Networks".

[21] Ebay Online Marketplace: http://www.ebay.com/.

[22] O'Hara K., Alani H., Kalfoglou Y. and Shadbolt N., "Trust Strategies for the Semantic Web", in: *ISWC'04 Workshop on Trust, Security and Reputation on the Semantic Web*, 2004.

[23] Page L., Brin S., Motwani R., Winograd T., "The PageRank citation ranking: Bringing order to the web", Tech. rep., Stanford Digital Library Technologies Project (1998).

[24] Feldman M., Lai K., Stoica I., Chuang J., "Robust Incentive Techniques for Peer-to-Peer Networks", in: *ACM Conference on Electronic Commerce (EC'04)*, 2004.
[25] Feldman M., Padimitriou C., Chuang J., Stoica I., "Free-Riding and Whitewashing in Peer-to-Peer Systems", in: *ACM SIGCOMM 2004, Workshop of Practice and Theory of Incentives and Game Theory in Networked Systems*, 2004.
[26] Gnutella. The Gnutella protocol specification v0.4, 2001. Available at http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf.
[27] Falcone R. and Castelfranchi C., "Social trust: a cognitive approach", In *C. Castelfranchi, Y.-H. Tan (eds.), Trust and Deception in Virtual Societies*, pp. 55-90, Kluwer, 2001.
[28] Dingledine R., Freedman M., and Molnar D., "Accountability measures for peer-to-peer systems", In *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*, O'Reilly, 2000.
[29] Douceur J., "The Sybil attack," *Proc. Int'l Workshop on Peer-to-Peer Systems*, March 2002.
[30] Jøsang A., Ismail R., and Boyd C., "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, pp 618-644, March 2007.
[31] Ruohomaa S. and Kutvonen L., "Trust management survey", *Proc. 3rd International Conference on Trust Management (iTrust),* LNCS 3477, May 2005.
[32] Li H. and Singhal M., "Trust management in distributed systems," *IEEE Computer*, pp 45-53, February 2007.
[33] Grandison T., "Conceptions of Trust: Definition, Constructs and Models," In *R. Song (ed.), Trust in E-Services: Technologies, Practices and Challenges,* IGI Global, 2007.
[34] Gambetta D., "Can we trust trust?" *D. Gambetta (ed.), Trust: making and breaking cooperative relations*, pp 213-237, Blackwell, 1988.
[35] McKnight D. and Chervany N., "The Meanings of Trust," Technical Report, MISRC Working Paper Series 96-04, University of Minnesota, Management Information Systems.Research Center, 1996.
[36] Solhaug B., Elgesem D. and Stølen K., "Why Trust is not proportional to Risk," *Proc. 2nd International Conference on Availability, Reliability and Security (ARES),* pp 11-18, Vienna, April 2007.
[37] Mui L., Mohtashemi M., Halberstadt A., "A Computational Model of Trust and Reputation," Proc. 35th Hawaii International Conference on System Science (HICSS), 2002.
[38] Blaze M., Feigenbaum J., and J. Lacy "Decentralized Trust Management", Proc. IEEE Symposium on Security and Privacy, pp 164-173, Oakland, 1996.
[39] Lin C. and Varadharajan V., "A Hybrid Trust Model for Enhancing Security in Distributed Systems", *Proc 2nd International Conference on Availability, Reliability and Security (ARES)*, pp 35-42, Vienna, April, 2007.
[40] Winsborough W., Seamons K., and Jones V., "Automated Trust Negotiation", *Proc. DARPA Information Survivability Conference and Exposition (DISCEX '00)*, pp 88-102, January 2000.
[41] Kamvar S., Schlosser M., and Garcia-Molina H., "The EigenTrust algorithm for reputation management in P2P networks", *Proc. 12th International World Wide Web Conference*, Budapest, May 2003.
[42] Castelfranchi C., "Why we need a non-reductionist approach to trust", *Proc. 4th International Conference on Trust Management (iTrust),* LNCS 3986, Pisa, May 2006.
[43] Bateson P., "The biological evolution of cooperation and trust", in *D. Gambetta (ed.), Trust: making and breaking cooperative relations,* pp 14-30, Basil Blackwell, 1988.
[44] Burke R., "Hybrid Recommender Systems: Survey and Experiments", *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, November, pp. 331-370, 2002.
[45] Resnick P., Kuwabara K., Zeckhauser R., and Friedman E.. "Reputation systems". *Communications of the ACM*, 2000.
[46] Koshutanski H., Ion M., and Telesca L. (2007) "Distributed Identity Management Model for Digital Ecosystems". In *Proceedings of International Conference on Emerging Security Information, Systems and Technologies (SECURWARE'07)*, (October), Valencia, Spain, IEEE press.
[47] Telesca L. and Koshutanski H. (2007) "A Trusted Negotiation Environment for Digital Ecosystems". In *F. Nachira, M. Le Louarn, L. Rivera Lèon (eds) Building the foundations of Digital Ecosystems: FP6 Results and Perspectives*. Publisher: European Commission.
[48] McGibney J. and Botvich D., "A trust overlay architecture and protocol for enhanced protection against spam", *Proc. 2nd International Conference on Availability, Reliability and Security*, April 2007.

[49] Dam M. and Stadler R., "A generic protocol for network state aggregation", *Proc. Radio Science and Communication conference (RVK)*, Linköping, 2005.
[50] Buragohain C., Agrawal D., Suri S., "A game theoretic framework for incentives in P2P systems", *Proc 3rd International Conf on Peer-to-Peer Computing (P2P 2003)*, Sept 2003.
[51] Pavlou P. A., Tan Y.-H. and Gefen D., "The transitional role of institutional trust in online interorganizational relationships". In *Proceedings of the 36th Hawaii International Conference on System Sciences*. Jan 2003, IEEE Press.
[52] Levich R. M., Majnoni G. and Reinhart C. M. (ed), 2002. *Ratings, Rating Agencies and the Global Financial Systems*. Kluver Academic Publishers.
[53] Timothy J. Sinclair, *The New Masters of Capital: American Bond Rating Agencies and the Politics of Creditworthiness* (Ithaca, NY: Cornell University Press, 2005).
[54] Sandage S. A., "*Born Losers: A History of Failure in America"*, Harvard University Press, 2005, chapters 4-6.
[55] IOSCO Report on "Activities of Credit Rating Agencies", September 2003: http://www.iosco.org/library/pubdocs/pdf/IOSCOPD153.pdf.
[56] DBE Deliverables n. 32.1 on "DBE Legal Analysis (public and private regulatory framework) and Taxonomy" and n. 32.2 on "Benchmark and Analysis of 'Legal ICTs'", at http://www.digital-ecosystem.org/ (last access 15.08.07).
[57] Nachira, F. (2002) "Towards a Network of Digital Business Ecosystems Fostering the Local Development". Discussion Paper. http://www.digital-ecosystems.org/ (last access 15.08.07).
[58] Preece J., "Online Communities: Designing Usability, Supporting Sociability". Chichester, UK: John Wiley & Sons, 2000.
[59] Mill J. S., "On the Definition of Political Economy; and on the Method of Investigation Proper to It," London and Westminster Review, October 1836. Essays on Some Unsettled Questions of Political Economy, 2nd ed. London: Longmans, Green, Reader & Dyer, 1874, essay 5, paragraphs 38 and 48.
[60] Persky J., "Retrospectives: The Ethology of Homo Economicus." The Journal of Economic Perspectives, Vol. 9, No. 2 (Spring, 1995), pp. 221-23.
[61] Hollis M., "Trust Within Reason". Cambridge University Press, 1998.
[62] Simon H. A., "Models of Bounded Rationality", Vols. 1 and 2. MIT Press, 1982.
[63] Luhmann N., "Trust and Power". Chichester: Wiley, 1979.
[64] Cummings L. L. & Bromiley, P., "The organizational trust inventory: development and validation". In R. Kramer & T. Tyler (Eds.), Trust in organizations: Frontiers of Theory and Research. Thousand Oaks, CA, Sage Publications, 1995.