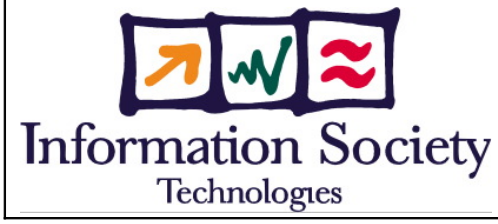
OPAALS PROJECT

Contract n° IST-034824

WP3: Autopoietic P2P Networks

Del3.9 – Final Identity and Trust Models



Project funded by the European Community under the "Information Society Technology" Programme

Contract Number: IST-034824

Project Acronym: OPAALS

Deliverable N°: D3.9

Due date: M29

Delivery Date: M33

Short Description:

This deliverable provides a theoretical summary of identity in a DE and an update of our identity model, based on further research and feedback from the implementation process.

We outline how identity is built on trust, and how identity and trust are mutually generative, which anticipates the integration of the identity and trust models.

We give the latest state of identity operations, which form the basis for the implementation work in to be reported in D3.11. We also give an a pertinent use case of a sign-on operation for a JXTA environment.

We further develop the rating agencies model which enables entities to establish trust relations based on objective and verifiable data. In this way we provide a more complex trust model for Des.

We discuss algorithms for the evaluation of trust for both direct experience based trust as well as trust based on referrals.

We provide integration points for the trust model with the Distributed Accountability and Distributed Transactions models.

We provide scenarios to show how an implementation of our Trust Model is used to evolve trust in both a transaction context and in a Rating Agencies context. model already developed to encode and exchange trust claims. This allows to take advantage of the existing security mechanisms and shortens adoption and deployment time and costs.

Author: Mark McLaughlin (WIT), Paul Malone (WIT), Mihaela Ion (CN) , Jimmy McGibney(WIT), Dmitri Botvich(WIT)

Partners contributed: CN, WIT

Made available to: Public

Versioning		
Version	**Date**	**Name, organization**
V 0.1	01/10/08	Mihaela Ion (CN)
V 0.2	05/12/08	Mark McLaughlin (WIT)
V 0.3	13/01/09	Paul Malone (WIT), Mark McLaughlin (WIT)
V 0.4	05/02/09	Jimmy McGibney (WIT), Dmitri Botvich (WIT)
V 0.5	16/03/09	Mark McLaughlin (WIT), Paul Malone (WIT), Mihaela Ion (CN),
V 1.0	24/03/09	Paul Malone (WIT), Mark McLaughlin (WIT)

Quality check

Internal Reviewers: Jo Stanley (CAM), Ossi Nykänen (TUT)

Dependencies:

Achievements	We provide a theoretical summary of identity in a DE and an update of our identity model, based on further research and feedback from the implementation process. We define how identity is built on trust, and how identity and trust are mutually generative, which anticipates the integration of the identity and trust models. We give the latest state of identity operations, which form the basis for the implementation work in to be reported in D3.11. We also give an a pertinent use case of a sign-on operation for a JXTA environment. We further develop the rating agencies model which enables entities to establish trust relations based on objective and verifiable data. In this way we provide a more complex trust model for Des. We discuss algorithms for the evaluation of trust for trust based on direct experience as well as trust based on referrals. We provide concrete integration points with the Distributed Accountability and Distributed Transactions models. We provide scenarios to show how an implementation of our Trust Model is used to evolve trust in both a transaction context and in a Rating Agencies context.
Work Packages	The work contributes to the provision of integrating Identity and Trust in the developing peer-to-peer platform via WP5. This is achieved by the specification of an identity model and a trust model together with algorithms and design choices for integration. Identity and Trust Models also provide a contribution to the development of the OKS in WP10. There has been much discussion between the partners on the application of these models in the implementation of the OKS. The work also provides a discussion point for WP 12 where a socio-economic framework for Identity, Trust and Accountability is being addressed.
Partners	IPTI (platform development), TechIdeas (integration), Surrey (p2p platform),
Domains	Identity, Trust, Distributed Computing, Security
Targets	Other Researchers, System Implementers, SMEs, Public Administrators,

	Social Scientists
Publications*	The initial Identity Model was published in Koshutanski, H., Ion, M. and Telesca, L., 2007, *A distributed identity management model for digital ecosystems*, in Proceedings of International Conference on Emerging Security Information, Systems and Technologies (SECURWARE'07), IEEE Press The Trust Overlay Model was published in McGibney, J. & Botvich, D., 2007. A Trust Overlay Architecture and Protocol for Enhanced Protection against Spam. In Proceedings of *The Second International Conference on Availability, Reliability and Security*. IEEE Computer Society, pp. 749-756. and McGibney, J. & Botvich, D., 2008. A trust based system for enhanced spam filtering. *Journal of Software*, 3(5), 55-64. The Trust Algorithms are published in McGibney, J. and Botvich, D., 2007, "Distributed dynamic protection of services on ad hoc and p2p networks", in *Proceedings of 7th IEEE International Workshop on IP Operations and Management (IPOM)*, San Jose, CA, USA, Lecture Notes in Computer Science (LNCS) 4786, pp 95-106, Springer, November 200
PhD Students*	N/A
Outstanding features*	The development of the theory of identity and trust in digital ecosystems represents a number of moderate advances from the state of the art in distributed identity, user-centric identity, identity federation and computational trust. The trust overlay approach represents an incremental advance to the state of the art in trust evaluation for entity-centric distributed systems. The algorithms for trust evaluation provides a similar advancement in extracting trust values from reports of dependability and reliability. The use of a modeling framework to build generic identity protocols (operations), with the possibility of multiple, re-usable bindings and integration with the trust, represents a significant advance in the state of the art beyond a SAML-inspired, identity federation approach. The ongoing work on relative naming and identity, including a URI scheme and piecewise, potentially privacy preserving URI resolution, coupled with an encoding of entity-centric trust relationships into URIs, has the potential for a significant advance in a combined theory of identity, trust and naming in digital ecosystems and decentralised

	environments as a whole. The work performed on identity and trust by the partners in OPAALS has been published to communities outside the digital ecosystems community (See Publications above).
Disciplinary domains of authors*	Mark McLaughlin, WIT, Computer Science Mihaela Ion, CN, Computer Science Paul Malone, WIT, Computer Science Jimmy McGibney, WIT, Computer Science Dmitri Botvich, WIT, Computer Science

Contents

1 Introduction

In D4.1 we proposed an identity model for a Digital Ecosystem (DE) based on several components:

1. An initial identity model and default protocol communication scheme that uses federation semantics and techniques to perform Single Sign-On in a DE, where entities sometimes compete and sometimes collaborate and form unstable coalitions (published in [2]).
2. A pure-SAML, extensible, operation modelling framework was introduced, capable of implementing (1), and verifying other identity claims, using generic components such as Actors, Connections and SAML-like Profiles and Bindings. This framework uses binding implementations to ensure interoperability in heterogeneous environments.
3. An initial integration of identity and trust models, where entities consult trust managers to ascertain their trust in other entities participating in an operation. In order for an operation to succeed, certain trust relationships between the actors, representing the entities involved, must be sufficient.

In D4.3 we proposed a trust model for a DE based on several components:

1. A distributed identity management model that authenticates the entities of the DE. Secure authentication of an entity is a first requirement for building trust in online environments like DEs. In D4.1 we provided a detailed description of the model that allows authenticating entities across domains. An implementation of the model based on SAML will be provided in D3.11. We chose SAML because it achieves interoperability in heterogeneous systems and cross-domain authentication through single-sing on (SSO).
2. A peer-to-peer reputation model which allows entities of the DE to express recommendations about their neighbours or known peers. The recommendations are then aggregated into a local trust value. The details of the model can be found in D.4.3.
3. A rating agencies model in which specialized institutions asses the trustworthiness of entities based on “hard” data like measurements, testing, and expert opinion.
4. An evolutionary component. Trust relations in DEs get created, evolve and die, similar to the entities that create them. It is essential to provide this kind of behaviour in our models. We try to achieve a level of evolution in each of the previous three components. The

simplest way to achieve that is through dynamic trust relations that change constantly when the conditions change (e.g., an entity dies, an entity starts misbehaving, new entities are introduced to the system, services improve their quality, data gets outdated etc.).

In this deliverable we provide the following extensions to the models developed in Phase I:

1. We provide a theoretical summary of identity in a DE and an update of our identity model, based on further research and feedback from the implementation process.
2. We outline how identity is built on trust, and how identity and trust are mutually generative, which anticipates the integration of the identity and trust models.
3. We give the latest state of identity operations, which form the basis for the implementation work in to be reported in D3.11. We also give an a pertinent use case of a sign-on operation for a JXTA environment.
4. We further develop the rating agencies model which enables entities to establish trust relations based on objective and verifiable data. In this way we provide a more complex trust model for Des.
5. We discuss algorithms for the evaluation of trust for trust based on direct experience as well as trust based on referrals.
6. We provide concrete integration points with the Distributed Accountability and Distributed Transactions models.
7. We provide scenarios to show how an implementation of our Trust Model is used to evolve trust in both a transaction context and in a Rating Agencies context.

2 Distributed Identity Model

The identity model outlined here follows on from that presented in D4.1. We update the theory of identity in a DE and show how the model has evolved towards implementation. We will talk of identity with regard to entities in a DE, where entities encompass all agents that are capable of acting independently in a DE. (Other objects may have identifiers and attributes but they are not said to have an identity as such.)

During the course of our work, it became apparent that the structure of DEs pose a number of unique challenges to the way we approach identity. The requirement for *no single point of failure or control* causes us to dispense with the notion of centralised identity provision and centralised identity authorities. This means that the global uniqueness of identities cannot be assumed. However, taking this requirement to its logical conclusion, we must also call the use of centralised naming registries into question. Ultimately, the ubiquitous Domain Name Service (DNS), though decentralised architecturally, is subject to a central authority (ICANN). Although it would be difficult to remove the use of DNS entirely from a DE, particularly at first, where legacy technologies may play a significant role, we must ensure that centralised naming registries do not underpin the DE architecture. Since all interactions between elements in a DE rely on identity and naming, it was considered a priority to thoroughly develop the theory and practice of applying identity in a DE.

2.1 Identity in Digital Ecosystems

In order to avoid central servers or authorities for naming and identity, we formulated a theory of identity based on trust, utilising small world relationships to form local contexts in which identities (and naming) can be considered unique. Naming and identity must be unique in order to support a range of applications where unique addressing of entities is paramount, and also in order to pursue the general strategy of RESTful naming in the OPAALS DE environment. However, since we cannot guarantee global uniqueness, we attempt to build the DE from the bottom up, as the set of entities involved in contextual networks of trust, or alternatively, as the overlapping union of identity contexts. The concept of an identity context tallies well with user-centric principles, which expects that users will use different (partial) identities in different contexts. We define an identity context as a grouping of entities that identify themselves for a particular purpose (often for a fixed period). Therefore, the 'local' view of identity not only provides maximum independence from

central points of control or failure but also forms logical contexts in which an entity might want to use a given partial identity (rather than a single global identity). It is hoped that local identities that are not expected to apply outside of a given context will not lend themselves easily to being matched to other identities that an entity might use, or to be traced back to a global real world identity (where that is not the specific intention). Hence this infrastructure should also be privacy preserving.

Another important requirement of a DE is that it must be a platform on which business transactions can be conducted. Business transactions are of course subject to the laws of the jurisdiction in which the transaction is said to occur. Legal systems themselves constitutute single points of failure; however, we must allow for the use of centralised naming and identity, even if we must not be reliant on them, in order to support these transactions. For example, if an entity wishes to use a CA cert to assert its identity, which in turn makes use of DNS, then we must allow for this.

2.1.1 Relative naming

A RESTful, relative naming scheme has been proposed for the OPAALS DE. Relative naming guarantees unique naming across an identity context, which does not require naming authorities or even global naming. The resolution of these names can be distributed, in that the name can be partially resolved by a chain of entities. An example of such a RESTful, relative URI, which can be resolved in a distributed fashion, is as follows:

http://john.mark.alex/resource

where 'resource' is what is being named, and this resource is associated with john, who is known directly (by this unique name) to mark who is known directly (by this unique name) to alex. These kind of URIs are valid only from the perspective of a given entity, since they are relative to that entity. In the example above, the party relying on the resolution can resolve alex, who in turn can resolve mark, who in turn can resolve john.

Since each entity holds a unique reference to all its direct contacts, and since each URI is composed of a chain of these unique references, the URI itself is unique (from the perspective of the relying party). This naming scheme admits a number of possibilities for resolving a URI to a contactable address. Connections can be made directly, or via a relay between the entities in the chain to provide enhanced privacy. Each entity in the chain can potentially be addressed using a different

addressing mechanism. For example, one URI might resolve to a URL, another might resolve to an address in another defined protocol.

For the purpose of conducting business, there is nothing to prevent a relative URI being mapped to a real world identity, asserted by a legal authority. The only caveat is that each such mapping increases the DEs dependence on authorities and therefore single points of failure.

The usage and conceptualisation of relative naming is still in active development and there are some outstanding issues, such as how resources associated with entities are persisted in the OPAALS DE distributed storage layer, which is global namespace.

2.1.2 Update of Identity Model and the Operation Model

Locality in Identity Contexts

We define 'locality' in identity contexts by the following requirements:

1. The number of entities participating in the context should be 'small'.
2. The scope of the context should be concisely and narrowly defined (delimited by purpose and time).
3. Identities should be (as far as possible) unique or pseudo-unique[1] in an identity context.

The lack of a central registry and requirement 3, enforce requirements 1 and 2, since only by making the context 'local' can naming and identity be (effectively) unique. This is important since many applications assume uniqueness of identity.

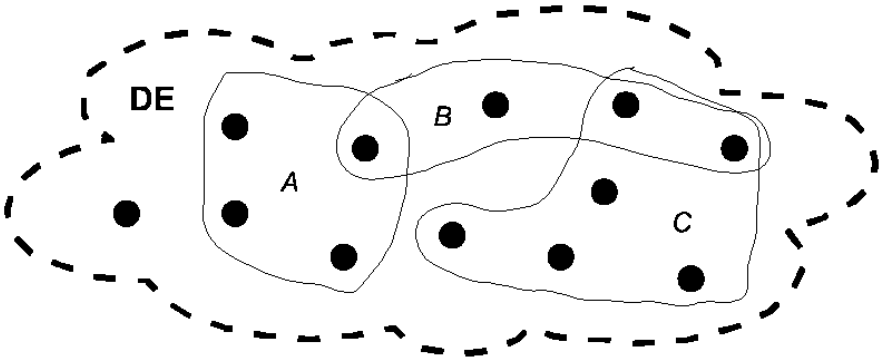


Figure 2.1: Entities participating in a number of identity contexts in a DE

1 Pseudo unique identity derives from an identifier which was generated from a pseudo random seed, or from a seed (from a very large 'seed space') that is very unlikely to occur more than once.

Figure 2.1 shows entities in a DE participating in identity contexts. Identities only apply in individual identity contexts, therefore entities do not have have a single canonical identity but a set of identities. The naming of A, B and C in Figure 2.1 is purely symbolic: actually identity contexts are not named explicitly, but emerge and evolve as the scope is constructed (see below).

It must also be noted that keeping the scope small and narrowly defined is an important aspiration of identity in a DE, but it may be difficult to avoid a scope that is large and poorly defined, since the restriction is not enforced technically. Locality represents an aspiration that identity contexts do not become dominant in a DE, and a guide for methods of constructing the scope technically (below).

Construction of the Scope

We identified two mechanisms that combine to determine the scope of identity contexts:

1) Locally coordinated Identity: specifies the set of entities that share an IdP for a certain identity they hold, which trust the asserted identity of the other entities in the set implicity.

2) Trust transitive networks: specifies the interconnected network of entities whose identities are trusted (sufficiently, since trust levels may be graded) by other entities with whom it has a trust relationship.

In (1), there is a single, local identity provider, or co-ordinator, that provides identity for entities. It is important that the IdP provides identity for a ‘local’ portion of the identity context, so that participant numbers remain relatively small (since otherwise the IdP would be a single point of failure). In (2), a web of trust using contextual trust transitivity, with trust ratings indicating a (directional) measure of trust between individual entities in the context of identity referral and/or provision, can be developed. Trust values can be made to vary dynamically based on referral or experience[1]. Trust overlays[1], measuring trust using individual experience or referral trust can be utlised. These two mechanisms are illustrated in Figure 2.2.

Both of these mechanisms combine to define the set of entities in the scope of identity contexts. Many entities belong to locally co-ordinated identity groups and when these entities also accept referrals from other entities, potentially other IdPs, we can imagine a graph of trust relationships, some implicit and absolute, others explicit and graded, which defines the scope of the context. The identity context also represents a reference frame in which entities wish to be identified using a particular identity in order to participate in a given type of activity. Therefore, the scope is determined by the activity in question, is a frame of reference in which entities identify themselves, and is realised using mechanisms (1) and (2). An example of the scope of an identity context in a

sign-on operation is given in Figure 2.4.

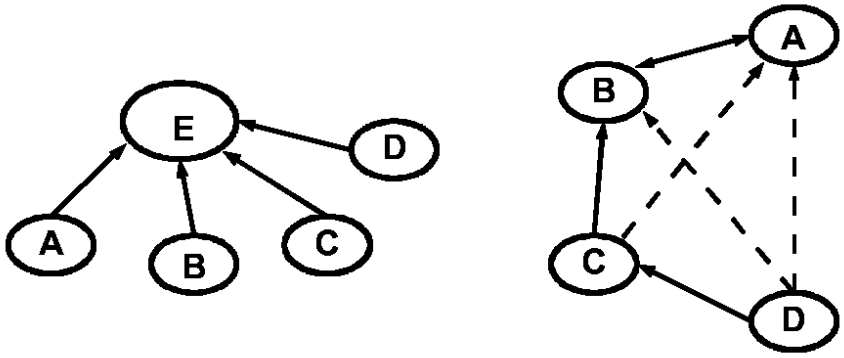


Figure 2.2: Locally co-ordinator identity (left), and trust transitive networks (right).

Via the two mechanisms outlined, the scope of the identity context can be said to emerge organically from a dynamically inter-connected network of trust relationships. It may therefore be impossible or infeasible to enumerate the entities in an identity context.

The Identity Model

The identity model specifies the role elements in a DE play in establishing identity in an identity context. Following on from the development in [2], we outline the basic actors and elements in identity interactions. We define an identity operation (see D4.1 and section 2.3) as a contingent network protocol that verifies or asserts a claim, e.g. To verify that the 'speaker' has the identity that was claimed.

1. Subject: Often User or User Agent. The subject of the identity operation.
2. Relying Party (RP): Often Service Provider. The party that relies on the result of an identity operation.
3. Identity Provider (IdP): Credential Provider in [2]. The party that is asserting a claim in an identity operation.
4. Identity Context: The context in which the above actors interact and in which their identities are valid and pseudo-unique.
5. Digital Ecosystem: The DE. The set of all interrelated entities in the ecosystem. Entities behave as one of the above actors during operations in DEs.

We use generic actor designations to recognise the fact that actors can interact for many reasons to complete any operation, hence ‘subject’ rather than ‘user’ and ‘relying party’ rather than ‘service provider’. Entities in a DE may play the role of different actors at different times and in different contexts.

The Operation Model

The Operation Model is a meta-model for operations. The identity model, above, orientates the operation model around identity operations.

1. Actor: A role that an entity plays in an operation. (The common actors, Subject, RP and IdP, are given in the identity model.)
2. Connection: A relation between two actors that also provides an abstraction of a uni-directional communication between them.
3. Profile: A scheme that dictates how a portion of a protocol is conducted, by specifying a contingent ordering of connections.
4. Binding: The transport definition and logic adopted by connections in a given pro le.
5. Operation: A specification of an operation, including the profile(s) required to conduct it.

The use of profiles is inspired by the SAML v2.0 specifications. We require an operation model rather than a prescribed set of profiles since DE environments are dynamic and heterogeneous, rather than a collection of stable federations. For example, complicated interactions may be required in order to produce a result. e.g. An RP might require three separate IdPs to vouch for the identity of a subject.

2.2 Identity backed by Trust

Operations specify how actors should communicate to perform an identity related task. Trust can be built into operations by requiring redundant consent or assent from other parties, where the input of these parties would not be required in a ‘trusting environment’[2]. Trust can also be evaluated dynamically between participants during an operation, which we can mandate must be sufficient in order for the operation to succeed. The entities involved in an operation are a subset of those in an identity context.

2 Where it is assumed that all entities trust each other completely.

One of the variables in the connection relation can refer to a trust threshold. If trust levels between the two actors are lower than the threshold, the communication delivered by the connection can be considered untrusted, and the operation can either fail or be deemed unsuccessful. Alternatively, or in addition to this mechanism, trust transitivity can be used to aggregate the trust between actors participating in an operation to produce an overall score, which must exceed a certain minimum trust threshold for the operation to be considered successful.

2.2.1 Formal Description

We begin by explicitly describing trust in a decentralised environment, such as a DE. Trust has been formulated similarly in [3]. Let P be the set of all entities on a network and C the set of all trust contexts. Let T be the set of all directed trust relationship evaluations between any two entities $p_x, p_y \in P$, in a given trust context $c \in C$.

We define a function k that gives the evaluation of the trust places in p_y in context c, determined by the triple (p_x, p_y, c), such that

$$t_{x,y} = k(p_x, p_y, c), k : P \times P \times C \rightarrow T \quad (1)$$

where $t_{x,y} \in T$. The function k represents some computation that calculates p_x trust based on referrak and/or experience reports, performed by an impartial system or trust overlay. Trust can also be deemed to be absolute.

2.2.2 Using Trust to Establish the Identity of a Subject

Trust forms the basis of identity. Here, we show how trust relationships are used to obtain and verify a subject's identity, which forms the basis of the important sign-on operation (see section 2.3.1).

In order for the identity of a certain subject to be established by an identity provider to the satisfaction of a relying party, pre-existing trust relationships are exploited. We extend (1) as follows.

Let P_s be the set of all identifiable subjects, P_r be the set of all relying parties and P_i be the set of

all IdPs, where $P_s, P_r, P_i \subseteq P$. The trust placed by the relying party in the IdP asserting the sub ject's identity is

$$t_{r,i} = k(p_r, p_i, c_{id}), k : P_r \times P_i \times \{c_{id}\} \rightarrow T \quad (2)$$

where $p_r \in P_r$, $p_i \in P_i$, $c_{id} \in C$ representing the trust context of "p_r providing identity". We can express the same result in terms of identities.

Let Δp be set of all identity contexts, δp, in which an entity p can be identified. δp can be seen as the context in which a given identity card is used, while Δp can be seen as the set of all contexts for which the entity has an identity card. Let I_p be the set of all identities used by p. Let each p be identified by an identity i in some instance, such that $i = \lambda(p, \delta p)$, where λ is a bijective function {p} × Δp ↔ I_p, and δp is the context in which p is operating. Let I be the union of all I_p for each p ∈ P in the given instance. In other words, in any given instance, each p ∈ P is identified by an i ∈ I (which in turn is determined by the identity context that p is operating in.)

Let α be a bijective function P ↔ I, such that i = α(p).

We rewrite (2) as

$$t_{r,i} = k(\alpha^{-1}(i_r), \alpha^{-1}(i_i), c_{id}), k : I_r \times I_i \times \{c_{id}\} \rightarrow T \quad (3)$$

where $I_r, I_i \subseteq I$ and $i_r \in I_r, i_i \in I_i$.

Let th be some trust evaluation threshold that is in some sense comparable to $t \in T$. We say that if $t_{r,i}$ exceeds or equals th, p_r has sufficient trust in p_i to assert the identity of p_s as is, but otherwise the level of trust is not deemed sufficient. We say that "asserting identity" is a task consistent with the context c_{id}, for which p_i is trusted. Let $i_{sattempt}$ be the result of an attempt to obtain is where

$$i_{sattempt} = \begin{cases} i_s, \text{if } t_{r,i} \text{ exceeds or equals } th, \text{ or} \\ i_{null}, \text{otherwise} \end{cases} \quad (4)$$

When trust thresholds are used in referrak and experience based frameworks for the purposes of comparison, it is typical to use real number values in the range (0,1). By letting $T = \{x : x \in R, 0 \leq x \leq 1\}$ and $th \in T$, we could simply replace "*exceeds or equals*" with $\geq$.

Figure 2.3 illustrates the thematic link between trust and identity.

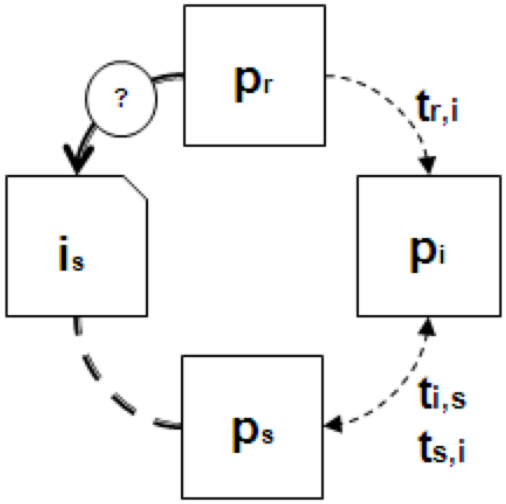


Figure 2.3: The link between trust and identity

In order for p_r to accept that i_s is the identity used by p_s , $t_{r,i}$ must be sufficient, so that p_r has trust in assertions made by p_i . Also, $t_{s,i}$ must be sufficient, since p_s must trust p_i to maintain personal information and/or credentials on its behalf. Similarly, $t_{i,s}$ must be sufficient, so that p_i can be certain that p_s is the entity that has a valid claim on i_s , which is why p_s is required to authenticate with p_i . These are the particular sufficiency requirements for the basic sign-on operation, other operations define their own trust sufficiency requirements as part of the operation design.

2.2.3 Conclusions on the Relationship between Trust and Identity

From equations (3) and (4), in the previous section, we see that trust and identity are mutually generative. In order to test one trust relationship (in one direction) we assume that other trust relationships are sufficient.

In the case of user-centric identity, where the user chooses an IdP to use, and is free to use any IdP,

$t_{r,i} \simeq t_{r,s}$, since the subject is essentially asserting his or her own identity. i_s can then be expressed in terms of itself, since in order for p_r to trust the identity assertion, there must already be trust between p_r and $p_{i/s}$. Only if the IdP chosen by p_s also has a trust relationship with p_r can the assertion be trusted with anything more than blind faith.

Interestingly, for services (the relying party), the question "Is this person X?" is not as important as the question "Is this person the same person that claimed identity X the last time?" It is generally not in the user's best interest to share his/her identity with others, so the second question can generally be verified by the service provider even if the first one cannot. An example of this is that a service provider is often more concerned with verifying that the person they are dealing with is the person with whom they have entered into some agreement with in the past, rather than necessarily verifying the person's name. Train and bus tickets, that are not transferrable, verify that the holder has entered into a service agreement with the transport provider, but makes no statement about the holder's identity, except that it is assumed that the holder has the same identity as the ticket buyer.

Some trust evaluation mechanism must be used to moderate trust ratings. An example of such a system on a decentralised network is a trust overlay [1]. It is the responsibility of the overlay to make trust ratings resistant to interference. The overlay must be provided with a set of initial trust relationship ratings for bootstrapping so that the it can begin in a viable state. These relationships can be drawn from real world trust relationships. Our trust model, outlined in D4.3 and expanded in section 3.2, provides a trust evaluation mechanism and overlay that can satisfy this requirement.

Trust ratings can be modified based on experience using schemes such as [4], where $t' = g(t, e)$, where g is a function that modifies $t \in T$ according to some measure of experience, e, drawn from interactions between entities in P.

2.3 Identity Operations

Identity operations were introduced in D4.1. We gave an overview there of how operations are constructed and processed into actor state machines, and how an operation specification can lead to operation execution (between the actors involved). The elements of an operation are re-iterated, in section 2.1.2, in the refined operation model. It was also necessary to define other elements of the operation model for the implementation. The full detail of the identity and operation model implementation will be described in D3.11. The entities that participate in operations form a subset of those represented in an identity context. The identity context could be seen as set of entities that

could potentially participate in a given activity, including operations and transactions.

Operations use trust is verify identity claims, in the form of assertions, make by one party (IdP) for the consumption of another (RP). An assertion can be any statement of the form, “Party A asserts fact X about party B to party C.” Thus, operations can be used to verify any attributes of any party that pertain to identity, such as names, memberships, ownerships, relationships, and other qualities.

Identity operations are designed to yield arbitrary protocol flows that verify identity claims in heterogeneous environments. This flexible and extensible design approach was chosen in order to support a range of technical infrastructures and to accommodate the evolving requirements of DE research.

2.3.1 The Sign-On Operation

The Sign-On Operation signs a user into an identity context, and is the most important operation in a DE since it must be performed before entities can interact (in a non-anonymous identity context). The identity claim that is verified by this operation is given by the assertion, “the entity that is 'speaking' holds identity X.”

Single Sign-On (SSO) is the equivalent in federation parlance, where two domains become federated and an identity in one domain applies in the other, and where when the entity authenticates in its home domain, there is no need to authenticate in the other. (Assertions are made from the identity provider of the home domain to that of the other domain instead.) Since most sign-on operations in a DE using our identity model require one or more federation type arrangements, which are usually temporary or contingent, and since a new sign-on operation is required for each identity context, SSO is less accurate as a title for this operation.

The implementation of the sign-on operation used depends on the construction of the identity context scope, which in turn depends on the infrastructure used in the particular portions of the DE in which the participating entities reside, and how this infrastructure is used. In OPAALS, a JXTA P2P infrastructure has been selected as a prototype DE platform. We describe below a sign-on operation that can be built upon this infrastructure.

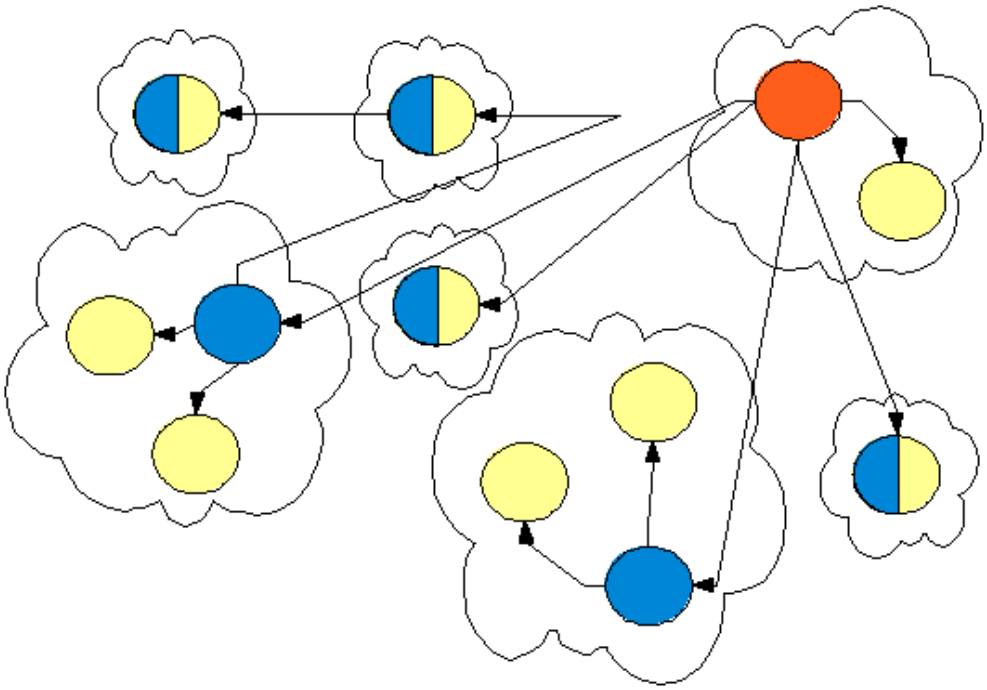


Figure 2.4: Scope of Identity Context in a JXTA based portion of a DE (such as that in a Sign-On Operation)

One feasible definition of an identity context scope, taking into account the OPAALS transaction model and a JXTA infrastructure, is defined by the statement, “those JXTA peers whose identity can be determined to the satisfaction of the initiator of a transaction.” Since identity is based on trust, the initiator must trust the asserted identities of these peers, or trust peers sufficiently to refer other peers' identities to it that it can trust. The chain of referral can reach as deep as trust transitivity allows (determined by trust policies on each node in the chain). In JXTA, each peer in a given PeerGroup trusts the identity of other peers implicitly, because the identity of each peer has been provided by the same MembershipService (IdP) (which may be an interface for many authentication backends). PeerGroups can be used to mimic small real-world groups of individuals who know each other well. Members of one PeerGroup can vouch for the identity of members of another PeerGroup if members are part of both PeerGroups (bearing in mind the requirement for locality), or if a member has a trust relationship (in the context of providing/referring identity[3]) with a member of the other PeerGroup.

Given in Figure 2.4 are a set of PeerGroups, illustrated by clouds, which are sets of peers that have

3 For simplicity, we do not distinguish between the trust context of providing identity and the trust context of referring identity claims provided by others. These are similar but potentially separate contexts.

an implicit trust relationship between them. The orange node (dark node in the top right corner) represents the initiator of a transaction, which must be able to verify the identity of all nodes participating in a transaction. The yellow nodes (light nodes) are participants, the blue nodes (other dark nodes) assert identity (are IdPs) on behalf of participants, and the mixed blue/yellow nodes perform both roles. The arrows represent trust relationships, in the context of providing/referring identity. We see that the overall identity context of the transaction (which may hold over many transactions) is the super set of small-world, implicit trust relationships, determined by PeerGroup membership.

We have spoken of the trust relationships above as though they were absolute. However, as we know from section 2.2.1, these trust relationships are graded. In the real world, chains of referrals are quite short, because although we may trust an individual to act as a referee for somebody else, we would put considerably less trust in an individual who has been referred by someone who in turn is being referred to us. This is the case to a greater or lesser extent in our identity model, which uses the trust model introduced in D4.3 and expanded on in section 3.2, depending on certain trust transitive policies. For example, if trust ratings are real numbers between 0 and 1, a referred trust rating might be attenuated by a value in the same range (e.g. 0.5) by the trustor. Thus, an entity whose referee trusts 0.8 might be attenuated by 0.5 for each referee in the chain. The identity, trust 0.8, and referred twice, would therefore be trusted 0.8 x 0.5 x 0.5 = 0.2 by the trustor, which in our model is the transaction initiator. If the referred trust rating is sufficiently low (below a certain threshold, say, 0.6), the initiator will not trust the identity of the entity sufficiently to allow the entity to participate in the transaction. It is clear that this attenuation of trust ratings through levels of referral serves to prune the graph of (sufficient) trust relationships, thus re-enforcing a localised identity context, inimical of the real world.

3 Distributed Trust Model

This section provides a revised description of the Distributed Trust Model for digital ecosystems. A detailed discussion of Rating Agencies is followed by a description of some trust algorithm types and one such algorithm will be examined to show how it performs under certain conditions for both trust based on direct experience trust trust based on referrals. Integration with the Accountability Model and also with the Transaction Model is discussed. An explanation of how this model will be deployed in the digital ecosystem is also provided. Finally two scenarios, one based on Transaction Model integration and one on Rating Agency trust evolution are supplied.

3.1 Rating Agencies and Institutional Trust

Trust building mechanisms are essential for the sustainability of the DE because trust is a prerequisite of every successful interaction between entities. The most widely used trust assessment mechanisms on the Internet today are based on recommendations of peers. For example, Amazon allows customers to rate products and eBay allows users to rate other users. In D4.3 we provided an extensive overview of peer-to-peer recommendation systems. Such recommendations can be global (i.e., a sum of all recommendations expressed in the system) or local (i.e., specific to the peer that makes the enquiry). Also in D4.3 we showed that local trust is suitable for DEs in which entities form coalitions and collaborate with their trusted peers in order to create new structures. In the model we have proposed, entities exchange recommendations with their “neighbours” in order to discover the trustworthiness of a particular peer.

However, peer recommendations (a.k.a. social recommendations) are subjective and entities do not have a good understanding of the way in which the recommendation value was computed (e.g., used criteria). An alternative to this are institutional recommendations, issued by authorized institutions. They make use of “hard” data in computing a recommendation value. We define as “hard” data any information about the entity that can be verified by a third party. We will call such kind of recommendations “institutional ratings” and the institutions issuing them “rating agencies”. As opposed to social recommendations, institutional ratings are repeatable and objective. Our goal in this chapter is to propose a model that allows DEs to benefit from institutional ratings. We will first provide an overview of institutional recommendations in the literature and of the way they are conveyed in online environments. Afterwards we will propose a framework, identify the main

actors and the information (or documents) that need to be exchanged between them. We will further derive requirements for the implementation of these documents.

3.1.1 Institutional Recommendations

Many studies have been dedicated to the importance of *trusted third parties* in the form of institutions in building trust between entities in online environments. Palmer et al. [6] and Sharkar and al. [7] call them *intermediaries* and show their importance in building trust on the WWW and online transactions. Other studies refer to this kind of recommendations as *institutional trust*. In 1986, Zucker[8] identified institutional trust as the most important way by which trust is created in impersonal economic environments where there is no sense of community. She gives as an example, farmers who previously dealt with local markets are no involved in more international markets with little local community interactions. She identified two dimensions of trust: (I) third-party certifications and (ii) escrows that guarantee the outcome of a transaction. McKnight and Chervany [9] also identified institutional trust as a critical part of Internet transactions. In their paper, they propose an interdisciplinary trust typology that relates trust constructs to e-commerce consumer actions. For institutional trust, they identified two sub-constructs: (I) structural assurances or protective structures (e.g., guarantees, contracts, regulations, legal recourse, processes or procedures) that are in place and conduce to situational success, and (ii) situational normality, meaning that the Web consumers believe the Internet situation is normal and, hence, success in the transaction can be achieved. Pavlou et al. [6] identified two dimensions of institutional trust and described their role in building online inter-organizational trust: (I) third-party institution-based trust established through intermediaries such as B2B marketplaces, and (ii) bilateral institutionalized trust which defines processes, standards and norms used by two institutions to manage transactions.

Institutional recommendations or ratings issued by some trusted third party assure consumers that a firm confirms to a particular policy or that a product satisfies certain requirements. This information is conveyed to the user by displaying the logo of the institution on the Web site of the firm or product. For example, TRUSTe and BBB Online are the most well-known companies that guarantee Web sites respect consumer's privacy. Other companies such as Tucows or SnapFiles guarantee that software applications satisfy certain requirements (e.g., usability, documentation and support, cost). SiteTrust Network guarantees that approved online merchants satisfy certain Efficiency, Security, and Legality standards. The ratings or recommendations issued by these companies enable trust

relations between consumers and e-vendors.

Another example of institution recommendations are diplomas and certificates in the real world. Other examples are institutions that certify certain products comply with norms and standards established by an authority. For example, cars, bio-products, sunscreens can only be commercialized if they comply to the norms established in a particular country by a public body. Hotel or restaurant ratings in the form of "stars" are another form of institutional ratings.

Inspired by institutional trust research and institutional ratings on the Internet and in the real world, we will propose in the next section a framework for rating agencies.

3.1.2 Rating Agencies Framework

In the following, we first provide an overview of institutional ratings and derive requirements that make them usable and trusted. Second, we identify the actors involved in IR and their roles. Then we identify and describe the documents that are exchanged between the actors. Lastly, we identify problems of current representation and exchange of such documents and propose a solution that addresses the problems.

Ratings and their evaluation

A Rating Agency (RA) is a trusted institution that uses well-defined objective criteria to evaluate some entities (e.g., products, services, users, companies, infrastructure etc.). Ratings issued by RAs need to be:

1. consistent: define some total partial order among entities
2. repeatable: any individual would reach a similar rating value for a given entity if the rating is based on the same criteria and evaluation method.

These properties distinguish institutional rating from social rating. In its study about institutional ratings, Sint [5] also argues that for an RA to be accepted by the concerned public, it needs to be able to provide "consistently useful ratings".

The rating criteria used for evaluating the entities depends on the rating agency selecting the criteria and on the type of entity rated. Sint [1] identified the main methods used to rate entities. In the following we list some of them:

- Measurement and experiment: testing products can give information about their composition and properties (e.g., durability, security, conformance to standards or norms)
- Experts: most formal ratings are done by experts or specialists who define a formal procedure to reach an objective result. For example, the International Organisation of Consumer Unions (IOCU) published guidelines for testing and the European Testing Group (ETG) organises co-operative testing of products by different national testing organisations and additionally involves experts for special domains under consideration.
- Jury: a jury of experts can work informal or establish some formal procedures.
- Peer groups: assessment of scientific work is based on peer review.
- Automatic procedures: for example the number of visitors of a Web page can be automatically counted.

Whatever criteria are used, it is important for RAs to precisely define them and describe how the criteria are assessed. This is an essential requirement to ensure consistency and repeatability of ratings. Moreover, for the rating to be useful to users, they need to have access to the description of the used criteria.

Because the used criteria depends on the type of entity evaluated (e.g., user, service, data) we will not define in this deliverable the criteria and how it should be measured. We only require that the criteria (i) yield to objective and repeatable ratings of the entities, and (ii) are available to users. We can say that the criteria represent the semantics of the rating.

Because many criteria could be used to rate an entity, usually RAs aggregate the scores into an overall rating (e.g., a number of stars in hotel rating) which is meaningful to users. The aggregation could be done in several ways and give more weight to different criteria. The choice of the aggregation function usually depends on the targeted users (e.g., average users, expert users, users interested in one particular aspect more in the others).

We call all the information needed for a user to understand a rating such as description of criteria, evaluation method, aggregation methods, *rating metadata*, and we require that is made available to users.

Actors and their roles

We distinguish three main roles involved in the process of institutional rating:

1. **Rating Agency**: the institution that rates entities (e.g., users, services, data, infrastructure) based on well-defined criteria.
2. **Rated Entity**: as explained in D4.3, our goal is to provide trust at all levels of the DE, so in our model, the rated entity can be any identifiable entity.
3. **User**: makes use of ratings issued by one or several RAs to assess the quality or trustworthiness of entities.

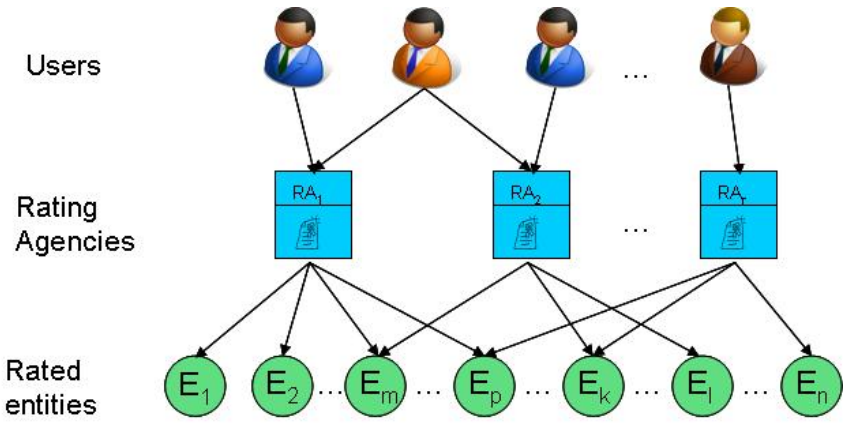


Figure 3.1: Roles and interactions

Depending on the context, an entity could have any, two or all of the above roles. For example, a company that is rated by an RA can consume ratings issued by other RAs to assess potential business partners or products that may buy. Universities are a real-world example of entities that assume all three roles. They are accredited by authorized institutions in their country and ranked by many specialized organizations (e.g., a ranking of European business schools). In the same time, they issue certificates or diplomas to their students. For admission or hiring process, universities consume diplomas that potential students or professors obtained in other universities.

Figure 3.1 shows a three party graph in which the identified roles interact. RAs rate entities based on some criteria and users consume ratings issued by RAs. An important fact to note is that an RA

rates only a limited number of entities. The sets of rated entities of two RAs could be disjoint or overlap in any degree. Because of this, users frequently need to rely on ratings from several agencies.

Documents exchanged in Institutional Ratings

In this section we derive which information or documents need to be exchanged between the entities in Figure 3.1 in a way that allows RAs to issue ratings and users to consume them in a secure and trusted way.

In order to trust a rating, a user needs to:

1. trust the agency that issued it,
2. verify the identity of the agency (i.e., that the rating was indeed issued by the trusted agency),
3. (possibly) verify that the agency is authorised to perform the rating,
4. understand the meaning of the rating,
5. verify that the rating is genuine and has not been modified by a third party.

Trust in an agency can be built in different "soft" ways such as peer recommendation, reputation of the agency, bilateral relations with the agency, disposition to trust of the user (e.g., Web browsers warn users about unknown certificates and most often users choose to trust them). This kind of trust building mechanisms was the focus of deliverable D4.3 and would not be addressed here. In this work, our goal is to define the "hard" data that allows users to securely identify the agency and discover if the RA is authorized to perform the rating or not. To achieve this, the following documents are needed:

1. ***identity certificate*** of the agency which allows users to authenticate an agency. This also allows authenticating any documents issued and signed by the agency. We will refer to any document that carries the signature of the RA, as *certificate,* to underline this aspect.
2. ***attribute certificate*** of the agency, which states if the agency is qualified of authorized to perform the rating. For example, a public body authorizes certain institutions to evaluate "bio" products.

3. ***rating metadata certificate.*** This document needs to be signed because it is public and users need a way to check its validity. A rating should have a link to this certificate.

4. ***rating certificate.*** Ratings need to be signed to make sure they are indeed genuine and issued by the claimed RA. The identity certificate of the RA is needed to verify the signature on this document.

Hence, in order to verify the authenticity of a rating, the rating certificate needs to contain the following information:

- the identity of the RA that issued the rating
- identity information or a reference to the entity being rating
- a reference to the rating metadata which allows users to understand the rating
- the overall rating and the rating for each criteria
- additional information such as rating validity

Before proposing a solution to the representation of these certificates, we will examine current Web-based representation of ratings and identify their main shortcomings.

Shortcomings of current Web-based representation of ratings

Currently, ratings are represented to the user as logos linked to the RA that issued them. These logos can show a number of stars or can simply say that the Web site is certified by an authority. To check they are valid, users need to click on the logo and then are redirected to the Web page of the institution that provided it. Sometimes users are redirected to some report issued by a specialist that evaluated the entity. We can identify several shortcomings of this approach:

(a) The ratings represented like this can be easily faked and so can be the Web page of the agency (e.g., phishing attacks).

(b) Information about how the rating was computed is missing many times so the users have a hard time understanding the rating.

(c) Ratings cannot be exchanged using some widely used protocol between different entities.

(d) Ratings cannot be automatically processed by applications. It could be desirable to have an application that is able to list entities based on the ratings they obtained, for example, starting with the highest score.

The above problems can be solved if:

1. ratings, and also the other documents exchanged between the different roles, are expressed in a *standard* way.

2. security mechanisms are in place to correctly authenticate the documents and enties (i.e., RAs, rated entities). This includes authenticating the agency that issued the rating and checking its legitimacy and competence to issue the rating, verifying the authenticity and validity of the rating, correctly identifying the entity to which the rating refers.

3.1.3 Rating Agencies in DEs – the interoperability problem

The framework we propose raises many interoperability issues in distributed and heterogeneous environments such as Digital Ecosystems. In this section we will identify the main interoperability issues and propose solutions for addressing them. This is important because it ensures the model scales to the distributed nature of the DE and ratings can be used even outside specific domains or circles or trust.

To introduce the interoperability problem, we take a real world scenario and then show how it translates to DEs and our framework. Let's consider the case of foreign diplomas recognition and equivalence. Each country has established its own educational system. There are a number of defined diplomas or certificates that a school can issue (e.g. high school certificate, degree, engineer, Master). For obtaining each qualification, specific requirements that need to be met have been defined (e.g., number of study years, minimum number of credits, minimum grade to be obtained). Moreover, schools need to be accredited by the state to issue such diplomas.

By relating this particular example to our model, schools become RAs, students are rated entities, and diplomas are rating certificates.

In the same country, a diploma is understood and accepted because people:

1. are familiar with the system (i.e., understand the rating metadata),
2. know and trust the schools, and
3. trust the diplomas are genuine because they are hard to forge or can be easily verified.

However, when going abroad, diploma recognition becomes a problem and special authorities are in place to perform the recognition or equivalence with local diplomas. Usually, agreements between countries and predefined rules clearly indicate the recognition process and the institutions

authorized to perform it. For example, in Switzerland, the Federal Office for Professional Education and Technology (OPET) is responsible for establishing equivalence between foreign and Swiss diplomas (i.e. "Berufsattest", "Fähigkeitszeugnis", "Berufsmaturitätszeugnis", "Fachausweis der Berufsprüfung", "Diplom der Höheren Fachprüfung", "Diplom einer Höheren Fachschule" and “Diplom einer Fachhochschule”).

If someone obtains a diploma abroad, in China for example, and wants to apply for a job in Switzerland, they would need to go to OPET and ask for diploma recognition. Usually, OPET would need to do the following: (i) verify that the diploma was issued by an accredited school in that country, China in our example, and (ii) give an equivalence with one of the Swiss degrees. OPET would issue a new document stating the recognition. Companies in Switzerland would accept the document because of the trust they have in OPET.

At the moment, this process is being done manually. However, by transposing the scenario to DEs and the proposed RA framework, the following steps and documents would be needed:

1. The university in China obtains an identity certificate (e.g. X.509 certificate from VeriSign).
2. The university gets accredited by the state and is issued an attribute certificate. This certificate could state which degrees is authorized to issue.
3. The university issues a rating certificate to a graduated student. The certificate states the degree that was obtained, and depending on the case it might include the overall grade and/or each course attended with the grade obtained etc.
4. The former student then requests the recognition and uploads the rating certificate and the attribute certificate of the institution to an application running at OPET.
5. The application at OPET, which trusts the institution accrediting schools in China, and hence, has its identity certificate stored locally, can check that the school was accredited and then check the validity of the rating certificate. If this succeeds, OPET automatically issues a new certificate stating the recognition and equivalence.

The rating criteria of two agencies could be different. Users trust and understand only some agencies and their criteria. To allow them to make use of ratings issued by other RAs, we need to provide a mapping.

3.2 Trust Model

The OPAALS Trust Model for digital ecosystems is described in [13] and is shown below in Figure 3.2 below. The approach is to use a trust overlay network for providing a community based approach to trustworthiness based on the reputation of entities. The Entity can represent a node, service, resource, a service provider or a service consumer. Each entity has a Trust Manager associated with it. The entities gain experience from interacting with other entities and publish reports of these experiences to the Trust Manager. Using a pre-defined context dependent algorithm the Trust Manager updates the entities' local trust and based on a policy of sharing trust information provides trust updates to other Trust Managers in the overlay network.

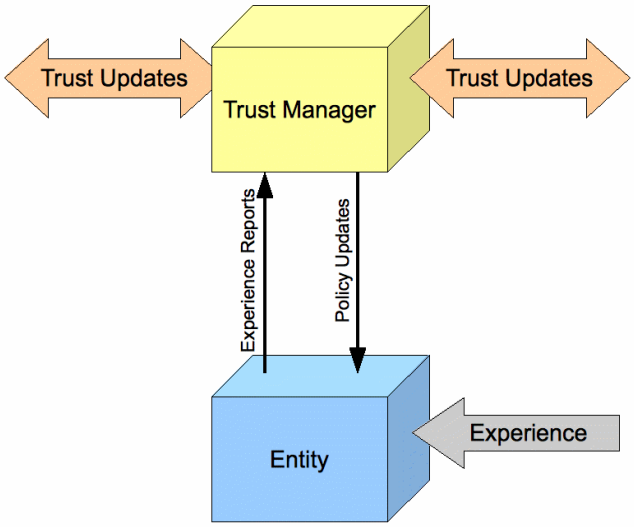


Figure 3.2: OPAALS Trust Manager

A UML Class Diagram of this model is provided in Figure 3.3. Entities are responsible for creating trust algorithms. Each Entity has an associated TrustManager. A TrustAlgorithm is associated on a per entity basis with an trustor and a context. Entities create these algorithms and publish them to the TrustManager. The TrustManager maintains a set of these algorithms and when it receives an ExperienceReport, it uses the appropriate TrustAlgorithm by performing a lookup on trustor-context tuple.

The TrustManager also maintains a set of TrustValue objects which it updates after performing the algorithm on the ExperienceReport. The Entity can request PolicyUpdates from the TrustManager. These policy updates are derived from current TrustValues and are used by the entity in making choices about future interactions with other entities. The TrustValue class includes a 'source' element indicating whether the value was derived from direct experience or based on referrals. The class also has a timestamp attribute, as recently gathered trust might be of more value than older values. Also there is a confidence attribute, used to denote the confidence in this Trust Value (e.g. this value can be increased each time the Trust Value is updated, i.e. 100 updates to the Trust Value implies more confidence in the Trust Value than 10) .

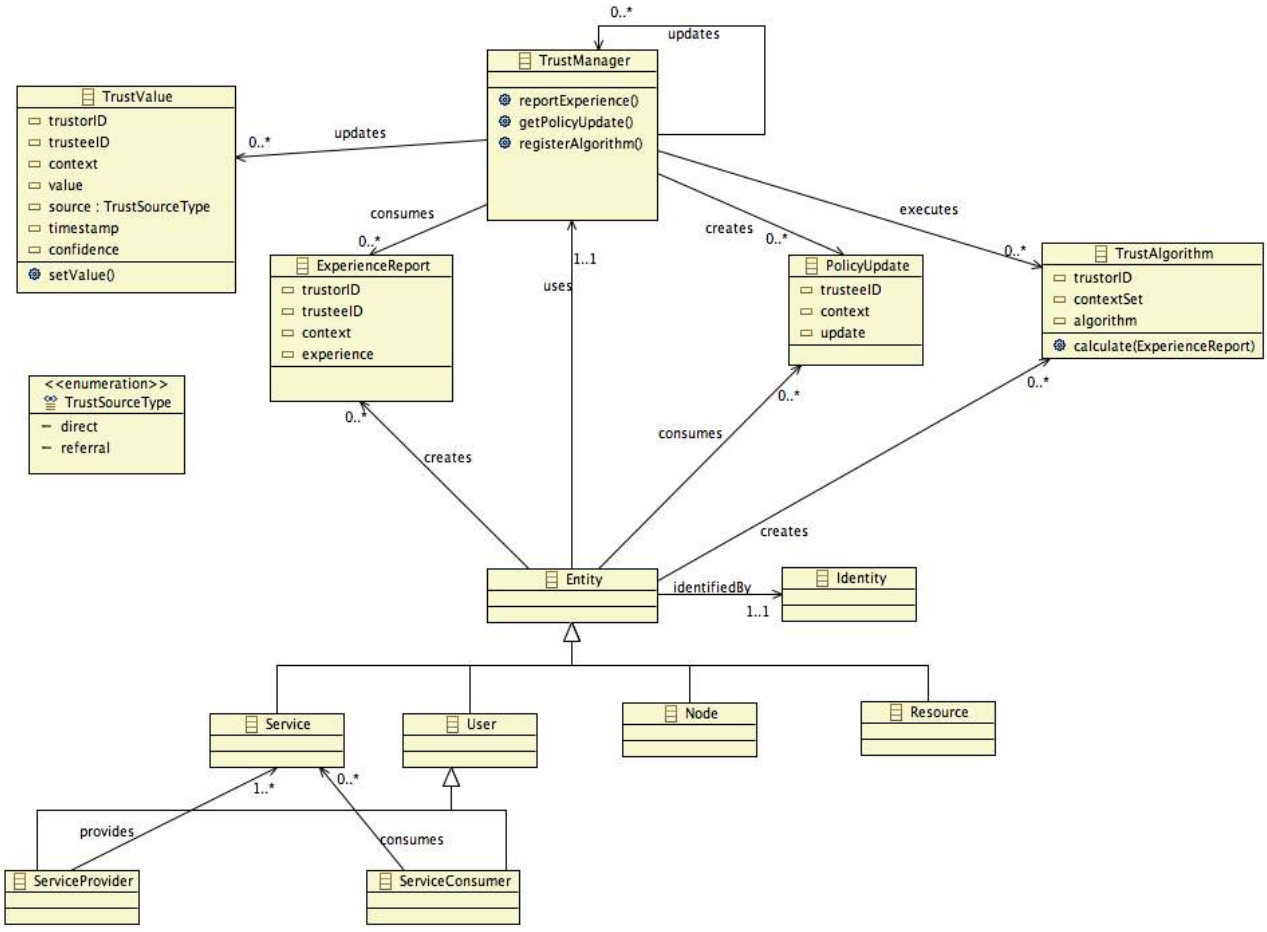


Figure 3.3: UML Class Diagram of TrustManager and associated classes

3.3 Algorithms

While the focus of the work performed within OPAALS is to develop a model and not necessarily to provide or recommend specific algorithms to provide evolutionary trust updates, it is nonetheless useful to examine trust algorithms and how they might be deployed in the OPAALS digital ecosystem trust model. In fact, we take a plug-in approach to algorithm deployment, where the end users publish how trust can evolve in particular contexts. Here we will outline some algorithmic approaches to trust evolution.

There are two primary ways of determining trustworthiness. Firstly trustworthiness can be established through direct experience or first hand information (direct functional trust). Secondly, trustworthiness can also be established through referrals or second hand information (indirect functional trust). A more complete view of trustworthiness can be achieved through the availability of both direct experience data and referral data combined with suitable algorithms within a given context. Reputation can be derived from a set of referrals.

Here we briefly discuss some common approaches to developing algorithms for trustworthiness. Then there follows examples of how one of these algorithms (exponential average algorithm) perform under varying conditions. Many of these algorithms do not “remember” the data history. In the case where this is desirable it can addressed by the persistence of historical experience reports.

3.3.1 Moving average

Each new assessment of trust (experience or referral) is fed into a simple moving average, based on a sliding window. Direct experience can be given more weight than third party referrals in the averaging if desired. More advanced moving averages are also possible, where old data is “remembered” using data reduction and layered windowing techniques.

3.3.2 Exponential average

Each new assessment of trust is fed into a simple exponential average algorithm. 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. Direct experience can be given more weight than referrals by using a different (higher) parameter.

3.3.3 Exponential average, per service parameters

Parameters depend on service. Some services are more revealing than others of user's trustworthiness. Usage of a service that provides little opportunity for exploitation shouldn't have much impact on trust in the user.

3.3.4 No forgiveness on bad experience

A node behaving badly has its trust set to zero forever. This could be considered suitable for critical services.

3.3.5 Second chance (more generally, *n*th chance)

Intrusion detection and other security systems are prone to false alarms. Also, good nodes can be temporarily hijacked and should be given the opportunity to recover. Thus a suitable variation on the "no forgiveness" algorithms is to keep a count, perhaps over a sliding time window, of misdemeanours. Trust is set to zero (possibly forever) on n misdemeanours.

3.3.6 Hard to gain trust; easy to lose it

To discourage collusion between bad nodes, there is a case for making it hard to gain trust and easy to lose it. Thus attackers will require lots of effort to artificially increase their trust scores, either by repeated benign use of low-threshold services or by issuing repeated positive recommendations about one another. Even a little malicious activity will cause trust to fall significantly.

3.3.7 Use of corroboration

To prevent an attack by up to k colluding bad nodes, we could require positive recommendations from at least $k+1$ different nodes.

Next, we examine a simple exponential average algorithm in the context of both direct experience and referral.

3.3.8 Exponential Average Direct Experience Trust Algorithm

In the case of direct experience the exponential average algorithm, published by us in [15], will be written as:

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

Where:

$T_{i,j(n)}$	is the nth value of trust placed by i in j
E	is the latest direct experience report $0 \leq E \leq 1$
α	is the rate of adoption of trust, where $0 \leq \alpha \leq 1$

The value α determines the rate of adoption. $\alpha = 0$ means that the trust value is not affected by experience. If $\alpha = 1$ the trust value is always defined as the latest experience and no memory is maintained.

The algorithm behaviour is illustrated in Figures 3.4 through 3.6. Each of these illustrations considers the case where experience is a measure of the dependability of a service and that dependability is measured is measured as a binary value (1 or 0).

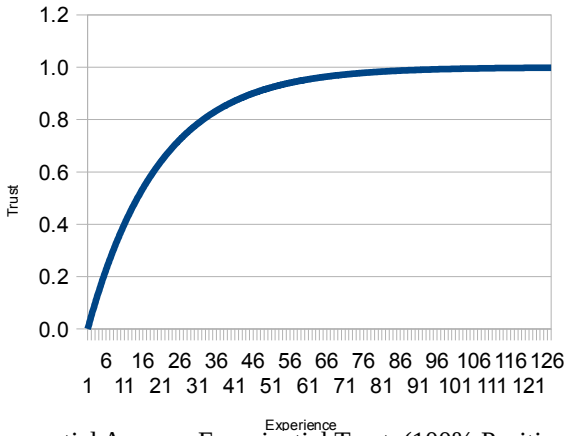


Figure 3.4: Exponential Average Experiential Trust (100% Positive Experience)

When the experience E is always 1 (i.e. the service is always dependable) this trust evolves as in Figure 3.4. Figure 3.5 shows how this trust evolves where every 20th experience is reported as a

failure. There remains a trend of exponential recovery but the failures impact on the trust value while allowing for recovery in subsequent success reports. Similarly, Figure 3.6 demonstrates how the algorithm performs when clusters of failures occur.

The rate of adoption and reduction in trust in this algorithm is the α value. In some cases it might be desirable that this value varies depending on the latest experience. For example in some cases it can be argued that while it is difficult to gain trust (slow adoption of trust) it is easy to lose it (fast reduction) based on bad experience. To model this we can vary the α value as follows:

if $T_{i,j(n-1)} < E$
then
$\alpha = 0.1$
else
$\alpha = ٠.٤$

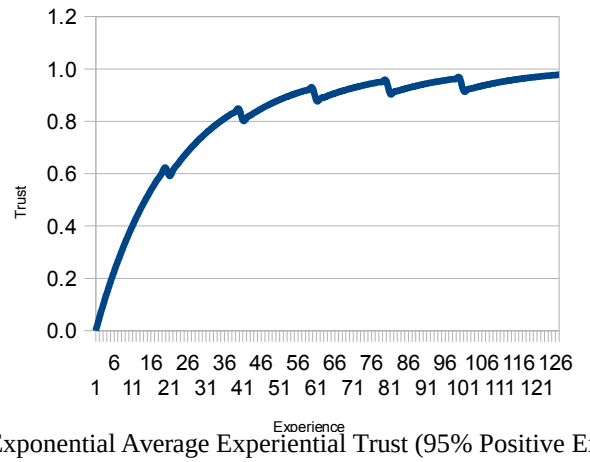


Figure 3.5: Exponential Average Experiential Trust (95% Positive Experience)

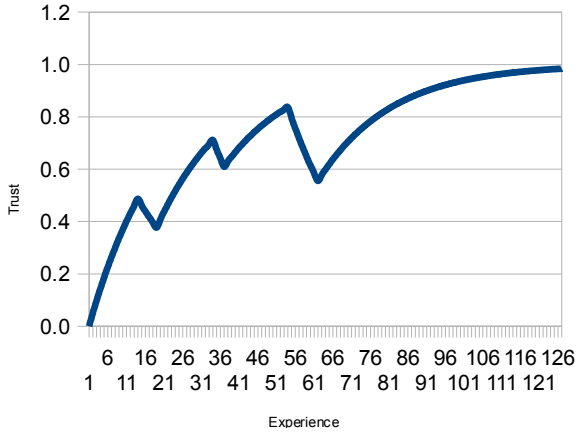


Figure 3.6: Exponential Average Experiential Trust (with failure clusters)

3.3.9 Exponential Average Referral Based Trust Algorithm

In the case of referral based trust the exponential average algorithm, published by us in [15], can be written as:

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

Where:

$T_{i,j(n)}$	is the *n*th value of trust placed by *i* in *j*
$T_{i,k}$	is the trust *i* has in *k*'s referral
$T_{k,j}$	is the trust *k* has in *j*
β	is the influence that referrals have on local trust ($0 \leq \beta \leq 1$)

The value of $T_{i,j}$ represents an indirect functional trust that i has in j. In this case i has received a recommendation of direct functional trust, $T_{k,j}$. $T_{i,k}$ is the trust that i has in k's referral. The larger the value of $T_{i,k}$, (i.e. the more i trusts k's referral), the greater the influence of k's trust in j on the newly updated value of $T_{i,j}$. Note that, if $T_{i,k} = 0$, this causes $T_{i,j}$ to be unchanged.

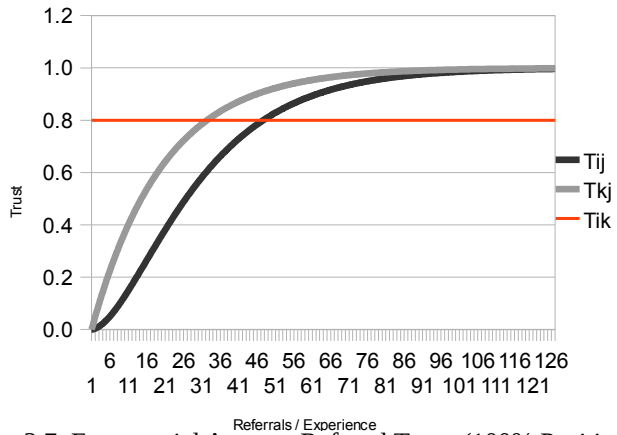


Figure 3.7: Exponential Average Referral Trust (100% Positive Experience, β = 0.1)

The algorithm behaviour is illustrated in Figures 3.7 through 3.12. Each of these illustrations considers the case where experience is a measure of the dependability of a service and that dependability is measured is measured as a binary value (1 or 0).

Figure 3.7 above shows how this algorithm behaves for a fixed value for $T_{i,k}$ and $\beta = 0.1$. $T_{k,j}$ is a direct functional trust derived as described above and shown in Figure 3.4. $T_{i,j}$ indicates how the indirect functional trust evolves. Figure 3.8 below shows how this is affected when the direct functional trust experiences one failure in every 20 experience reports. Figure 3.9 shows what happens when the direct functional experience reports yield one failure in 20 experiences followed by a general failure after which only negative reports are received. Each of these examples consider the referral trust to be constant. The following three figures (Figures Figure 3.10, 3.11 and 3.12) show how the indirect functional trust evolves when the referral trust varies.

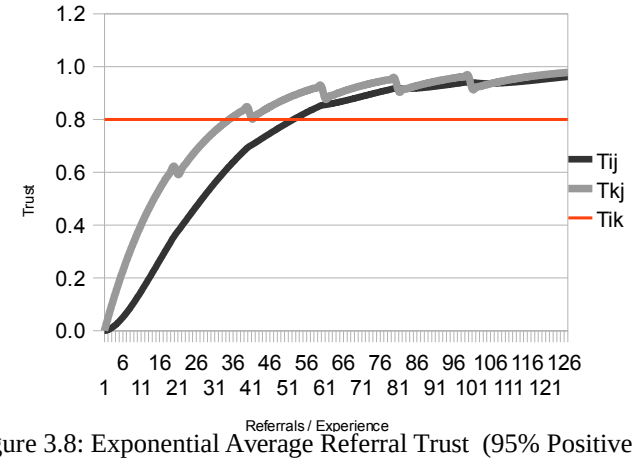


Figure 3.8: Exponential Average Referral Trust (95% Positive Experience, β = 0.1)

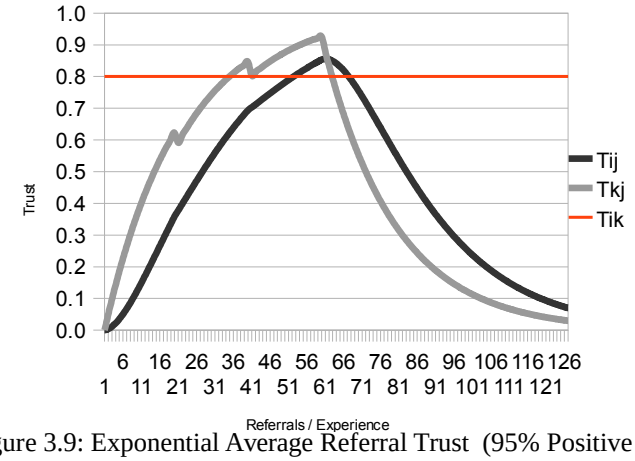


Figure 3.9: Exponential Average Referral Trust (95% Positive Experience, general failure after 60 experiences, β = 0.1)

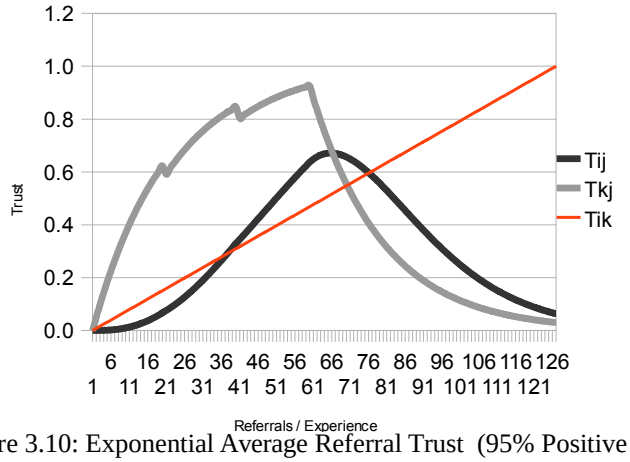


Figure 3.10: Exponential Average Referral Trust (95% Positive Experience, general failure after 60 experiences, β = 0.1, linear increase in referral trust)

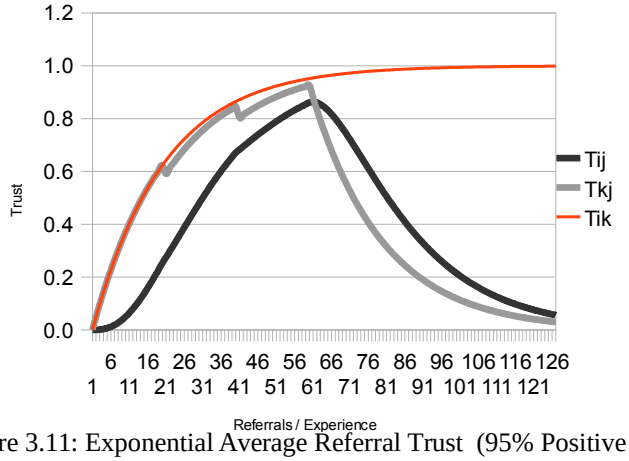


Figure 3.11: Exponential Average Referral Trust (95% Positive Experience, general failure after 60 experiences, β = 0.1, exponential recovery in referral trust)

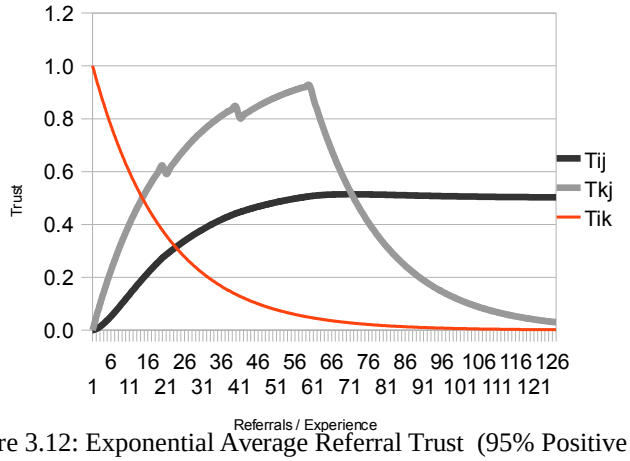


Figure 3.12: Exponential Average Referral Trust (95% Positive Experience, general failure after 60 experiences, β = 0.1, exponential decay in referral trust)

3.4 Integration with Identity

Identity operations, introduced in D4.1 and clarified further in section 2.3, specify the contigent protocol flow of messages between entities (which are represented by actors in an operation). Operations also specify the minimum trust sufficiency criteria for an operation to succeed. It is not only necessary that actors perform their role in the operation in terms of message processing and message passing, but these actions must be trusted by the other actors involved. Without trust, the validity of the messages passed between the actors cannot be given any credence.

In Figure 3.14 an operation gives rise to the protocol flow comprising connections 1 to 8. We isolate connection 6 in order to illustrate integration with the trust model. IdP1 receives a communication from IdP2 represented by connection 6. In order for IdP1 to trust this communication, which is composed of some identity assertion, IdP1 must trust IdP2 sufficiently. From section 2.2.2, we can say that connection 6 succeeds only because $t_{IdP1,IdP2} \geq th$ (th is some trust threshold).

The trust threshold for that connection is specified in the operation design, while the dynamic trust evaluation of IdP2 is returned to IdP1 from its trust manager as part of a policy update. The experience of IdP1, and the trust updates from other trust managers concerning IdP2, should indicate a history of IdP2 making sound assertions representing identity claims.

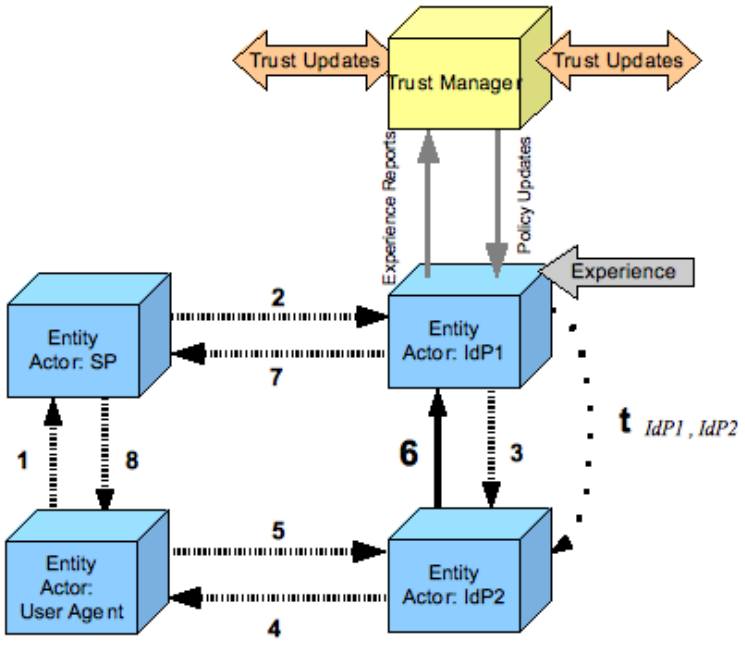


Figure 3.13: Identity and Trust

3.5 Integration with Accountability

Accountability data is a source of experience for Trust Manager overlay network. See Deliverable D3.8 for details of the Accountability Model. There are several ways in which the data can be used for gathering Experience Reports for trust evaluation.

- Accountability data can be used directly as Experience Reports. Algorithms can be be developed to operate directly on the accounted data and published to the Trust Manager.
- Retrieved data from an Account Holder can be analysed by the Entity and used as an input to pre-published trust algorithms.
- Access to accountable data can be inserted in algorithms and the Trust Manager can run the algorithm against the resultant query.
- Alarms raised by Mediators and Accounting Authorities can also be used for the generation of trust. These alarms are reported in the accounted data.

Other sources of experience reports for trust can also be checked against the accountability data. For example, if a peer's trust is degraded due to false reports, the accountability framework provides a means of the peer disputing the false report with accountable evidence. In this regard, private accountability presents a situation where false reports cannot be defended against without revealing private keys to view the evidence. See Deliverable D3.8 for more detail on these issues.

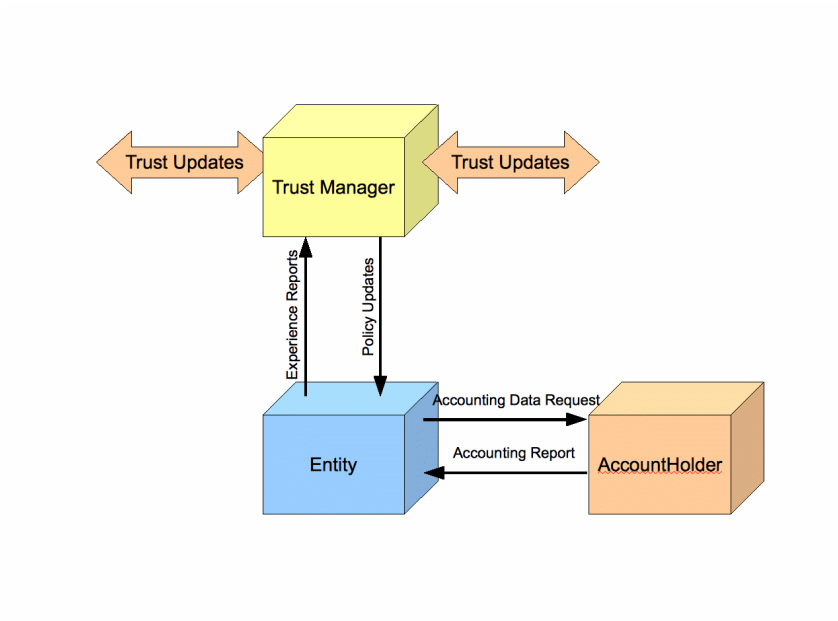


Figure 3.14: Accountability and Trust

3.6 Integration with Transaction Model

Transaction results are another experience report which can be used as an input for Trust Algorithms. The Transaction Model is document in [14]. A schema to represent the transaction context has been developed. Based on discussions between WIT and Surrey it was decided to insert an extra element in this schema which can represent the trustworthiness of a service in specific contexts. When a transaction execution finishes a transaction log is generated. This log is used as an input to a Trust Algorithm which can analyse the outcome of the transaction execution and generate updates to Trust Values of services in the contexts that the logs can capture. The policy update of this results in an updated Transaction Context document which can influence future executions of the transaction in the choice of services making up the composition. The specific targeted case where this is most useful is the rating of alternate service offerings in the case of the sequential alternate coordinator. Updates to these rating can result in changes in the order of which these alternates are selected.

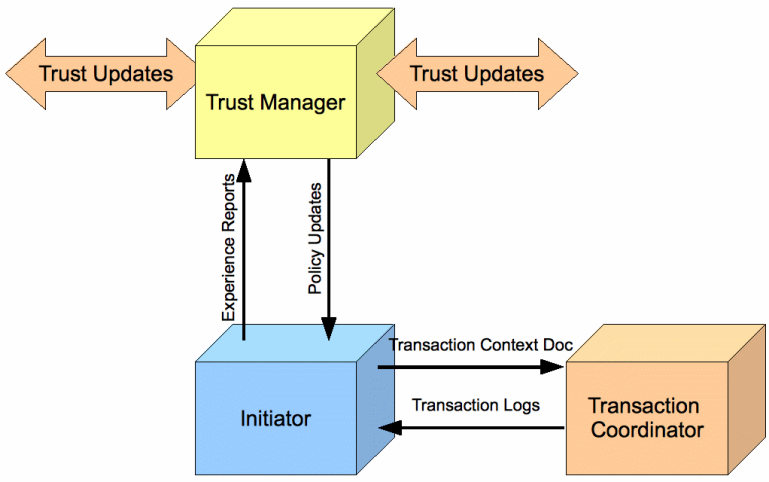


Figure 3.15: Transactions and Trust

3.7 Deployment of Model

For deployment of the model we make three design decisions as follows:

1. We take a plug-in approach for Algorithm management. Algorithms are created by the end user and published to the Trust Manager. Each Algorithm is relevant to a context or a set of contexts. For example an end user would define different different algorithms for service

availability and service accuracy.

2. Referrals are retrieved on a pull basis. Trust Managers request from other Trust Managers trust values for a user in the appropriate context. An alternative approach is to use a publish/ subscribe model where Trust Managers are alerted of trust updates in a particular context.

3. Policy Updates are retrieved on a pull basis. For example, before an end user chooses which service to consume, it sends a request for an update to the Trust Manager. This is different a publish/subscribe model where the end user subscribes to certain types of updates and receives updates published by Trust Managers.

In the remainder of this section we describe the three primary operations of the Trust Manager overlay model. Publishing algorithms describes how user's make algorithms available to the Trust Manager for future trust value calculation. Experience Reporting describe how direct experience reported to the Trust Manager is used in updating trust values. Updating Policy describes how the end user can retrieve current trust values based on past experience and trust referrals in the overlay network.

3.7.1 Algorithm Publishing

The end user creates an Algorithm through the use of a Algorithm Editor. Each algorithm is associated with a context (or a set of contexts). The algorithm can be used for the calculation of trust based on direct experience or on referrals. In the case of direct experience the algorithm is dependent on the format of the Experience Report which will be used as an input. In this case, a schema of the document will be loaded into the Algorithm Editor to aid the algorithm design. Once the Algorithm is created the user publishes this algorithm together with its context to its Trust Manager. The Trust Manager saves this algorithm in a local store. The process is shown below in Figure 3.16.

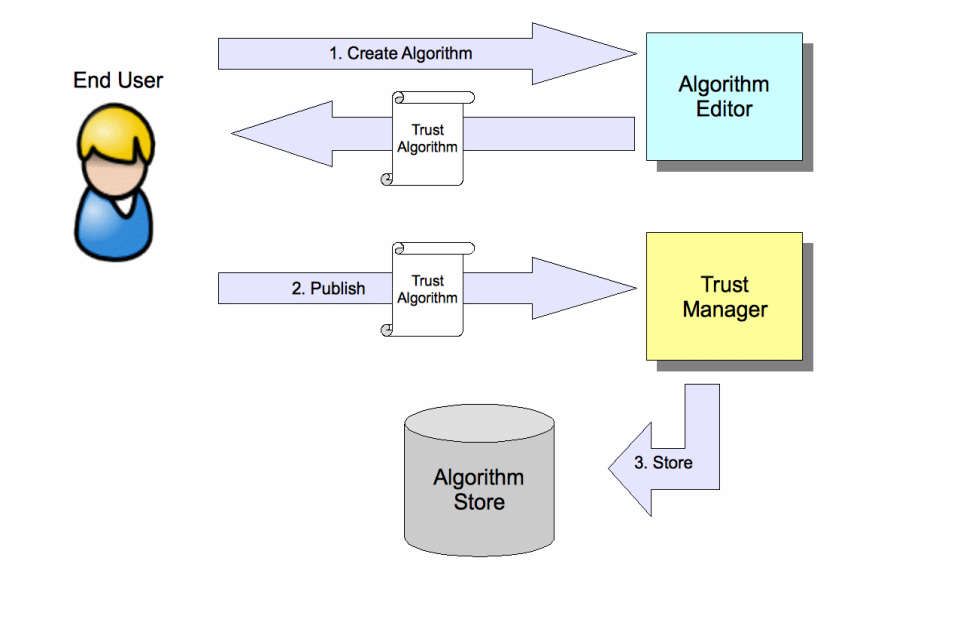


Figure 3.16: Publishing Trust Algorithms

3.7.2 Reporting Experience

The Entity sends an Experience Report to the Trust Manager. The Trust Manager examines the experience report, determines the context and retrieves the appropriate algorithm for the calculation of local trust. When the algorithm is run against the experience report the Trust Value in the local store is updated with the new value. The Trust Manager examines whether any referrals have been received for this trustee and context pair. If it has it retrieves an algorithm to update the Trust Value of the referrer in the referral of the context. The process is shown below in Figure 3.17.

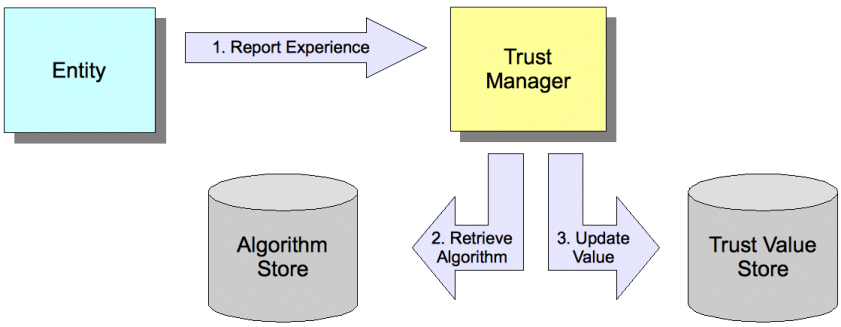


Figure 3.17: Experience Reporting in Updating Local Trust

3.7.3 Updating Policy

We show a pull approach to Policy Updates. The Entity requests an update for a given trustee and context from the Trust Manager. The Trust Manager retrieves the value from the Trust Value Store, if one exists and also performs a request for referrals. The Trust Manager queries the Algorithm Store for suitable algorithm to evaluate referrals in the context. If a Trust Value is present, the Trust Manager retrieves an algorithm for the combination of referrals and direct experience Trust Values and calculates the update to return to the Entity. The process is shown below in Figure 3.18.

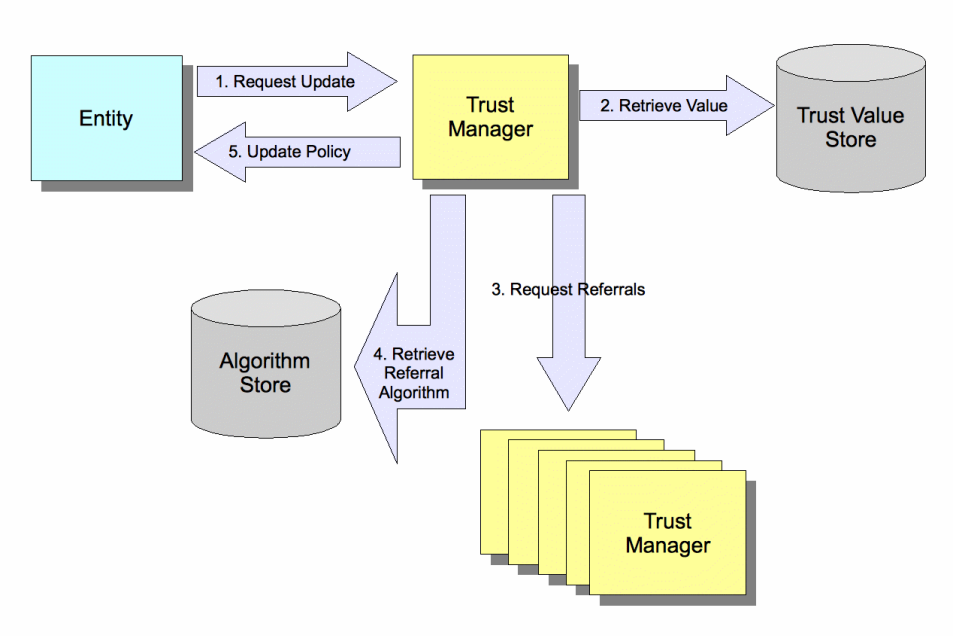


Figure 3.18: Policy Updates

3.8 Scenarios

In this section we will examine two scenarios which will describe how the trust model will be used in practice. The first scenario examines how the trust model described here can aid the Transaction Model [14] in the prioritisation of service selection for alternate sequential coordination. The second scenario examines how the trust model can be used to derive trust values in conjunction with the Rating Agency model described in section 3.1.2 above.

3.8.1 Distributed Transaction Scenario

Alice sells goods through an online web store. She uses the OPAALS Transaction Framework to manage her workflow. In this transaction context she uses services from several different providers including payment delivery and an SMTP mail delivery service to deliver confirmation of orders and tracking numbers to her clients. When Alice initially sets up the transaction context she is aware

of three different SMTP services. As she has never used any of these services in the past she requests from the trust manager a reputation rating for each of the services in the context of “email delivery”. All three services are present in the transaction context coordinated as a sequential alternate in order of the reputation . This means that the coordinator will try each service on order of the sequence until one succeeds. The order of the services is initially based on the reputation values that the Trust Manager has provided Alice with. Alice creates a trust algorithm which will take a transaction log as input and output an updated transaction context as a result.. The algorithm is designed to reduce trust if the service is unreliable in a combination several aspects (e.g. availability, accuracy, timeliness) and these experience of these aspects (contexts) are reported in the transaction. Each time the transaction is executed the resultant transaction log is delivered to the Trust Manager which updates the transaction scenario document with the updated trust values. Before Alice performs the transaction again she requests an updated transaction context from the Trust Manager. The order of the the service execution now reflects the order of trustworthiness of that service according to the algorithm. In this way less reliable services will move towards the back of the execution chain and more reliable services will move to the front, improving the efficiency of the transaction execution over time.

3.8.2 Rating Agency Trust Scenario

Bob runs a travel agency and books flights and hotels for clients. When rating Hotels for recommendations, Bob uses a combination of client ratings and star ratings issued by third party rating agencies. No international standard of hotel ratings exist and each country has different rating systems and standards. Some countries have strict regulations where others are not consistent. Bob develops a referral based algorithm to evaluate trustworthiness of hotel ratings. For example, if Hotel Astra is given a 3 star (out of 5) rating by a Rating Agency called HRA (Hotel Rating Agency), this is a referral based trust evaluation, i.e HRA gives this hotel a trust value of 0.6 to deliver across a range of criteria. Bob recommends this hotel to a client who makes a booking. Bob has also developed a client feedback application which enables him to evaluate the level of satisfaction his customers have had from a hotel booked through his system. This feedback acts as an experience report for which Bob has developed a direct experience trust algorithm. When Bob receives this report from the client who used Hotel Astra. The experience report is delivered to the Trust Manager who retrieves the appropriate algorithm and updates the Trust Values for Hotel Astra. The Trust Manager also compares this value with any recommendations that it has received for the hotel (in this case from HRA). As discussed above in Section 3.3, the trust in HRA is an influencing factor in evaluating trustworthiness of the hotel actually delivering this level of satisfaction. Using a third algorithm the Trust Manager updates the Trust Value for HRA in the context of “hotel referral”. The recommendations by third party Rating Agencies evolve over time

towards a more true rating based on a combination of client ratings (direct experience) and Rating Agency recommendations (referral).

4 Conclusion

In this document, we have outlined final identity and trust models that are fully integrated and compatible, informed by the the state of the art in identity and trust, and our work on distributed, decentralised identity and trust in digital ecosystems. The theoretical and design discussions in this work form the basis of the software implementation, which will be described in D3.11, and integration with the OPAALS DE environment and OKS in workpackage 5 and 10.

Identity and trust combine to verify arbitrary identity claims, via the semantics of (SAML) assertions, in identity operations (detailed in D4.1, and clarified further in this work). Operations are generic, extensible and platform agnostic. They allow for multiple binding implementations, and are based on a modelling framework, amenable to non-technical identity and domain experts.

We have also described in detail the concept of Rating Agencies and how they can aid in the provision and evaluation of trust. We have described algorithms with various characteristics. The choice of algorithms is context specific. We have also given examples of how one of these algorithms (exponential average) performs under various conditions for both direct experience trust as well as referral based trust. We have described how our trust model integrates with the accountability and transaction models provided two scenarios to illustrate how our trust model will work in practice.

5 References

[1] McGibney, J. & Botvich, D., 2007. A Trust Overlay Architecture and Protocol for Enhanced Protection against Spam. In Proceedings of *The Second International Conference on Availability, Reliability and Security*. IEEE Computer Society, pp. 749-756.

[2] Koshutanski, H., Ion, M. & Telesca, L., 2007. Distributed Identity Management Model for Digital Ecosystems. In *Emerging Security Information, Systems, and Technologies, 2007. SecureWare* 2007. The International Conference on. pp. 132-138.

[3] Platzer, C., 2004. Trust-based Security in Web Services. Masters Thesis. Information Systems Institute, Distributed Systems Group, Technical University of Vienna. Available at: http://www.infosys.tuwien.ac.at/Staff/sd/DA/ChristianPlatzer.pdf.

[4] McGibney, J. & Botvich, D., 2008. A trust based system for enhanced spam filtering. *Journal of Software*, 3(5), 55-64.

[5] Aitken, D., Bligh, J., Callanan, O., Sint, P. P. and Sciences, A. A. O. 1997. Institutional rating in everyday life. In *Proceedings of the Delos Workshop on Collaborative Filtering*.

[6] Palmer, J. W., Bailey, J. P., and Faraj, S. 2000. The role of intermediaries in the development of trust on the WWW: The use and prominence of trusted third parties and privacy statements. *Journal of Computer Mediated Communications.*

[7] Sarkar, M. B., Butler, B., and Steinfield, C. 1995. Intermediaries and cybermediaries: A continuing role for mediating players in the electronic marketplace. *Journal of Computer Mediated Communication 1, 3*.

[8] Zucker, L. 1986. Production of trust: Institutional sources of economic structure. *Research in Organizational Behavior* 8, 8, 53–111.

[9] McKnight, D. H. and Chervany, N. L. 2002. What trust means in e-commerce customer relationships: An interdisciplinary conceptual typology. *International Journal of Electronic Commerce* 6, 2, 35–53.

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

[11] ITU-T. 2005a. The directory: Authentication framework - 08/05. ITU-T Recommendation X.509, available at http://www.itu.int/rec/T-REC-X.509-200508-I.

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

[13] OPAALS Consortium, *D4.3: Trust Model for the DE*, 2008, http://files.opaals.org/OPAALS/Year_2_Deliverables/WP04/D4.3.pdf

[14] OPAALS Consortium, *D3.5: Full Architecture Definition*, 2008, http://files.opaals.org/OPAALS/Year_3_Deliverables/WP03/D3.5.pdf

[15] McGibney, J. and Botvich, D., 2007, "Distributed dynamic protection of services on ad hoc

and p2p networks", in *Proceedings of 7th IEEE International Workshop on IP Operations and Management (IPOM)*, San Jose, CA, USA, Lecture Notes in Computer Science (LNCS) 4786, pp 95-106, Springer, November 2007.