

A Practical Approach to Identity on Digital Ecosystems Using Claim Verification and Trust

Mark McLaughlin and Paul Malone

Telecommunications Software & Systems Group,
Waterford Institute of Technology, Waterford, Ireland
mmclaughlin@tssg.org, pmalone@tssg.org
<http://tssg.org/>

Abstract. Central to the ethos of digital ecosystems (DEs) is that DEs should be distributed and have no central points of failure or control. This essentially mandates a decentralised system, which poses significant challenges for identity. Identity in decentralised environments must be treated very differently to identity in traditional environments, where centralised naming, authentication and authorisation can be assumed, and where identifiers can be considered global and absolute. In the absence of such guarantees we have expanded on the OPAALS identity model to produce a general implementation for the OPAALS DE that uses a combination of identity claim verification protocols and trust to give assurances in place of centralised servers. We outline how the components of this implementation function and give an illustrated workflow of how identity issues are solved on the OPAALS DE in practice.

Keywords: Identity, trust, digital ecosystems, federated identity.

1 Introduction

Central to the ethos of digital ecosystems (DEs) is that DEs should be distributed and have no central points of failure or control. This means that failures are always local and that system-wide authorities are not necessary for the ecosystem to function. On the OPAALS project, it has been a policy to re-use existing technologies and standards, where appropriate, in building the DE infrastructure in order to focus resources on what is truly innovative. These requirements have essentially shaped the identity model and implementation that have emerged from the project. Thus we have reused existing authentication and authorisation mechanisms, as well as federated identity specifications and technologies, to build a solution for identity on DEs, which are not centrally governed or maintained.

Traditionally, an identity was merely a username and a password (or perhaps an X.509 certificate) supplied by an identity provider (IdP) for a system, and it applied universally across the system. As a result, there is a continued expectation that in order to access a system or service, a user authenticates somewhere and then is fully identified to all parties on the system and can consume any service. This is what users, administrators and service providers want and expect.

The problem is that on a DE, we do not assume a centralised authentication or authorisation infrastructure. If a user authenticates, the user perhaps has access to local services, but not automatically to services hosted on other systems. Other contractual or trust based mechanisms are required to bridge the inter-domain gap. Clearly, authentication on a DE is only the first step, other mechanisms must then be used to assure service providers (SPs) in one domain of the validity of an identity asserted by an IdP in another.

In this paper, we state our motivation for supporting decentralised identity provisioning; we outline what we mean by identity on a DE, what this identity can be used for. We also outline what steps an entity must perform in order to be identified. We describe how entities can have an identity on a DE that is derived from an identity that they previously used to authenticate with for another purpose. We describe how the OPAALS identity model implementation, based on the open source project, IdentityFlow, can be used to provide this functionality. Finally, we describe how IdentityFlow integrates with Flypeer, the P2P infrastructure on which the OPAALS DE is based.

2 Motivation

DEs are devolved network environments with no central points of control or failure. The power structure inherent in particular digital systems is often reflected in the identity provisioning scheme adopted. For example, if identity is provided centrally, the centralised IdP will exert a large amount of control over the entire system due to its control over sensitive user data and the processes of authentication and authorisation. If identity is provided by a number of parties exerting only localised control, no single IdP will be able to exert control over the entire system. Therefore, our primary motivation is to devolve identity provisioning to a large number of localised IdPs whilst still maintaining the possibility of an identity that is applicable systemwide.

Fig. 1 gives a typical identity domain with centralised identity provisioning, in this case, Google. All of the Google services illustrated use a common identity that is backed by the Google IdP (not shown). The more services within that identity domain, and the more users consuming those services, the more power that is centralised in Google's hands by virtue of their ownership of the IdP and control over all aspects of identity provisioning. Microsoft attempted to create a global identity domain, administered by them, called Microsoft Passport. Passport offered the possibility of single sign-on (SSO) to all participating services on the web, which would have centralised a huge amount of power and responsibility in Microsoft's hands. However the service was cancelled due to concerns about user privacy and a change of strategy based around the user-centric ideals embodied in Kim Cameron's Laws of Identity[4].

Subsequent attempts to introduce SSO across heterogeneous service platforms focussed on the federation of identity domains rather than the expansion of one dominant domain. Federated identity allows identities on one domain to be used

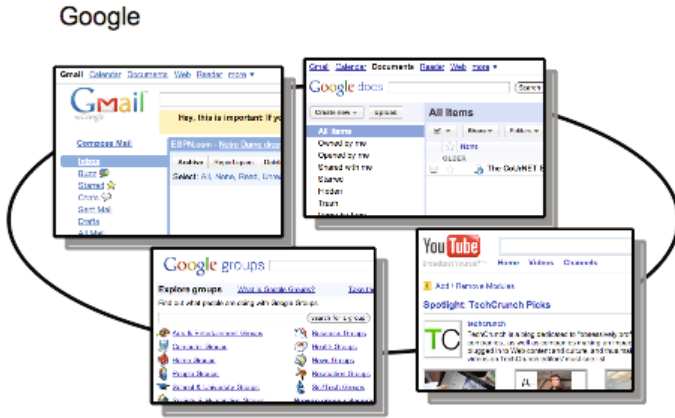


Fig. 1. An illustration of services in the Google identity domain

to access services on another. This is facilitated by inter-domain identity assertion protocols, such as SAML, and a business and/or legal agreement between the identity domains, specifying how identity attributes are passed between the domains. Fig. 2 gives a prominent example of a UK-based federation of academic identity domains, the UK Access Management Federation for Education & Research (UK AMF). Constituent identity domains host IdPs or SPs or both, allowing, for example, university students to authenticate with their university IdP and potentially use any service hosted by any university within the federation (e.g. online academic databases). Thus, although identity provision is local to the university, the federation agreement ensures that identities are applicable throughout the federation.

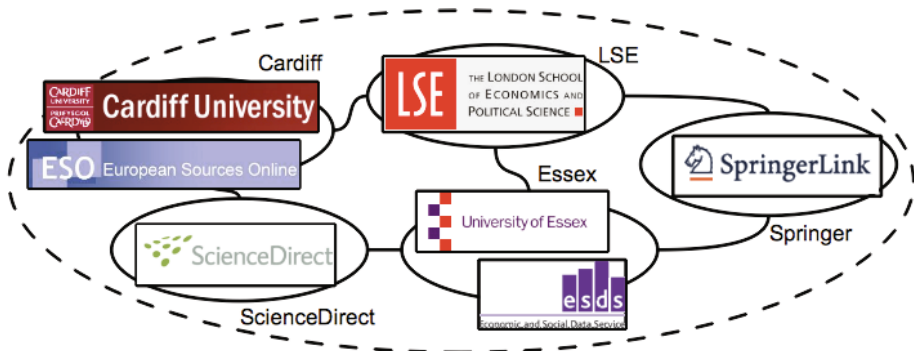


Fig. 2. Selected UK Access Management Federation for Education & Research domains

Although the federated domain scenario is a significant improvement on the large, monolithic identity domain scenario, there are still a number of issues that predicate against its immediate adoption in DEs:

1. All parties must sign up to a single agreement, which limits the scope and the scalability of the federation.
2. When federations become large, a single authoritative entity is required in order to moderate federation membership (e.g. Eduserv for the UK AMF).
3. Federation agreements are difficult to change because a consensus of all participants is required in order to enact proposed changes.

The focus of our work is to overcome these issues and arrive at a solution for decentralised identity provisioning in DEs, and to do this in such a way as to maximise the re-use of federation specifications and software. The particular approach adopted has been to replace the fixed federation agreements with contextual, transient agreements, based on measures of computational trust. These agreements strengthen and weaken in particular contexts based on the experiences of the IdPs and SPs in the DE, allowing unstable federations to be continuously created, destroyed and modified. Overlapping federations, or sets of federations with common participants, can potentially extend the applicability of identities to the entire DE (see Fig. 3), thus ensuring scalability and avoiding the need for authoritative entities.

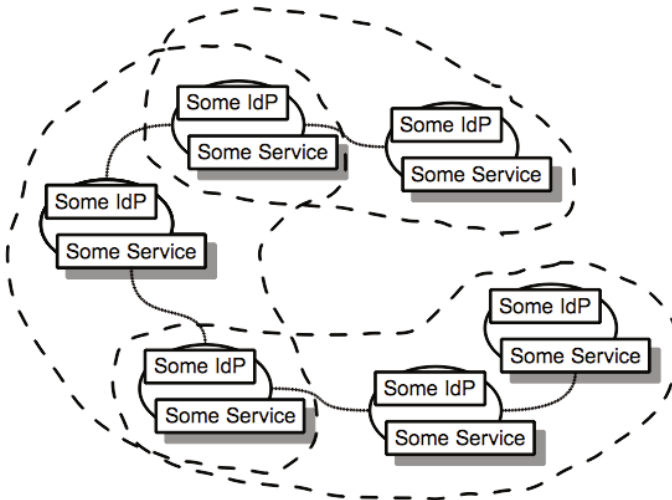


Fig. 3. Dynamic overlapping identity federations based on trust

The current work furthers this agenda by laying out in concrete terms how identity claims arising from a user-centric, local authentication can be asserted to other entities in a DE, and how these claims differ in scope from traditional, centralised or federated identity claims.

3 State of the Art

There is currently much work being conducted in the area of online identity, from a social as well as from a computer science viewpoint[7]. Many definitions have been put forward for digital identity[6],[4],[15],[24] and partial identities [8],[6] [5],[24]. When we talk of identities, we define them similarly to partial identities in [8],[6]: ‘that which represents a person in a particular context in the online world’, where not otherwise specified. There is also much work being conducted in the related areas of trust and reputation[10],[13]. In this work we use trust to refer to reliability trust in [10]. Federated identity[17] is concerned with federating previously separate identity domains, across large organisations and the enterprise, such that users in one domain can consume services in another. Federation specifications and protocols, such as SAML[1], facilitates federated identity management.

Digital Ecosystems can be described as “distributed adaptive open socio-technical system, with properties of self-organisation, scalability and sustainability, inspired by natural ecosystems” [3]. The term has been used in other contexts [2], however, we refer here to the body of research initiated under the heading of Digital Business Ecosystems (DBE), that is intended to promote the pervasiveness of ICT in SME and move organisations towards a “more, fluid amorphous and, often, transitory structures based on alliances, partnerships and collaboration” using biologically inspired metaphors [23].

Identity management in DEs draws inspiration and influence from the intersection of distributed/decentralised identity management[26],[17] and user-centric identity management[15],[18], and utilises the SAML federated identity specification. The implications of trust in these identity management configurations is explored in [11]. The initial work in DEs was performed by[16],[9], which form a precursor to the final OPAALS identity model, given in [21]. The identity model and implementation includes a generalised identity protocol framework (which includes SSO), and a number of bindings, including a JXTA¹ binding.

Trust has long been a topic of study in psychology, sociology, philosophy and economics; but in the nineties it has also found application in e-commerce, particularly in online markets such as eBay [25]. Trust can be described as “a directional relationship between two parties that can be called *trustor* and *trustee*,” [10] where a trustor is said to trust, or not to trust, a trustee, in a particular context. Trust can be used as a form of ‘soft security’ [10] or, by reflecting the real world social relations, as an enabler of “trade, competition, collaboration and so on” [25]. There are numerous models for computing trust and reputation² [25, 10] on various systems and networks, including decentralised P2P networks [19].

¹ JXTA is a P2P platform: <https://jxta.dev.java.net/>

² “The overall quality or character [of some trustee] as seen or judged by people in general.” [10].

4 Identity in Digital Ecosystems

An identity is something that is unique to an individual, or other entity, and is in some sense intrinsic to the entity in question. Traditionally, identities on computer systems have been represented by an identifier and a set of other attributes, describing that identity. The identifier is usually that attribute that is guaranteed to be unique systemwide. This guarantee of uniqueness is enforced by a centralised identity provider (IdP) on the system, which is responsible for maintaining a registry of all identities on the system.

In the real world, human identities are unique because no two human beings have identical biology and history. For the purposes of administration, the nation state will maintain identifiers and certain sets of attributes on behalf of each individual in the state, and it will guarantee the uniqueness of these by similarly using a centralised IdP or public service registrar(s). These attributes are linked to the real human being using some biometric data (e.g. a passport photograph) that are encoded along with them, as well as on identifying documents. However, in a DE, where there are no central points of failure or control, uniqueness of identifiers cannot be guaranteed, since there are no absolute identity authorities.

In the absence of a global identity authority, asserting the uniqueness of an identity, and maintaining a registry of unique identifiers, identity must operate on a different basis. Digital identities can no longer be unique, since there is no maintainer of a global identifier namespace. Our solution is to base identity on decentralised, inter-personal relationships, analogous to real life experience, where the following hold true,

1. Entities can be introduced to other entities.
2. Entities are recognisable to other entities.
3. Entities can make identity claims to other entities.
4. These claims can be verified by other entities.

In explaining 1., since no authority maintains the lifecycle of an identity, identities are effectively created, as far as other entities are concerned, by the process of ‘introduction’, analogous to the real world. Only when an entity has introduced itself, or represented itself to another entity with a given identity, can a history of interactions between the two begin. In explaining 2., entities must be ‘recognisable’ to other entities as being the entity that they have dealt with before, since they cannot rely on the system to ‘announce’ entity identities, which are accepted without question by the system as a whole. Recognisability can be affected by re-using the same public-private key pair for signing messages in a PKI³, or by re-using the same shared secret passphrase.

1. and 2. allows entities to introduce themselves to others, to be subsequently recognised by those parties, and therefore any two entities can build a shared history between themselves. This shared history between entities is a basis for the decentralised uniqueness of identity and as a surrogate for the uniqueness offered by a canonical registry. It can also, itself, be later used for recognisability, since

³ Public Key Infrastructure.

either entity can pose a question about their shared history that only the other entity can know the answer to. This shared history can also be used to evaluate trust between entities. Trust can then be used as a basis for accepting identity claims from asserting entities.

In explaining 3. and 4, we note that there are a number of identity claims that may be made about a subject, pertaining to issues such as ‘age’, ‘address’, ‘fingerprint’, ‘portrait photograph’, or ‘occupation’, such as “john murphy’s fingerprint is represented by the following image” or “user longday11 is over the age of 18”. These claims were formerly asserted by the ‘system’, however in decentralised environments, we must use a federated identity-like process of claim and claim verification, where any entity can make a claim about another entity and other entities can attempt to verify those claims. However, in the absence of a federation agreement, there is no straightforward mechanism for evaluating when the claims made by others are trustworthy. Our approach is to use computational trust and networks of transitive trust between entities in a DE, to evaluate the trustworthiness of these claims.

We discuss the process of claim and claim verification and their applicability to identifier claims in a DE in the proceeding two sections.

5 Identifier Claims from IdPs and the Role of Authentication

Here we discuss how identifiers, derived from authentication, can be used in a DE, even though there are no centralised authorities to back these identities or control the identity ‘namespace’, and therefore guarantee the uniqueness of these identifiers. The use of a pseudo-unique, ‘naïvely distinguishing’ identifier is still useful even though absolute uniqueness is impossible. For example, humans find it convenient to apply first name labels, such as ‘john’ and ‘philip’, even though these identifiers do not identify these individuals uniquely. The kind of identifiers we propose for DEs will be similarly provisional. The ability to verify that this ‘john’ is the ‘john’ that I know well means that I will not be deceived by a multitude of individuals calling themselves ‘john’, even though the name john is not unique.

These non-unique identifiers can be identifiers that are already used by an entity for another purpose, such as an email address (used by Yahoo! and Google), e.g. bob@bob.org, or an LDAP distinguished name, e.g. dn: dc=org,dc=bob,username=bob. If the IdPs for these identities are represented on the DE, then they can be used. However, these IdPs, which act as authoritative IdPs on their respective domains, are not authoritative on the DE, and therefore the identifiers that they are responsible for are only naïvely distinguishing on the DE. These identifiers are in effect self-asserted by entities, since they can choose an arbitrary IdP to ‘assert’ them on their behalf. (They could in fact assert their own identifiers if they had the functionality of an IdP.)

In order to make use of these identifiers, authentication mechanisms will usually be used, allowing entities to authenticate with their IdPs. Authentication may appear pointless when all identifiers are non-unique and self asserted, but it does serve the

purpose of binding an entity to its IdP in an implicitly strong trust relationship, and provide a strong basis for trust between that entity and other entities using the same IdP. Hence authentication can be of benefit in building out a trust network. It also provides the benefit of outsourcing IdP functionality, such as generating and verifying identity assertions, to third parties equipped with this functionality, which also allows identities on legacy systems to be re-used, if the IdP in question has been modified to interoperate on the DE.

In order to refer to other entities, it is clear that some kind of identifier is required by referring entities, whether it is unique or not, however authentication is not a strict requirement for obtaining this identifier.

6 Verifying Identity Claims with Identity Operations

In federated identity, identity domains are federated such that identity claims made in one domain can be accepted in another. Single Sign-On (SSO) is the process by which a user can sign-on in one domain and consume services on the other domain using the same login or username. From a technical point of view, SSO is a protocol flow between the user, the service provider (SP), and the user's (domain) IdP, in which the IdP passes identity assertions to the SP, often via the user (see Fig. 4). These assertions are accepted by the SP because there is a federation agreement in place that allows IdP and SP to trust each other in the context of a certain set of identity claims. The process has been adapted to make identity claims possible in a DE.

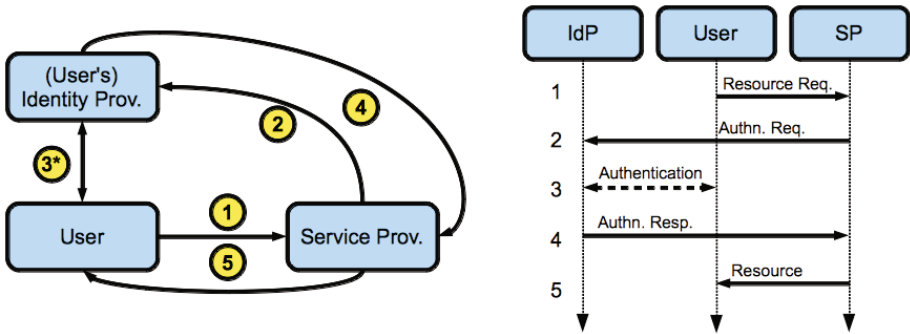


Fig. 4. High level view of federation SSO protocols (e.g. SAML)

IdentityFlow is a SAML-based, programmable framework for building arbitrary protocol flows for verifying identity claims. It is capable of operating in a DE composed of heterogenous platforms, by allowing separate bindings to be specified for portions of the overall protocol flow. The approach taken is to model DEs as a set of unstable federations between overlapping identity domains. However, rather than relying on federated agreements to provide trust between IdP and SP, we propose the use of a trust network for deriving explicit trust evaluations between

two entities in the context of asserting identity claims. These trust evaluations would change over time based on the experiences of entities dealing with would-be trusted entities, as well as the referrals of other trusted entities. As a result, unstable federations are constantly made and unmade, and the claims supported by these federations are trusted or not trusted accordingly.

IdentityFlow allows us to modify the standard SAML SSO profile to include an extra actor, the SP’s IdP. As a convention, we say that all entities have an IdP that is capable of generating and consuming identity assertions on its behalf, even if the entity can behave as its own IdP, just as the SP does in accepting assertions in the standard SAML SSO profile. This allows us to consider the trust between IdPs only, in the context of asserting identity claims, as the criteria for whether claims can be accepted. We term *IdentityFlow* protocols that use unstable federations based on trust to assert identity claims, *Identity Operations*. Identity operations were introduced in [21], and are based on SAML-like profiles and bindings.

A basic, but typical identity operation protocol flow is given in Fig. 5, which is equivalent to the typical federation SSO protocol flow illustrated in Fig. 4 except the SP is split into ‘Entity B’ and ‘Entity B’s IdP’.

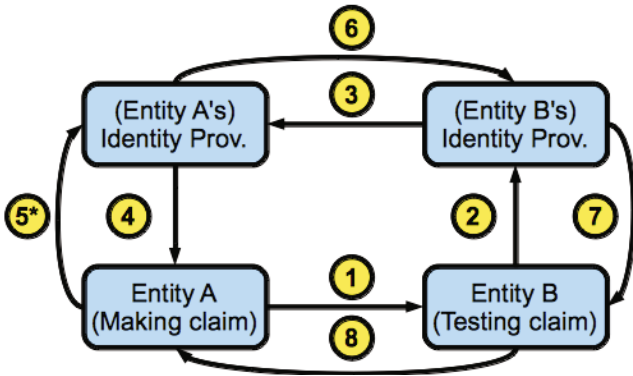


Fig. 5. Basic identity operation protocol flow

7 The Role of Trust in Identity Operations

During the execution of the protocol flow there will be appropriate points at which to measure the level of trust relationships to ensure they are sufficient for the operation to succeed, since messages from untrusted parties are useless. Fig. 6 gives an identity operation triggered by a service request. The User claims a given identifier, ‘real world’ identity or makes some other identity claim, and the SP verifies the claim before service access is granted. We see that the SP’s IdP makes a call to the User’s IdP (connection 3) and later receives a response (connection 6); it makes sense to test the level of trust that the SP’s IdP has in the User’s

IdP in the context of making identity claims, prior to making connection 3, since other connections are redundant if there is insufficient trust to accept the response sent in connection 6. Trust checks are particularly important where it is felt that trust relationships are non-absolute and/or evolving over time.

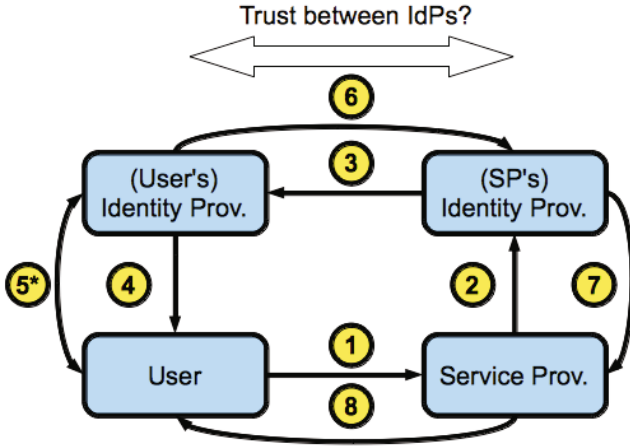


Fig. 6. Identity operation triggered by service access conducts trust checks

Trust ratings are conveyed in (trustee, context) pairs, where trustee is the trustee and context is the trust context. A component called the trust manager records ‘performance’ trust ratings based on direct experience and is capable of gathering ‘referral’ trust from third parties. Referrals are conveyed from an entity with performance trust in the trustee back to the trustor. Trust managers have the following functions,

1. Maintain a set of trust ratings with entities with whom the trustor has direct experience.
2. Discover trust transitive paths between trustor and trustee in the given context.
3. Aggregate these paths using appropriate strategies and algorithms to produce a trust rating.
4. Be capable of checking the integrity and authenticity of referrals from referees on the trust paths.
5. Have some mechanism for updating trust ratings based on experience.

2. is a challenge in decentralised environments, or otherwise, where a virtual trust network must be traversed in order to discover trust transitive paths. A number of options for discovering such paths in decentralised networks with various topologies are given in [22]. A scheme for aggregating trust paths and using belief calculus to produce a compound rating is given in [12]. [14] gives the

rationale and methodology for verifying the integrity and authenticity of referrals. Entities can update their trust ratings, and strategies for deriving trust ratings based on experiences from interacting with other entities. ‘Experience reports’ can be automatically or manually generated and submitted to the trust manager to derive an updated trust rating, according to some subjective scheme. The processes and algorithms for trust evaluation are described in [20]. The open source project *Trustflow*⁴ is actively developing an implementation of a trust manager for use in P2P environments.

We leave a discussion of appropriate strategies for generating and incorporating experience reports on the OPAALS DE and the specification of the trust manager for a future work.

8 A Practical Approach to Identity on DEs

Thus far we have expressed a set of requirements and potential strategies arising from a critical appraisal of identity in DEs vis a vis identity in centralised and federated environments. In this section, we will outline a workable solution for asserting identity claims in the OPAALS DE environment.

The OPAALS DE environment is based on *Flypeer*⁵, which in turn is based on JXTA. Two main steps are outlined for verifying identity claims on the OPAALS DE.

Step 1: Identifier Claims from Authenticating IdPs

JXTA provides a MembershipService framework for authenticating to Peer-Groups, which *IdentityFlow* implements as an extension to its core functionality. This allows *Flypeer* to authenticate users to the main Flypeer group. The IdP- MembershipService, which *IdentityFlow* implements, is available to each node on the JXTA network and operates as follows,

1. A set of login parameters, and the JXTA advertisement of a preferred IdP, is supplied via Flypeer as a JXTA Authenticator on the authenticating node.
2. The IdPMembershipService contacts the IdP node and supplies it with the correct authentication parameters.
3. The IdP returns an authentication result to the authenticating node, allowing the IdPMembershipService to complete its PeerGroup ‘join’ attempt.
4. If authentication has been successful, the IdPMembershipService generates a credential from the result passed back from the IdP, which acts as proof that the entity’s identity is indeed asserted by the IdP.
5. This credential can then be used as a means of filtering access to services in a JXTA-based environment.

⁴ TrustFlow, <http://sourceforge.net/projects/trustflow/>

⁵ <http://kenai.com/projects/flypeer>

The scheme for generating and using the credential assumes that all IdPs have a public-private key pair, and is as follows,

1. Following a successful authentication attempt, an IdP will sign the 'claimed identifier'⁶ with its private key.
2. The identifier, the IdP's public key, the encryption method and the signed identifier are packaged with other metadata into a JXTA Credential object.
3. Other entities can verify that the IdP in question authenticated the authenticating entity by obtaining the Credential object and using the IdP's public key to decrypt the signed identifier and comparing it with the claimed identifier.

The Credential implementation used by *IdentityFlow* marshals and unmarshals to and from XML. An edited (for space) example of a marshaled Credential is given below.

```
<?xml version="1.0"?>
<!DOCTYPE jxta:Cred>
<jxta:Cred type="jxta:LocalCoordinatorCred" xml:space="..." xmlns:jxta="http://jxta.org">
<PeerGroupID>urn:jxta:uuid-FEF..102<PeerGroupID>
<PeerID>urn:jxta:uuid-596..403<PeerID>
<SignedPeerID>MCwPf4i...K302oEPw=</SignedPeerID>
<SignAlgorithm>DSA</SignAlgorithm>
<GroupPublicKey>MICC...MYboJk=</GroupPublicKey>
<Username>identityflow@Guigoh</Username>
</jxta:Cred>
```

Step 2: Verifying an Identifier Claim

Identity claims are verified using identity operations, as discussed in sections 6 and 7. The identity operation specifies⁷

1. A set of contingent ordered connections that are executed in sequence (from one actor to the next), potentially with conditional logic affecting the protocol flow.
2. A set of actor tasks for each possible state of each actor in the protocol flow, that are executed when an actor assumes a particular state.
3. A binding, or set of bindings, that specify connection transport functionality in particular environments (e.g. JXTA binding).
4. A set of trust checks that must be made at appropriate points during protocol execution to ensure that there is sufficient trust between actors for the operation to succeed.

Identity operations are typically triggered by resource access, where it is necessary for entity identity claims to be accepted by the SP before the accessing entity can

⁶ This is currently the PeerID of the authenticating entity, but the entity's public key would be less spoofable.

⁷ A detailed specification of identity operations is beyond the scope of this work.

be granted access. Claims concerning identifiers issued by IdPs are important examples of claims that must be verified. In the particular case of the OPAALS DE, operations that verify user identifiers are simplified by the fact that each user propagates a credential which effectively encodes an identifier claim made by the user's IdP on behalf of the user. Therefore, only connections 1, 8 (service request and response) and 2, 7 (identifier claim verification request and response) in Figs. 5 and 6 need to be executed.

For all other identity claims, pertaining to any property that an entity can claim to possess and that its IdP can verify, identity operations are necessary. Examples of other claims are "I am over the age of 18" or "I am a member of the OPAALS consortium" or "My real name is John Murphy". Whether or not these claims are accepted by other entities will depend on whether the accepting party's IdP trusts the claiming party's IdP in the context of asserting identity claims.

Since the OPAALS DE is a JXTA-based environment, a JXTA binding is used, which means that all operation connection messages (by current convention) are passed via JXTA pipes. Using JXTA transport functionality ensures entities operate in a pure P2P overlay, where entities can be contacted by name without the need for knowing a peer's underlying transport address (i.e. IP address); and also provides transparent firewall traversal.

Identity operations will tend to be designed to address a particular need, such as identifier verification, and will tend to be triggered by an actor seeking to verify a claim during the course of some activity. Therefore, it is anticipated the development of identity operations will be in response to the needs of SPs. However, it is likely that the template of the simple operation given in Fig. 5 will suffice for most purposes. An example of a more complicated identity claim would be a situation where an SP would not accept a particular claim unless two or more IdPs asserted the claim on behalf of the user. This would necessitate a more complicated protocol flow involving 5 rather than 4 actors. In general, if any of the items in the list above change, modification will be required, which should be simplified by the modular, extensible design chosen by *IdentityFlow*.

9 An Example of an Identity Workflow on the OPAALS DE

A user logs into his *Flypeer* node by specifying his work chat server node as his IdP and his usual username and password (e.g. using the authentication mechanisms of the XMPP⁸ chat protocol). The chat server is already on *Flypeer* and can act as an IdP on the DE because it has implemented *IdentityFlow* JXTARequestInterceptors, which intercept *IdentityFlow* messages. The user's local node supplies the authentication parameters to the IdP and the IdP node authenticates the user remotely by passing these parameters to the backend XMPP chat server. If successful, the IdP will then create an AuthenticationResult, which will be passed back to the user's node. The user's node will then create a Credential object from the AuthenticationResult. Because the IdP node has

⁸ <http://xmpp.org/protocols/>

signed the claimed identifier it has returned to the user node, the user cannot choose another identifier for this Credential, because he will be unable to verify that his IdP signed it.

Once the user has authenticated, and has generated a Credential, he has general access to the DE, for discovering other users and services, and so on. Unlike a centralised system, the user does not have automatic access to resources determined by centrally maintained access control lists, permissions or an authorisation service. Instead, he has provisional access to resources determined by the parties to whom those resources are assigned and depending on whether sufficient trust between the user and those parties exist. Similarly, identities are not global and absolute, but merely local (on the user's trust network) and provisional (in that trust levels can change).

At some point the user attempts to join a collaborative document editing session, coordinated by an SP node. This SP node has an IdP that is responsible for verifying identity claims. When the user attempts to join the session, the SP triggers an identity operation to verify that "the user's claimed identifier (stored in the credential) is asserted by a trusted/competent IdP". The SP triggers the operation by requesting its IdP to verify this claim. The SP's IdP examines the user's Credential and retrieves the user's IdP's public key; the SP also retrieves the PeerID of the user's IdP that was sent as part of the user's request to join the session. The SP's IdP checks that the claimed identifier in the Credential was signed by the owner of the public key by decrypting it with the public key and matching it to the unsigned string. The SP's IdP then does a trust check on the user's IdP's in the context of asserting identity claims. The trust check indicates that it currently has sufficient trust in the user's IdP to proceed. If the SP's IdP does not already know the user's IdP's public key, it will contact the user's IdP to request it, and conduct a 'signed conversation' to prove that it is the true owner. Once the SP's IdP is satisfied, it will send a response to the SP informing it that the claim regarding the supplied identifier is verified.

If the SP were to later decide that only users over the age of 18 were allowed to join a particular document editing session, it could specify this to the users, and it could inform them that for the purpose of verifying age, it (or rather its IdP on its behalf) would only accept assertions from one particular IdP. This IdP might take measures to ensure that the users registered on its system are the age that they report themselves to be. In this case, users could register with this new IdP, and use this different IdP as their IdP when dealing with the SP's IdP in the context of asserting age claims. The SP's IdP will verify these claims by starting an identity operation, which entails performing a trust check on the user's IdP in the context of age claims, which ought to succeed since the SP's IdP recommended the user's IdP in the first place; and then initiating/continuing the protocol flow, as illustrated in Fig. 5.

10 Conclusion

In this paper, we state the critical differences between identity in a centralized environment and identity in a decentralised environment, such as a DE. These

differences are so fundamental that they invalidate many accepted assumptions, such as the belief that an identifier, or name, can be unique on a decentralised system, or the belief that one can authenticate with a decentralised system and obtain a globally applicable identifier. (Unfortunately, the JXTA MembershipService framework itself tends to give this impression.) What can be achieved however, is a kind of ‘local’ uniqueness, based on the principle that entities that have been encountered before can be recognised again in the future, and that entities can form themselves into a trust network, which adapts to entity behaviour towards other entities, good and bad. This makes it possible to build trust into recognised entities, and to use the trust network to propagate referrals such that the benefits of experience can be shared.

We also give a sketch of how our solution to identity problems are being implemented on the OPAALS DE. We recommend a two step approach to identity problems in general: i) users authenticate against an IdP of their choice, ii) SPs trigger identity operations to verify identity claims that are necessary for allow users access to services. *IdentityFlow* has been integrated into *Flypeer* from version 0.6, which allows users to authenticate to IdPs using a number of authentication mechanisms, i.e. LDAP, XMPP and Guigoh⁹. *IdentityFlow* has provided a framework for building operations since the start, and has added HTTP GET/POST and JXTA bindings. A template operation has also been provided that is akin to that illustrated in Fig. 5.

We are currently in the process of integrating trust checks into operations, and in general integrating *IdentityFlow* and *TrustFlow*. We intend to give the full identity operation specification, the trust manager specification, and strategies for generating experience reports and updating trust evaluations in future works.

Acknowledgment

This work is funded by the EU FP6 Network of Excellence OPAALS, <http://www.opaals.org>

References

1. OpenSAML v2.0, <https://spaces.internet2.edu/display/OpenSAML/Home>
2. Briscoe, G.: Digital Ecosystems. Ph.D. thesis, Imperial College London (2009)
3. Briscoe, G., De Wilde, P.: Digital Ecosystems: Evolving service-oriented architectures. In: Conference on Bio Inspired Models of Network, Information and Computing Systems. IEEE Press, Los Alamitos (2006), <http://arxiv.org/abs/0712.4102>
4. Cameron, K.: The laws of identity, <http://www.identityblog.com/?p=354>
5. Damiani, E., di Vimercati, S.D.C., Samarati, P.: Managing multiple and depend-able identities. *IEEE Internet Computing* 7(6), 29–37 (2003)
6. Glasser, U., Vajihollahi, M.: Identity management architecture. In: IEEE International Conference on Intelligence and Security Informatics ISI 2008, pp. 137–144 (2008)

⁹ <http://www.opaals.org.br>

7. Halperin, R., Backhouse, J.: A roadmap for research on identity in the information society. *Identity in the Information Society* (2008)
8. Hansen, M., Berlich, P., Camenisch, J., Clau, S., Pfitzmann, A., Waidner, M.: Privacy-enhancing identity management. *Information Security Technical Report* 9(1), 35–44 (2004)
9. Ion, M., Danzi, A., Koshutanski, H., Telesca, L.: A peer-to-peer multidimensional trust model for digital ecosystems. In: 2nd IEEE International Conference on Digital Ecosystems and Technologies, DEST 2008, pp. 461–469 (2008)
10. Jøsang, A.: Trust and reputation systems. In: *Foundations of Security Analysis and Design IV* (2007)
11. Jøsang, A., Fabre, J., Hay, B., Dalziel, J., Pope, S.: Trust requirements in identity management. In: *Proceedings of the 2005 Australasian workshop on Grid computing and e-research*, vol. 44, pp. 99–108. Australian Computer Society, Inc., Newcastle (2005)
12. Jøsang, A., Hayward, R., Pope, S.: Trust network analysis with subjective logic. In: *Proceedings of the 29th Australasian Computer Science Conference*, vol. 48, pp. 85–94. Australian Computer Society, Inc., Hobart (2006)
13. Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. *Decision Support Systems* 43(2), 618–644 (2007); *emerging Issues in Collaborative Commerce*
14. Jøsang, A., Pope, S.: Semantic constraints for trust transitivity. In: *Proceedings of the 2nd Asia-Pacific conference on Conceptual modelling*, vol. 43, pp. 59–68. Australian Computer Society, Inc., Newcastle (2005)
15. Jøsang, A., Pope, S.: User centric identity management. In: *Asia Pacific Information Technology Security Conference, AusCERT2005, Australia*, pp. 77–89 (2005)
16. Koshutanski, H., Ion, M., Telesca, L.: Distributed identity management model for digital ecosystems. In: *The International Conference on Emerging Security Information, Systems, and Technologies, SecureWare 2007*, pp. 132–138 (2007)
17. Maler, E., Reed, D.: The venn of identity - options and issues in federated identity management. *IEEE Security & Privacy* 6(2), 16–23 (2008)
18. Maliki, T.E., Seigneur, J.: A survey of user-centric identity management technologies. In: *The International Conference on Emerging Security Information, Systems, and Technologies, SecureWare 2007*, pp. 12–17 (2007)
19. Marti, S., Garcia-Molina, H.: Taxonomy of trust: Categorizing P2P reputation systems. *Computer Networks* 50(4), 472–484 (2006) (management in Peer-to-Peer Systems), <http://www.sciencedirect.com/science/article/B6VRG-4H0RYJ-1/2/b6a612053c7546bd311548c2b642c541>
20. McGibney, J., Botvich, D.: Distributed dynamic protection of services on ad hoc and p2p networks. In: Medhi, D., Nogueira, J.M.S., Pfeifer, T., Wu, S.F. (eds.) *IPOM 2007. LNCS*, vol. 4786, pp. 33–43. Springer, Heidelberg (2007)
21. McLaughlin, M., Malone, P., Jennings, B.: A model for identity in digital ecosystems. In: *Proceedings of the 3rd International Conference on Digital Ecosystems and Technologies (DEST)*, IEEE, Waterford Institute of Technology, Waterford, Ireland (2009)
22. Mello, E.R.D., Moorsel, A.V., Silva, J.D.: Evaluation of P2P search algorithms for discovering trust paths. In: Wolter, K. (ed.) *EPEW 2007. LNCS*, vol. 4748, pp. 112–124. Springer, Heidelberg (2007)
23. Nachira, F.: Towards a network of digital business ecosystems fostering the local development. Tech. rep., Bruxelles (September 2002)

24. Pfitzmann, A., Hansen, M.: Anonymity, unlinkability, unobservability, pseudonymity, and identity management a consolidated proposal for terminology. version 0.26 (2005)
25. Sabater, J., Sierra, C.: Review on computational trust and reputation models. *Artificial Intelligence Review* 24(1), 33–60 (2005), <http://dx.doi.org/10.1007/s10462-004-0041-5>
26. Weitzner, D.: Whose name is it, anyway? decentralized identity systems on the web. *IEEE Internet Computing* 11(4), 72–76 (2007)