

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

Mark McLaughlin & Paul Malone
Waterford Institute of Technology

3rd OPAALS Conference, Aracaju, Brasil.
22nd March, 2010



What do we normally mean by “identity”?

Essentialist approaches:

- Identity refers to the (unique) 'essence' of something.
- An identity is a set of attributes (where no two sets are the 'same').
- An identity is really only some unique number that I can use to refer to objects in my application.

Why is identity an issue on a DE?

There is no one identity or service provider (e.g. Google) out there that we can rely on to provide identity. (No single point of failure or control.)

So...

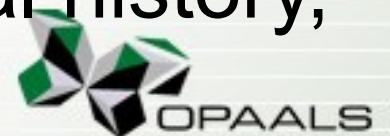
- No LDAP server out there to connect to.
- Nobody who can store user attributes.
- Nobody to even say how identities are represented and stored.



How can we view “identity” on a DE?

Modified essentialist approach based on a decentralised society & relationships:

- We still believe that there is a (unique) `essence', but we don't (necessarily) put a number on it.
- Entities can have different identities in different contexts (i.e. partial identities).
- My identity to you is based on our mutual history, on conversations, not on a number.



Decentralised identity requirements

- ***Introduction:*** we need a way of introducing one entity to another because entities are not objects in an IdP database.
- ***Recognition:*** we need to ensure that entities will recognise each other in the future since 'the system' will not act as intermediary.
- ***Claim Assertion:*** we need a way of asserting claims about an entity to other entities.
- ***Claim Verification:*** we need a way of verifying which claims should be accepted and which rejected.

What do these requirements give us?

- **If we can introduce ourselves and be recognised in future then we can build a mutual history between us.**
- **This mutual history can be used to reinforce recognition and provide a basis for trust.**
- **We can use SAML to make claims.**
- **If we have trust between recognisable entities, then we can determine whether we can trust the claims that entities make.**

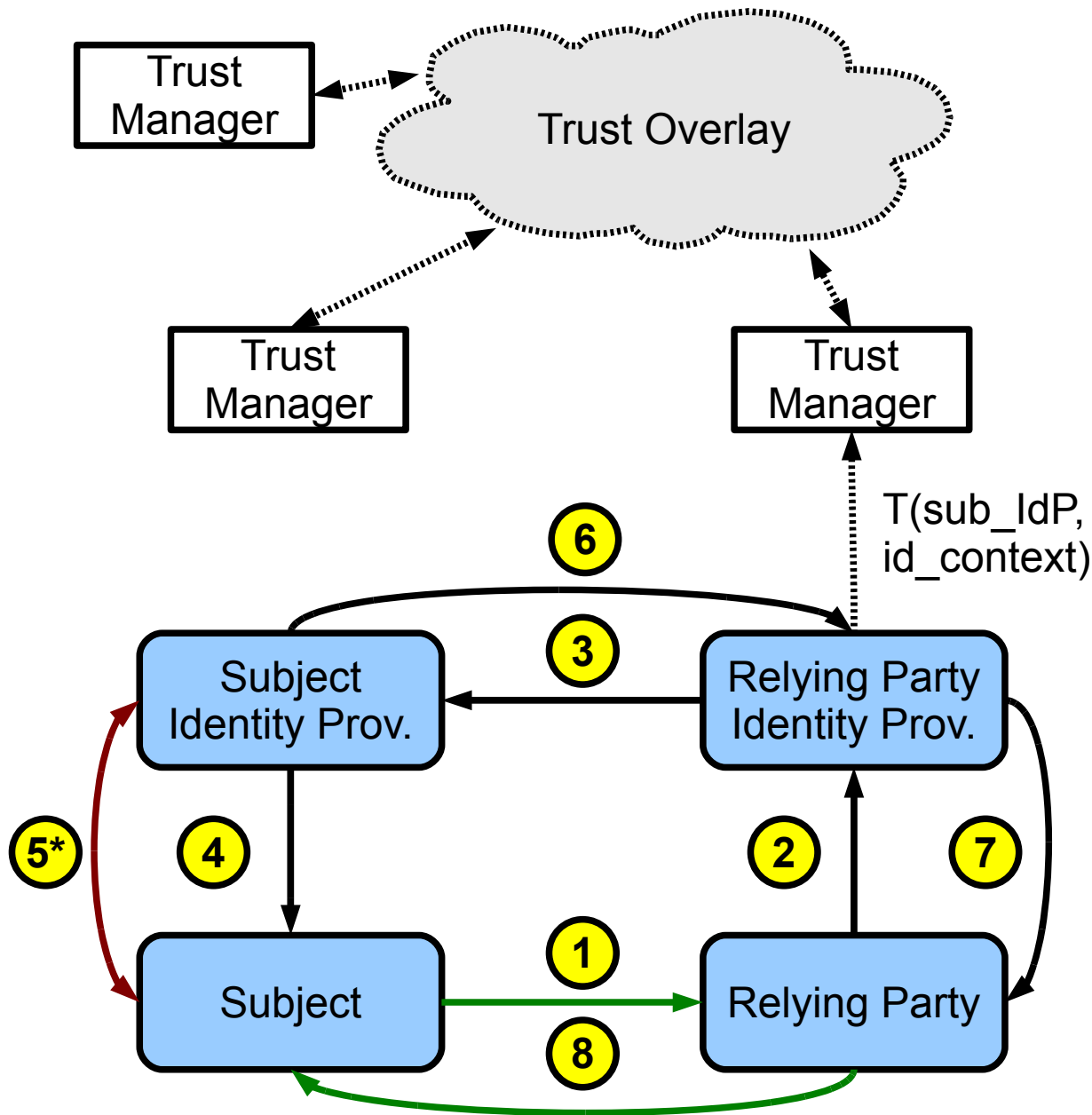
Getting from 'nice talk' back to software (1)

- In practice we use IdPs to authenticate entities, which generate Credentials.
- Entities can use an identifier like `mark@wit`.
- It's not unique, but other entities can go to the IdP called `wit` and verify that it has issued the cred.

Getting from 'nice talk' back to software (2)

- Essentially we have a system where claims are cheap, but we can always tell (recognition) whether an entity is the entity we know (introduction) or not.
- IdPs can assert other attributes as well, like address, age, qualifications, etc.
- Assertions are as strong as trust in IdPs.

The general case of verifying claims



Identity in the OPAALS DE

1. Subject requests resource from Relying party (RP).
2. RP requests id. assertion from his Identity Provider (IdP) for the Subject.
3. The RP's IdP requests the assertion from the Subject's IdP. (Provided it trusts the Subject's IdP – trust funct.)
4. If Subject has not authn. with his IdP, authn. is triggered.
5. Authentication step(s) if necessary.
6. Assertion returned from Subject's IdP to RP's IdP.
7. Assertion returned from RP's IdP to RP.
8. Resource access is granted to Subject by RP.

General conclusions on the identity work

- We have an entirely new way of looking at identity on computer systems, that holds in decentralised environments. (Relative addressing will be an important component.)
- We have some software in place to implement it.
- I am currently publishing this material with Gerard. Notably, we have been accepted for the Identity in the Information Society 2010 (IDIS10) Workshop, and will likely be included in the IDIS10 journal.

Thank you!

Questions?

