AU2EU

Techniques for assurance of claims

Vijay Varadharajan, Chun Ruan(MQ), Dieter M. Sommer,Daniel Kovacs, Patrik Bichsel(IBM), Peter Hannay(ECU), Sanjay Jha(UNSW)

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no: 611659

Project Information

Authentication and Authorisation for Entrusted Unions

Project number:	611659
Strategic objective:	ICT-2013.1.4.e
Starting date:	2013-12-01
Ending date:	2015-11-30
Website:	http://au2eu.eu/

Document Information

Title: Techniques for assurance of claims

ID:	D4.1.1	Type:	R	Dissemination level:	PU
		Month:	M14	Release date:	06.January.2015

Deliverable Description

Techniques for assurance of claims: Describes novel design/languages/techniques for specifying and verifying claims and the assurance levels of claims

Contributors, Editor & Reviewer Information

Contributors (person(partner): sections)	Vijay Varadharajan, Chun Ruan(MQ): Sections 1, 2, 3.1, 3.4, 3.5, 4.1.1, 4.2, 4.3, 4.4.1, 5.2, 5.3.3, 5.4 Dieter M. Sommer,Daniel Kovacs, Patrik Bichsel(IBM): Sections 3.2, 3.3, 5.1, 5.3.1, 5.3.2 Peter Hannay(ECU): Sections 3.6, 4.1.2, 4.1.3 Sanjay Jha(UNSW): Sections 3.1.2, 4.4.2, 4.4.3
Editor (person/partner)	Chun Ruan and Vijay Varadharajan/ MQ
Reviewer (person/partner)	Felix Gomez Marmol / NEC, Jerry den Hartog / TU/e,

Release History

Release number	Date issued	Milestone*	SVN version	Release description / changes made
1	10.11.2014	Proposed		Initial internal review
1.1	29.11.2014	Revised		Internal reviewers comments incorporated; released for external review.
1.2	30.11.2014	Released		
1.3	22.12.2014	Revised		Update after first period review
1.4	06.01.2015	Released		Resubmitted to EC

* The project uses a multi-stage internal review and release process, with defined milestones. Milestone names include abbreviations/terms as follows:

1. PCOS: Planned Content and Structure (describes planned contents of different sections);
2. Intermediate: Document is approximately 50% complete review checkpoint;
3. External: For release to commission and reviewers;
4. Proposed: Document authors submit for internal review;
5. Revised: Document authors produce new version in response to internal reviewer comments;
6. Approved: Internal project reviewers accept the document;
7. Released: Project Technical Manager/Coordinator release to Commission Services;

AU2EU Consortium

Full Name	Abbreviated Name	Country
Technische Universiteit Eindhoven	TU/e	Netherlands
Philips Electronics Nederland B.V.	PRE	Netherlands
Bicore Services B.V.	BIC	Netherlands
NEC Europe LTD	NEC	United Kingdom
IBM Research GMBH	IBM	Switzerland
Deutsches Rotes Kreuz	DRK	Germany
Thales Communications & Security SAS	THA	France
Commonwealth Scientific and Industrial Research Organisation	CSO	Australia
Edith Cowan University	ECU	Australia
Royal Melbourne Institute of Technology	RMI	Australia
University of New South Wales	NSW	Australia
Macquarie University	MQ	Australia

Table 1: Consortium Members

Table of Contents

List of Figures

1 Executive summary

Authentication and authorization in large scale distributed environments such as a cloud infrastructure involves generation and verification of claims from multiple parties. A claim is a statement that one entity makes about itself or another entity. A claim statement can be made about attributes such as a name, identity, address or key or property or capability of one or more entities. Assessing such claims is important to determine the confidence that can be placed upon them in the decision making. Hence the need for determining the level of assurance associated with the claims. Assessment will involve both validation of claims and the evaluation of trust on one or more entities that are vouching for the claims. Understanding the various types of claims, their specification and verification and determining the levels of assurance associated with them and the techniques that can be used to evaluate the assurance levels are the main objectives of this task. First this document describes state-of-the-art of review of claims based techniques and then addresses the use of claims in the specification of authentication and authorization. Then the document addresses the design issues involved in the evaluation of assurance levels associated with different types of claims in distributed security architecture. A claim statement involves the use of credentials both in terms of the entity which is making the claim as well as the entities that are validating and vouching for the claims. We investigate different types of claims and their use. We discuss credential based claims and how they are used in achieving authentication service. Then we describe how authentication claims are often pre-requisite to authorization where the decision as to whether an entity can access a particular service or resource is determined. We also discuss how claims play a key role in trust management in distributed systems. We outline the issues involved in the integration of authentication and authorization such as the need to restrict or minimize the number of claims dependent on the nature and type of authorization decision that needs to be made. We also address the use of trusted computing technologies that can be used to attest claims and describe different types of attestation such as property based attestation.

Then we address the claims languages and constructs that can be used in the specification and representation of different types of claims. We consider ABC4Trust-based claims, XACML-based policy claims and logic based claims and how they can take various forms such as authentication claims, policy claims, issuing and request claims. Having investigated the nature of claims and their specification using different languages, next we discuss the different levels of assurance that can be associated with the claims. In particular we investigate the different types of assurance characteristics and techniques that can be used to evaluate the assurance levels. In particular, we consider techniques for verification of claims, vouching of claims by trusted third parties and the use of trusted computing technologies for attesting claims.

An analysis of the use case scenarios in the AU2EU project and then presented and the different types of claims related to authentication and authorization are identified. This discussion highlights the need to evaluate the assurance level associated with the claims to derive the level of confidence that can be associated with them in the secure decision making whether it is authentication or authorization.

We discuss possible solutions for generating and verifying different types of claims in the security architecture, the assurance techniques that can be used and the levels of assurance that can be achieved. Finally we present a preliminary outline of design aspects that need to be considered when integrating claim assurance mechanisms into the security architecture.

We plan to expand this in the development and implementation of the architecture in the subsequent stages of the AU2EUproject.

2 About this Document

2.1 Role of the deliverable

This document describes different types of claims and techniques for assessing assurance of claims in the security architecture of the AU2EU project.

2.2 Relationship to other AU2EU deliverables

The use cases, D1.1.1, the authentication and authorization platform architecture, especially D2.1.1 and D3.2.1 are closely relevant to this deliverable.

2.3 Relationship to other versions of this deliverable

This is the first version.

2.4 Structure of this document

We present the state of the art in Chap.3. in which we introduce claims, and discuss various techniques proposed for claim specification and verification. We then discuss the challenges to security, privacy and trust occurring within the AU2EU use cases; we analyze use cases to identify claims related to authentication and authorization and the assurance of such claims in each scenario in Chap.4. In Chap.5 we propose our solutions in which we identify different types of claims used within the security architecture, address the techniques for generation claims using different languages, and the techniques for verification of claims and the levels of assurance that can be achieved. We also present different ways of integrating claim assurance techniques into the security architecture.

3 State of the Art

In this section we define the basic notions used in assurance of claims; claims, credentials and assurance. We then address authentication claim systems followed by methods to attest to claims. Finally, as an example, a health care scenario is analyzed.

3.1 Introduction

3.1.1 Claims

In a federated environment, it is needed to manage claim activities across multiple and heterogeneous security domains and autonomous systems, and deal with strategies for managing inter-domain behaviours. A *claim* is a statement that one entity makes about itself or another entity. The statement can be about a name, identity, key or capability etc, such as first name, last name, email name, roles or groups etc. If there is an entity that vouches for the claim, i.e. there is a credential associated with the claim, then we say the claim is certified. In a distributed computing environment, a claim can be made about the trust an entity has in another entity, trust delegation, or about whom the entity allows to access its resources. A claim based access control system will make access decisions by using the claims and its own logic. That is, claims are used by service requesters to gain access to resources, as well as by service providers to make decisions. Access to a given protected resource is determined by comparing the claims needed to access that resource with the claims associated with the entity attempting access. In Au2Eu, various claims can be made about request, authentication, authorization, consent, attribute issuing, ontology and environment etc. More details about this will be given in Chap.5.

A claim language is used by entities to express various claims. In general, a claim language should be intuitive and usable, expressive, and have well defined semantics. A well defined claim language should be able to express information such as:

- the issuer: from whom this claim is issued.
- the issuing place: from where the claim is issued.
- the context: under which context the claim is valid. This may be the platform, the application etc.
- the issuing time: at what time the claim is issued.
- the effective period: for how long the claim is active.
- the statement: this can be basically any fact about *attributes* (ie basic properties) of entities, access control information such as who can access what resource and in which way. This can also be the facts about trust relationship or any other facts.
- conditions: under which conditions the statement is true.
- purpose: for which purpose this statement is true.

3.1.2 Credentials

A credential is something that is proof of a claim one entity makes about itself or another entity. There can be different types of credentials associated with different types of entities. System entities typically include users, applications and platforms or hosts. Credentials can take a variety of forms such as identities, qualifications, competencies and attributes. They can be attested by third parties or self attested. In the case of users, the credentials include user identities and attributes of competence or ability to perform certain functions such as roles and clearances as well as biometrics such as fingerprints and retinal scans. In the case of applications, credentials include the identity of the application and certain attributes and characteristics associated with the applications such as the developer of the application or the version number and the type of platform it can run on. In the case of platforms or hosts, credentials could be the state of the platform, e.g. state attested by some trusted authority and the set of programs supported by the platform. For instance, in terms of business transactions, a user may not only wish to authenticate another user but also the device or the platform the user is using and whether this can be trusted or not.

Location and context can also play important roles in authentication and authorisation. The location information typically includes geographical and spatial data. The absolute location represents the location signature as geodetic location, which is basically a three-dimensional position comprising of latitude, longitude and altitude similar to GPS coordinates. Entity's relative location is a measure of its vicinity with respect to another reference/authenticated entity in an area. Moreover, there may exist variation in the closeness due to entity's mobility, thus demanding an infrastructure to detect the relative position as well as to calculate proximity. Proximity detection can be either entity/appliance-based or infrastructure-based provided that entity is capable of location sensing and inferring. Just like for any other credential, there may be a need for proof of location which can be provided by a trusted third party, which certifies someone's presence at a certain location at some point in time. In addition to the location, other contextual attributes include temporal information such as a time-stamp. This helps in the validation of entity's presence at a specific location and at a particular time. For example, an electronic notary function consisting of a location and a time can be included into a document to provide the evidence of its creation at a distinct location and at a specific time.

3.2 Claims-based Authentication

An *Authentication claim* is a statement that a user makes about herself or another entity, where that statement comprises information that can be used by the receiving entity in a decision process. For example, the claim can be about attributes such as the name, address, age, employer, job description, capability, or role. The strength of a claim lies in the fact that the statement may be certified by the third party, that is, there is an entity that vouches for the claim. We call this entity the issuer of the claim or the Identity Provider (IdP) and stress that a claim is a certified statement in such a situation.

In a typical authentication communication flow a Relying Party (RP) offers a service that a user wants to access. The RP requires the user to authenticate, that is, the RP requests a claim from the user. We can distinguish two possibilities of a user to obtain a claim. First, IdPs can vouch for users directly, that is, the issuer communicates the claim directly to the RP (i.e., service provider). In such case the assurance of the RP in the statement may come

from an authenticated channel used for the communication between IdP and RP. Note, the IdP is required to be available at the time of authentication of a user. Second, the user can obtain a *credential*, for example, a cryptographic signature on a set of attributes by an IdP, before the interaction with the RP. In such case, she may use the credential to derive a certified claim that is verifiable by the RP.

3.2.1 Online issued Claims

In the case of online IdPs that issue certified claims, the issuer retains the user's attribute values and when a user wants to authenticate, the RP and the IdP interact directly. We call such credentials 'online' as the IdP needs to be online for each transaction of a user. We can illustrate how online credentials work on the example of OpenID [17]. An OpenID provider, which may be seen as IdP who stores the user's attribute values, e.g., in a database. If the user wants to release any of her attribute values, she relates the RP to her IdP, i.e., her OpenID provider. The latter sends the attributes to the RP by using a secure, i.e., authenticated, channel to transfer the information. Based on the trust of the RP in the OpenID provider as well as the security provided by the communication channel, the RP derives the assurance about the communicated attribute values. Note that this information flow does not necessarily require cryptographically certified attribute information to be transferred.

3.2.2 Credential-based Claims

Credential technologies such as X.509-[18] or anonymous credentials [12] use a different approach. They add a certification value, i.e., a digital signature, to the credential. This signature allows the holder of a credential to create a proof that an IdP vouches for the attributes contained in the credential, i.e., the credential owner can create a claim based on the credential. It is important to note that such a claim can be created without involving the IdP at the time of authentication with a relying party (RP). Thereby, the IdP is not privy to the communication a credential owner has with relying parties, which is a privacy benefit for the relying parties. A further distinction w.r.t. online credentials is the fact using cryptography allows the RP to maintain a (verifiable) audit trail of its transactions. The assurance of the claim does not lie in system properties (such as the authenticated communication channel) but in the properties of the cryptographic algorithm (such as unforgeability of the signature scheme).

Standard credential technologies such as X.509 [18] offer the mentioned benefits in terms of assurance of claims on the verifier's side. However, more recent signature schemes allow us to extend the feature set of credentials, which makes them ideal for many more use cases. Anonymous credential system implementations, more specifically, Idemix [13] or U-Prove [10], do offer such credentials with additional features. . In essence, they allow a user to obtain a signature from an issuer on a number of attributes similar to standard certification technology. One important difference being that the issuer does not necessarily learn the attributes that it certifies. After a user has obtained a credential she can release the claim, i.e., the certified attributes, to a RP. In contrast to other certified credentials, the claim released to the RP is unlinkable to the credential generated by the issuer. That is, not even an issuer colluding with RPs can determine which user a claim comes from. In addition, not all attributes need to be shown for verifying the signature. Anonymous credentials enable a user to only release a subset of the attributes where the signature of the issuer can still be verified by the relying

party. This feature is called selective attribute [40] disclosure. Another important advantage of anonymous credentials lies in the fact that properties about attributes can be proven without revealing the attributes themselves. For example, using an anonymous credential containing a user's date of birth allows her to prove the certified statement that she is older than 21 years (provided this is indeed the case) without revealing the exact date itself. Note that while only very limited information about a credential is disclosed, the use of certified information achieves a high assurance at the verifying RPs.

An IdP may use tamper-resistant hardware in the scenario of credential-based claims as well. For example, an IdP may issue a smart card containing a cryptographic (base) credential that certifies a small set of values allowing the creation of a claim that (1) the smart card has not been tampered with, and (2) the RP can verify that the device has not been revoked. Due to such claim, the smart card may transmit further (uncertified) attribute values which are not cryptographically asserted/validated but instead inherit the assurance from the verified claim.

3.3 Authorization using Authentication Claims

We consider authorization to be the specification of the access rights of an entity, described by a set of attributes or properties, to certain resources for performing a given action. The enforcement of such specification we call access control. There are different authorization models that allow for the implementation of rules to enforce that only authorized entities are able to access stored data or, more generally, use any kind of resource. Some allow for a fine-grained specification of a subject's required attributes for being granted access to each resource.

The integration of authentication claims into any authorization framework poses a major challenge due to the difference in the assumed information flow. While currently implemented authorization systems expect all *possibly* relevant information on the user requesting access to be present, an authentication system based on claims requires that a user releases only the information minimally required for the operation at hand. Using this so-called data-minimization approach in an authorization system, consequently, requires changes of the current architecture or the communication flow. In an example of a current authorization system, we can assume a user being the son of an elderly person subscribed to a care provider may get read access to the medical information of his mother stored at the care provider. Traditionally, the authentication mechanism would make sure that the authorization system also knows that the very same person is a doctor, thus, he *also* may edit the medical information of all of his patients (possibly including his mother).

In an authorization system based on data-minimizing claims, the authentication operation performed in order to be authorized to read the medical data would only require the necessary information. It would not request the person to reveal that he is a doctor as this is not useful information for the current authorization decision. There needs to be a possibility to request additional claims extending the knowledge of the authorization system in case the person requests write access to data of any of his patients. Note, there are some limitations to the degree to which data-minimization can be applied in an authorization context. For instance, XACML allows for negative statements which deny requests in case a user possesses certain attributes. In such cases it is essential that all relevant attributes (those used in the conditions of such negative statements) are provided as otherwise the user may omit the ones that would prevent her from acquiring access.

On a high level we can distinguish two possible approaches for realizing *data-minimizing authorization*, where in terms of authorization we look in more detail at an XACML-based system as this is the only standardized authorization solution. The first approach uses existing extension points to achieve the integration, whereas the second one diverges from the traditional XACML information flow and requires changes in the architecture of the system.

Having discussed about authentication claims and authorization using the claims, we can now move to claims language.

3.4 Claims Language-based Policies

Over the years many systems have been developed to support claims- based policies. In this subsection, we will give a brief review of some of the important contributions. The first example below gives a brief description of the Microsoft Identity Model [31], in which claims are used to express access privileges. Then we consider the use of claims in identity and attribute management both in single domain as well as across multiple domains. The subsection on Trust Management considers the application of claims in a distributed environment where suitable trust models have been defined to establish and transfer of claims across different domains thereby achieving dynamic federation of claims. The subsection on logic method considers using logic- based languages to express various complex claims.

The Claims-Based Identity Model

Microsoft SharePoint and Windows Server 2012 are good examples of supporting claim-based fine-grained authorization. SharePoint assumes that if a user has at least one claim that is also assigned to a protected site or document, then the requested access is permitted. Windows Server 2012 has a language called SDDL (Security Descriptor Definition Language) which can be used to combine user claims and file classification information.

The Microsoft Identity Model [31] is a claims-based system. In this model a claim is the expression of an access right with respect to a particular value. An access right is read, write, or execute etc. A value is a database, a file, a mailbox, or a property etc. Claims also have a claim type, which decides the possible access rights over the values of this type. The claim type and access right are used for specifying capabilities with respect to the value. For example, if an entity has a claim of type 'File', with access right 'Read' over the value 'program.c', it then has read access to the file program.c. A claim of type 'Name', with access right 'PossessProperty' over the value 'Alice', means that the entity possesses a Name of 'Martin'. The Identity Model is extensible, allowing various systems to define additional claim types and rights as required.

3.4.1 Trust Management Language and Claims

In an open, decentralized system, the owner and requester often do not know each other, and the traditional identity based access control scheme becomes less effective. The trust management based approach to distributed authorization is thus proposed. In this approach, a requester submits a request, associated with a set of credentials issued by other parties, to a resource authorizer which manages the access control policies about the requested resource.

These access control policies specify what requirements need to be satisfied to be able to access a resource, such as what attributes are required and other conditions of access. The authorizer will then make a decision based on weather the credentials provided by the

requester satisfy the requirements. The policy specification language is used to express the distributed policy statements or claims.

PolicyMaker and KeyNote

PolicyMaker [8] and KeyNote [9] are two of the first trust mangement systems. PolicyMaker is an approach proposed by Blaze et al for trust management purposes. It uses a simple language for specifying trusted actions and trust relationships. The policy is specified in the form of a set of assertions, and each assertion is of the form:

```
source ASSERTS authorityStruct WHERE filter
```

Here source indicates where the assertion comes from, local policy or signed assertions. AuthorityStruct specifies the entities (represented by public keys) to whom the assertion applies, and the filter specifies the conditions that action string must satisfy. Each assertion states that the source trusts the entities to be associated with the actions that satisfy the condition.

Keynote is a successor to PolicyMaker. It tries to improve the PolicyMaker to be simpler, more extensible, and more expressive. In KeyNote, trusted actions are described as simple attribute/value pairs, assertion syntax is based on a human-readable "RFC-822"-style syntax, and predicates are based on c-like expresions and regular expressions. Credential signature verification is also built in to the KeyNote system.

Abdul and Hailes (2000)

Abdul-Rahman and Hailes [1] introduced a comprehensive trust model for virtual communities. From direct experiences and recommendations, each entity forms its trust. For the direct trust, they represent an entity's belief in another entity's (a) trustworthiness within a certain context (c) to a certain degree (td) by : t(a, c, td) where td =vt, t, u, vu, and vt, t,u and vu represents very trustworthy, trustworthy, untrustworthy, and very untrustworthy respectively, and the context c can be defined by the entity. For the recommender trust, i.e., the entity may also believe that another entity (b) is trustworthy to a certain degree (rtd) for giving recommendations about other entities with respect to a context (c), represented as: rt(b, c, rtd) They also present how to evaluate the direct trust and recommender trust.

Cabarcos et al (2014) and Tormo et al (2014)

Federation of trust has emerged as a key issue in an open distributed environment with multiple identity providers. In such an environment, typically users have multiple accounts and profiles in different domains and it is necessary to bridge trust between these domains, enabling identity information to be shared in a secure and trusted manner to improve usability and achieve greater interoperability. The paper [16] presents an identity reputation scheme that helps to achieve dynamic federation. Related to this, is the work in paper [41], which is concerned with the integration of reputation management with the OpenID protocol, which is an open standard for providing decentralized authentication to end users. This paper enhances the OpenID protocol so that providers can collect recommendations from users regarding different service providers and then provide appropriate recommendations to the users.

Carbone Trust Model (2003)

Marco Carbone et al [15]proposed a formal trust model in distributed networks and focused on the aspects of trust formation, evolution, and propagation. Principals' mutual trust can be

modelled as a function which associates to each pair of principals a trust value t, i.e., function m applied to a and then to b returns the trust value m(a)(b) T expressing a's trust in b. Each principal has a local policy which contributes to form the global trust via delegation. A policy expresses how the principal evaluates trust information based on not just his own beliefs, but also other principals' beliefs. a language for trust policies capable of expressing intervals, delegation, and a set of function constructions.

STRUDEL (2006)

Daniele Quercia et al [24] proposed a distributed framework STRUDEL, that contains a mechanism for the detection of malicious peers, a formal Bayesian trust model to assess trustworthiness, and a decision mechanism based on the maximisation of trust-informed utility. Principles keep track of others' reputation and team up with only trustworthy ones. This is done by means of a fully distributed Bayesian trust model that updates reputation data structures based on direct experiences and recommendations, and that produces trust assessments.

CARL (2010)

To tackle the problem that users commonly reveal more personal data than strictly necessary to be granted access to online resources, Jan Camenischa et [11] proposed an authorization language that allows for expressing access control requirements in a privacy-preserving way in a distributed system. The language is designed for specifying requirements on a user's cards to be used for obtaining access in any kind of open access control setting. The language explicitly specifies which attributes have to be revealed to access the resource, and distinguishes between the requirement to reveal a value and to satisfy a certain condition. It has defined the key word 'reveal' to enable an entity to describe the specific attributes that are required to disclose to some specific recipient. The key word 'where' enable entities to specify the conditions that an attribute should satisfy. The language can also specify the requirement for ownership of a card of a specific type by a specific issuer and the requirement of signature. The formal semantics of the language has been presented.

Privacy-ABC (2013)

Jan Camenisch et al [14] proposed a language frame work in XML schema that define and unify the concepts and features of privacy-preserving attribute-based credentials (Privacy-ABCs), and presented the API of a Privacy-ABC system that supports all the features we describe. The XML schema can specify credentials, which include various attributes associated with some identity id. It also covers issuance, presentation, inspection, and revocation, and supports pseudonyms and key binding. The framework makes it possible for application developers to specify security policies without having to deal with the underlying cryptography.

3.4.2 Logic-based Language and Claims

logic-based language has strong expressive power and well defined semantics and can be used to express various complex claims. Several schemes based on logic have been proposed with a view to formalizing authorization specifications and evaluations.

Woo and Lam(1992)

Woo and Lam [42] proposed an expressive language to authorization in distributed systems.

In their method, they consider structural properties inherent in authorization and provide formal semantics evaluation which is based on an extended logic program.

Abadi et al (1993)

Abadi et al. [3] proposed a modal logic -based approach for access control in distributed systems. Their work focussed on how to believe that a principal (subject) is making a request, either on his/her own or on someone else's behalf.

Jajodia et al (1997)

Jajodia et al. [25] proposed a logical language and illustrated how it can specify authorization, conflict resolution, access control and integrity constraint checking. It has the advantage to support different access control policies. Different policies can be specified on different objects, according to the needs of the users. The language supports groups and roles, and enable different rules to be specified to regulate the access control decisions.

D1LP and D2LP (2003)

In [29] another logic-based language has been developed, namely D1LP, to represent policies, credentials, and requests in distributed authorization. Their work focussed on how to express delegations based on an entity's attributes in an open system. D1LP is extended to D2LP to support negation and non-monotonic reasoning. It provides a predicate "overrides" which can only be used to specify priority information between rules.

RT_1^C (2003)

In [28], N. Li and J.C. Mitchell present a language called RT C1 which is based on constraint Datalog. The language allows first-order formulas in one or more constraint domains, which may define file hierarchies, time intervals, and so on, to be used in the body of a rule, thus representing access permissions over structured resources in a declarative language.

DKAL (2008)

DKAL [22] is a declarative authorization language for distributed systems which is based on existential fixed-point logic. It aims at expressive power to cater for complex policies by supporting the express features such as targeted communication, nested quotations and delegations. It allows expressions with unrestricted use of functions that can be nested and mixed, which was not allowed in many Datalog-based distributed authorization languages.

RBAP(2014)

Ruan and Varadharajan [35] proposed a logic-based framework that supports dynamic delegations for role based access control systems in a complex environments. It allowed delegation of administrative privileges for both roles and access rights between roles. It introduced the notion of trust in delegation and have shown how extended logic programs can be used to express and reason about roles and their delegations with trust degrees, roles' privileges and their propagations, delegation depth as well as conflict resolution. Furthermore, the framework is able to enforce various role constraints such as separation of duties, role composition and cardinality constraints.

Having presented various claims language, we now move to the attestation techniques for claims.

3.5 Attestation Techniques for Claims Verification

Trusted Computing Platforms or TCPs are a new genre of computing systems that support hardware based security. TCPs include a special hardware chip called the Trusted Platform Module (TPM) [39] . Using the TPM, a platform supports cryptographic functions and secure storage for secrets and data. But perhaps, one of the main features of a trusted platform is its ability to securely collect and report information about its own state. For example, a trusted platform can attest to a certain expected state i.e, the platform has all security critical functions running at a given time. This is achieved by the process of attestation. Attestation in trusted platforms includes two different phases. Phase 1 includes all the operations that correspond to the secure collection and storage of state information while phase 2 deals with safely reporting that information to a third party. The process of evidence collection starts at the time of platform boot. When the platform is booted, one small initial component that is inherently trusted measures (derives a hash) of the next program to be loaded and passes control to the second program. The second program measures the third program and so on. A bootstrapping process follows where all components of the platform are measured and their measurements are securely stored inside the TPM in special registers called Platform Configuration Registers or PCRs. In phase 2, the trusted platform reports its measurement to a third party. The third party is also provided with reference measurements which indicate the expected measurements for each of the components in the trusted platform corresponding to the good working state. The third party can then compare the reported measurements and the reference measurements to determine if the platform is in a good state and that the platform's state is acceptable.

This type of attestation is referred to as 'binary attestation' as information that is collected and reported about the state of a platform is in the form of binary hash measurements. Recently, the issues associated with the binary based attestation have become a much debated topic. One main reason being, binary measurements provide configuration and implementation details of the software components running on a trusted platform. This can potentially lead to security and privacy issues such as making finger printing attacks on the platform easier. Second, components keep changing all the time. Their configurations change, versions change and components are updated all the time. Each time a small change occurs, a new reference measurement has to be provided which dramatically increases the number of possible expected values for a component. Thirdly, binary measurements in some sense are indicative of the implementation of a component. This could favour some implementations over other. What seems to be more important is the ability to reason about security properties and functions of the component rather than its core implementation details. There are a range of different types of attestation approaches that have been proposed, and for each scheme, we provide a detailed analysis of its characteristics and limitations. In this context, we consider attestation in the form of derivation based attestation, delegation based attestation and enforcement based attestation. Here the contribution is that such an analysis will help to better understand the implications of these attestation schemes and the guarantees that they can provide when they are deployed in practice. The second contribution is to discuss the open problems and design issues that still remain when it comes to developing effective property based attestation schemes for distributed systems.

3.5.1 Property based Attestation

To address some of the issues associated with binary attestation as outlined above, the notion of property based attestation has been recently proposed. Property based attestation is founded on binary attestation but attests not to binary values but to security properties, functions or behaviour of systems. The main idea is that if a system can somehow prove that it is in a binary configuration that satisfies one or more security properties, then this information seems to be more useful to a verifier compared to hash measurements. The following are some of the advantages of using properties for attestation in trusted platforms. (a) Properties do not reveal implementation details of a system and can therefore hide system vulnerabilities. (b) Properties may not identify components and may provide a certain level of privacy. (c) Properties of components may not change as often as hash values particularly during updates. (d) Properties are easier to understand and can be useful to write meaningful access control policies rather than using a plethora of binary values. In this section, we discuss the different types of property based attestation.

3.5.2 Delegation Based Attestation

In delegation based attestation, one or more primary functions of property based attestation are delegated to a third party. The third party may either be trusted by the attester or by the verifier or by both. Subsequently, it may perform its functions on behalf of either or both of the parties. These functions include generation of mappings between binary values and properties, negotiation of what properties are required by the verifier and what may be disclosed by the attester and verification of an attestation report to determine if the attester satisfies a set of properties. In this section, we discuss four such delegation based schemes.

Certificate Based Attestation

Sadeghi et. al propose the idea of delegation based attestation in [36]. In this paper, they highlight the drawbacks of binary based attestation and propose a variety of solutions based on the existing trusted computing functionality. They leverage the existing TCG attestation mechanism and demonstrate how property based attestation may be realized on top of it. The model defines an 'ideal TC component' as one that is capable of evaluating a given system configuration for the properties the configuration provides. As an ideal TC component is hard to achieve, they extend the ideal model with a TTP that attests that a given platform S' fulfills a demanded property P'. A trusted TTP certifies that a configuration S' satisfies the property P' in a property certificate signed by the TTP. With the help of the property certificates, the authors move on to explore the different approaches that can be taken to realize the property based attestation mechanism.

The first approach is hardware based and requires extensions to the TPM component. TPM is extended to include a property certificate verification component that performs functions on behalf of a verifier. It verifies if the current system configuration of the attesting platform is the same as the configuration in the property certificate. If positive, TPM issues a new attestation certificate that confirms that the attesting platform provides the property in the property certificate. The second approach proposed by the authors is software based and requires a Trusted Attestation Service (TAS). TAS implements the extensions previously required to the hardware as software in a trusted way.

The main objective of this work is to realize property based attestation using property certificates. The authors define the notion of a property certificate that includes a mapping

between the system configuration and the system configuration. The system configuration information derived using TCG integrity report generation is a set of PCR values. The number of possible PCR values in a system can be very large depending on the number of components that exist with in the system and how often these components change. Therefore, it will be nearly impossible for a TTP to know all the possible PCR values of a system in order to issue a property certificate. An alternate approach of certifying individual properties of components may be considered using similar approaches as defined in this paper. However, when individual components are certified, a chain of property based certificates is required to maintain the chain of trust.

The authors also propose hardware and software based mechanisms to realize property based attestation. Including property based attestation features inside the hardware requires changes to existing standards and may increase the cost of the TPM. The software based approach on the other hand only solves the problem partially. With software based approach, it is still necessary to verify the integrity of the TAS, the isolation kernel and other related components that ensure secure paths and secure domain for the TAS, possibly using binary attestation.

Proxy based attestation

Poritz et. al propose a scalable and privacy friendly property attestation mechanism in [34]. The authors provide an attestation architecture that extends the TCG attestation mechanism using security properties of platforms. Attestation of properties between the attesting platform and a verifier is carried out only after the privacy policies of both parties have been satisfied. This helps to protect not only the privacy of the attesting platform but also the privacy of the verifier. The main components of the architecture include property certifiers, verifier, verified platform and a verification proxy.

- Property certifiers are trusted entities that certify the association of one or more properties with a certain component.
- The verifier is the entity that challenges a trusted platform for its system properties. The verifier includes it's property requirements in its security policy and details of the entities it trusts for signing and certifying property statements as trust policies. This may also include a privacy policy that determines to whom it wishes to disclose its security and trust policies.
- A verified platform is the trusted platform that attests its system properties to the verifier. The verified platform includes the properties that it can assure in its security policy and details of which entities to which such information may be disclosed as privacy policies.
- A verification proxy is the entity that mediates the attestation request and response between the verifier and the verified platform. The verification proxy performs a matchmaking and negotiation process of the verifier and the platform policies. It also performs the validation of the binary measurements of the verified platform using standard TCG verification mechanism. These measurements are then transformed into platform properties using the property certificates. The proxy is either deployed at the verified platform or at the dedicated machine.

The main aim of this work is to provide property based attestation while protecting the privacy of both the attesting and the verifying platforms. This is achieved by introducing a level if indirection in the form of a proxy that performs the privacy and security policy matchmaking and negotiation between the communicating parties. Policy verification may only be succeed if the same policy language and property semantics are used. In a distributed system where the attester and the verifier can often be strangers, this requirement may be difficult to guarantee. A mediation party may be required to translate policies between different domains.

Granularity based attestation

Nagarajan et. al propose a granularity based property attestation model in [27]. This seems to provide a first step to understanding what properties are and how property based attestation may be realized using property certificates in a distributed scenario. The authors argue that properties can exist in different levels of granularity. Based on the granularity level, the paper proposes a pyramid model. On the top of the pyramid, at level 1, are properties which are at the highest level of granularity. The lowest level, level n, has properties at the finest level of granularity. As we go down from level 1 to level n, the properties transition from high to low granularity levels. For example, on a 3 level pyramid, a property at level 1 may be confidentiality, property at level 2 may be encryption and property at level 3 may be AES algorithm. This means that an attesting platform can either reveal that it provides confidentiality service, or confidentiality using encryption or confidentiality using AES encryption algorithm.

The attesting platform must be able to choose to attest its properties at a level that is suitable to protect its privacy. Higher the levels in the pyramid, there is better privacy. Lower the levels in the pyramid, there is less privacy but better flexibility. A follow up paper [4] also describes how these properties can be used to in policy specification. It provides extensions to the XACML language to express policies using properties of systems at different levels of granularity. The authors also argue that it could be useful to group properties into 'aggregates' like compositions or property sets and map these sets to permissions. Changes to property requirements or granularity requirements then do not affect the property set-permission assignments.

The main aim of this work is to describe how property based attestation can be used in trust management systems for access control. It also divides properties at different levels of granularity for better policy expression and flexibility. This paper also assumes that property evaluation by a TTP is possible and properties can be guaranteed using property certificates. Although, this is a realistic assumption to make (as in the case of all other proposals), this is still a significant problem to be resolved. As seen previously, property based attestation by itself has challenges with respect to the semantics and the syntax of the properties when realized between entities of different domains. Having properties, and particularly a single property at different levels of granularities only complicates this problem as there is now a need to translate properties between different granularity levels. e.g how to translate a level 1 property in application domain A to level 2 property in application domain B. Also issues on how to distribute and revoke these granularity certificates need to be addressed. For example, will the same property at different granularities be distributed as different certificates? This will mandate managing more number of certificates and add complexity. On the other hand, if all the granularities of the property are defined in one certificate, then how would it be possible to attest with that certificate only the granularities of interest and hide the rest?

WS-Attestation

WS-Attestation [43] by Yoshima et. al extends remote attestation on a web services framework. It leverages the TCG attestation mechanism and provides a software oriented and dynamic approach for integrity reporting. This is achieved by extending the bootstrapping process for taking measurements inside a trusted platform. First, the core root of trust which is the trusted bios begins the measurement process and measures all components up to and including the operating system. The operating system then measures all the components up to and including the middleware of the web services framework. Then the middleware layer measures all data it loads or uses in the platform. This builds a transitive chain of trust from the bios to a web service application.

From the measurements, a structured data called the Platform Measurement Description (PMD) is created to include the platform state information. When a trusted platform is required to attest its state to a verifier, the platform presents its PMD to the verifier. The verifier then with the help of a trusted third party called the Validation Service (VS) validates these measurements. Validation of the PMD is a two-step process. First, TCG attestation verification is carried out on the binary measurements. Once the measurements are validated as true, VS then generates an attestation credential. WS-Attestation also proposes that for those system properties that may not be obtained using binary measurements, the attesting platform can use what are known as 'agents'. Agents are programs that reside inside the trusted platform and are required to report system properties. An example of such an agent is a local daemon that reads the system configuration files and adds to the PMD a structured message that describes the properties of the configuration (e.g., network setting, minimum password length, etc.). The other contributions of the paper are on the message exchanges between the attesting platform, the verifier and the validation service. There is also a prototype system that describes the implementation details of the proposed architecture using IBM's Integrity Measurement Architecture (IMA) [37].

This paper proposes the idea of translating PCR values to properties using a trusted third party called a validation service. The main objective of this is to protect the PCR values of the attesting platform from the verifier. This paper looks at property based attestation as a privacy enhancing mechanism for the attesting platform. Here the attesting platform has to place a significant amount of trust on the validation service to both keep the PCR values confidential and the verifier has to trust the validation service to only generate attestation credentials with valid properties. Also, the idea of using the agents to 'derive' properties at the attesting platform seems to be limited to reading high level information like a configuration files etc. It is unclear how more fine granular and rich information may be included in such attestation credentials.

3.5.3 Derivation Based Attestation

Derivation based attestation is based on how properties are 'derived' in a trusted platform. Here, there is no actual translation between binary values and properties. Instead, there is either a reference monitor or an analyzer that looks at the software program requiring attestation. It then derives properties from the program based on a set a pre-defined rules. Binary based attestation is usually used to support the integrity of the derivator program.

The idea of semantic remote attestation was proposed by Haldar et. al in [23]. Semantic remote attestation leverages techniques used by language-based security and virtual machines to perform remote attestation. Language-based security leverages program analysis

and program synthesis to enforce security policies on program code. Program analysis provides mechanisms to determine whether a program's execution will satisfy certain properties. Alternatively, program analysis provides mechanisms to ensure that the execution will satisfy certain security properties usually by rewriting a program to perform the necessary checks.

The remainder of this section focuses on this paper. The authors perform program analysis using a Trusted Virtual Machine (TVM). A virtual machine is a component that executes platform independent code (e.g. Java Virtual Machine) and is not to be confused with virtual machines as in hardware virtualization. The authors use a VM for the analysis purpose because code that run on virtual machines are not native code and have high level information. Also, all the code is run completely under the control of a VM, which makes it highly suitable for monitoring executions. A virtual machine is trusted if all measurements of software up to the virtual machine are validated using the TCG attestation mechanism. The basic idea is that a verifier can make a request to a trusted VM to check if a program code on the attesting platform has certain properties. We call this type of attestation as derivation based attestation because properties are derived on the attesting platform using a derivator program (in this case the TVM).

The authors propose that the verifier provides his security policies to the TVM and request the TVM to enforce these policies on the code at the time of execution. Although it is easy to see why such an enforcement mechanism can be useful, it seems to have certain limitations. Firstly, the verifier may not be willing to disclose his policies to the attesting platform. Secondly, the verifier's policies may not be correctly understood by the TVM, particularly if the communicating parties do not know each other. Thirdly, in order to ensure that policies are being correctly enforced, additional programs or code need to be provided by the verifier which the attesting party must be willing to install. We think the role of the TVM must be negotiable based on what suits the attesting platform and the verifier best. For example, where policy enforcements are not suitable on the attesting platform for reasons outlined above, the TVM can just send a signed report of all the properties satisfied by the code and play the role of a monitor rather than an enforcer. If the TVM acts as a simple monitor, it is also not clear exactly what types of properties its might be able to attest to. For example, is this limited to type enforcement and reading inputs and outputs or attest more complex properties that may require some form of evaluation runs. One other limitation of this approach is that it is only suitable for attestation in systems where program code is available. This could restrict its usage in applications like web services where the program code is hidden and only the program functions are exposed.

3.5.4 Enforcement Based Attestation

In enforcement based attestation, the attesting party is enforced with policies that are either chosen by itself or by the verifier. Then, by using binary attestation, the attesting party is able to prove to the verifier that its platform adheres to a set of rules as described in the policies. The verifier believes that the attester has the properties associated with the policy rules because there is enough proof of enforcement.

PRIMA

PRIMA [26] proposed by Jaeger et. al is an integrity measurement architecture and is an extension of IMA. IMA is founded on binary attestation and measures all programs/code in a platform at load time. Jaeger et. al argue that IMA/TCG attestation mechanism measures

many programs that are often not needed for a verifier to know. PRIMA only measures the programs that are of interest to a verifier. In order to ensure that programs are integrity protected during run time and not just at load time, PRIMA controls the flow of information in the measured programs at run time using Mandatory Access Control (MAC) policies. It uses a weaker version of Clark-Wilson (CW) integrity model [19] known as CW-Lite [38]. In this model, information is said to flow from subject 1 to subject 2 through an interface I in the program running as subject 1. CW-Lite enforces that (1) all high integrity objects meet integrity requirements initially (2) all trusted subjects meet high integrity requirements initially (3) all information flows from equal or higher integrity subjects are permitted (4) information flow from low integrity subjects are permitted only at filtering interfaces. PRIMA assumes that CW-Lite is enforceable on a system i.e all subjects and objects with in the system are associated with integrity levels high or low and filtering interfaces are defined for filtering subjects which allow sanitization of low inputs to high.

PRIMA performs the following measurements on the attesting platform at load time leveraging the TCG integrity measurement architecture. (1) Trusted subjects are a set of all MAC policy subjects that must be trusted by the verifier for the integrity of the system to be verified. PRIMA collects the list of all trusted subjects in the system and measures the list. This also includes the filtering interfaces for those subjects. (2) All program and data used by the trusted subjects are measured. (3) The binary MAC policy is measured for correctness.

After load time and during run time, the MAC policy governs the flow of information from one trusted subject to the other. At the time of remote attestation, the platform reports all the measured values to the verifier. The verifier verifies all the measurements to determine if the integrity of the subjects, objects and the policy has been protected. It then determines if the trusted subjects and filtered interfaces are acceptable by the verifier. If the verifier does not trust one of the subjects, then the remote party must assume that subject as receiving low integrity information flow. For each subject in the measurement list, the verifier verifies if all information flow connected to it are directly from other trusted subjects. If this is true, it considers the state of the system to be trustworthy with respect to those trusted subjects.

Jaeger et. al. propose a model where attestation is founded on enforcing MAC policies on the attesting party. If the integrity of the policy, its subjects and objects are protected, then the system can be said to provide a 'MAC policy enforcement' property at a high level. However, the elements of the MAC policy will determine the properties of the system at lower levels of granularity. PRIMA is an instantiation of the enforcement based attestation model where a simple policy like CW-Lite is attested. It is unclear if the verifier's requirements may be included in the enforced policies. For example, for a given and enforced MAC policy, can the verifier determine which subjects are considered high and which are considered low in the attesting platform.

Behaviour based attestation - 1

Li et al. [30] describe a model for attesting the behaviour rather than the binaries of platforms. The main objective of the model is to determine if a system will behave in a way that is compliant with a verifier's expectation. A system behaviour is defined as a quad-tuple a where a A = S P I O where S is a set of all subjects, P is a set if all executable programs, I is a set of all behaviour inputs and O is a set of behaviour outputs. A behaviour is said to be trustworthy if changes to a system's security policy caused by that behaviour is compliant with a verifier's policy. A system behaviour also transforms the state of a system. The new state of the system is considered trustworthy if the platform was in a trustworthy state and

the system behaviour that caused the transformation is also trustworthy.

The attestation mechanism in this model is based on capturing actions or behaviours inside a system and the state changes that happen as a result of such actions. The authors only capture behaviours that are of interest (which is defined in a verifier's expectation policy). This however, does not reduce the complexity of the model because it is still not clear how every behaviour in the system can be practically captured and recorded for analysis.

Behaviour based attestation - 2

Alam et. al propose a model based behaviour attestation using trusted platforms in [2]. In this model, the trustworthiness of a target platform is based on the behaviour of its policy model. A policy model is used to represent the security policies of a system in an abstract and formal way. The behaviour of a policy model captures how the model responds to the changes to its environment. A system is defined as trusted if the behaviour of the policy model conforms to a particular expectation behaviour.

The proposed model provides an high level framework for performing remote attestation using policy model behaviour. The framework consists of four main steps. (1) A behaviour identification and association process that identifies all the behaviours of the policy model that must be trusted. (2) The behaviour specification process defines the identified behaviours formally and produces a high level behaviour policy on expectation. (3) A transformation step where the behaviour policy is transformed to a low level policy language. (4) Behaviour attestation process where a target platform receives the expected behaviour policy and provides proofs that the expectations have been met. The authors then move on to demonstrate the applicability of this framework using the formal specification of the UCON [33] model.

The main advantage of this approach seems to be that for every policy model, once the behaviour specification has been set up and defined, it can be reused to model the behaviour of many policies with in that model. The complexity of defining the specification is directly dependent on the complexity of policy model. As this is only a one time problem, this overhead may be acceptable. However, the main issue concerning this paper is providing the proof that an expected behaviour has been met. In order to prove this, the authors create a behaviour attribute in the case of the UCON model and associate it with all subjects, objects and actions. The behaviour is either 'trusted' or 'untrusted'. For example, a policy states that an object O is allowed to be accessed by a subject S for a maximum of N times. The expectation policy here states that the access information I has to be updated by an update procedure U each time access has been granted i.e, access count to be decremented by 1. The authors argue that an update on I by U is trusted only if I and U are themselves trusted at the time of the update. It is not clear how the attesting party can prove to the verifier if these components are trusted or not. This might be possible using binary or some other type of property attestation. Secondly, for every policy associated in the system, there is a behaviour policy associated with it. While generation of these policies may not add overhead, the verification of both the main policy and its behaviour does not seem to be practical. This is especially true if the proof for behaviour policies are all provided in the form of certificates from an attesting platform (as proposed by the authors). Also, it is not clear how to trust the processes that generate the proofs at the time of attestation.

3.6 Health Care Scenario using Credential Based Claims

In this subsection, we give an example about identifying claims and analysing the assurance of these claims in a Health Care Scenario (AAL/HEC). We identify a number of claims which are made by various entities. These claims relate to the capability, state, identity or other information relating to the system as a whole. These claims are outlined below and analysed in order to determine which can be addressed through the use of credential based mechanisms.

3.6.1 Identified Claims

The following claims are used in AAL use case scenario:

Claim 1 Device State: The state of the device (e.g., the tablet of the Field Representative) can always be verified to not have malicious applications running in parallel with the current application.

Claim 2 Malware: The mobile device of the AAL/HEC field representative should have protection mechanisms against malware that may interfere with the customer registration and data acquisition platform.

Claim 3 Contract: A copy of the signed eHealth/AAL service contract has been generated and given to the Customer.

Claim 4 Access Control: Access control policies for the Customer data have been defined and enforced.

3.6.2 Credential Based Assurance of Identified Claims

The following credential based methods can be used for assuring the claims identified above:

Claim 1 and 2 An issued device could be configured in such a way that it will only allow the execution of whitelisted processes. The use of credential technologies such as X.509 in order to sign a given process present on the device and certify that this has been evaluated and identified as being non-malicious. This implementation is based on the principals of application whitelisting.

Claim 3 There are a number of potential approaches that could be implemented in order to achieve this outcome. One such implementation would be the use of a smartcard based signing mechanism to provide credence to a physical signature provided by the customer. In this implementation both the issuer and the customer to sign the transaction would present a smartcard.

Claim 4 Access control policies as defined in the scenario can be based on those present, state of device, location of the device and other criteria. In each of these cases it is possible to address these claims through the use of a combination of anonymous and X.509 based credentials. The usage of these can be employed based on the specific access control policy being enforced. In the case of geolocation for example the presence of cryptographically signed

Galileo or other GNSS signals could be used to attest to location. In absence of this the use of Bluetooth LE beacons providing cryptographically signed location implementation provide potential for use.

4 Challenges to Security, Privacy and Trust on AU2EU Use Case Analysis

In this section, we analyse the four use cases (see also [**?**]: AAL, DNA Data Management, Translational Research, and PACS. We give a short summery first, then identify the claims, which is followed by the main methods that affect the assurance level for the different claims.

4.1 Use Case: eHealth/ambient Assisted Living

4.1.1 Scenario Summary

The goal of the eHealth Scenario Use Case is for the Care Coordinator to provide the Care Giver the necessary and relevant healthcare data about the patient in question so that the Care Giver can react as fast as possible to service the patient's request. The Care Coordinator obtains the request from the Care Service Provider who is notified by the patient. The Care Coordinator verifies the authentication of the Care Giver prior to giving access to the patient's data. Once the patient data and the request have been passed to the Care Giver, the Care Giver takes actions to provide the appropriate service to the patient. The Care Giver may then update the Care Coordinator on the actions achieved along with updated patient data.

4.1.2 Claims in this Scenario

Identify a set of claims related to authentication, authorization and trust in this scenario.

Claim 1 Care Service Provider claims that it has a received a request from one of its patients

Claim 2 Care Service Provider provides a claim to the Care Coordinator which can be used to authenticate the Care Giver

Claim 3 The Health-Care Service Provider provides a claim to the Care Coordinator that the service request is authorised

Claim 4 Care Coordinator provides a claim to the Care Giver validating its authenticity

Claim 5 Care Giver claims to the Care Coordinator that it is a legitimate Care Giver

Claim 6 Care Giver requests access to patient data based on its authentication claim

4.1.3 Assurance of Claims in this Scenario

In this subsection, we discuss the mechanisms that are associated with determining the assurance level of each of these claims.

Claim 1 The care service provider is notified of the request via a monitoring system or the press of an emergency button. In either of these circumstances it is feasible for the originating device to cryptographically sign the request in order to provide increased level of assurance of the claim.

Claim 2 The use of cryptographic signing can be used to authenticate the Care Giver. In such a scenario the public key of the Care Giver (or alternately an equivalent authentication token) could be signed by the Care Service Provider and provided to the Care Coordinator in order to provide increased level of assurance of this authentication.

Claim 3, 4 & 5 The use of cryptographic signatures can be used in order to verify the claim.

Claim 6 As a result of the previous authentication claim a token could be provided which is used to provide proof that the authentication claim is current. This token could be sent alongside the request for patient data.

4.2 Use Case: DNA Data management in Clinical Trials

4.2.1 Scenario Summary

This use case is about biomedical research studies on human subjects. The goal of this use case is to produce information on safety and efficacy of an administrated intervention. There are a number of parties involved in a clinical trial, such as a sponsor, subjects, investigators, and researchers etc. The sponsor designs the trial in coordination with a panel of expert clinical investigators. The subjects are volunteer patients in a trial. During the trial, investigators recruit patients with the predetermined characteristics, administer the treatment(s) and collect data on the patients' health for a pre-defined time period.

During recruitment procedure, only authorized parties (sponsors, investigators) can enquire the DNA archiving system, and only the information whether a specified biomarker or genetic variation is present or not can be learnt. During analysis of the results of a clinical trial, only qualified representatives of a sponsor (researchers) can analyze the DNA information, and the researches can obtain the necessary answers from the data but are unable to observe the actual DNA information.

4.2.2 Claims in this Scenario

Identify a set of claims related to authentication, authorization and trust in this scenario.

Claim 1 Sponsor or investigator claims to the DNA management system that they need to make some queries to the DNA database for a clinical trial with a specified purpose

Claim 2 Sponsor/investigator claims to the DNA management system that it is a legitimate sponsor/ investigator

Claim 3 DNA management system provides a claim to the sponsor/investigator that they can use to perform the required queries

Claim 4 Sponsor/investigator query the system based on its authorization claim

Claim 5 Patient claims to consent for his medical information use in the clinical trial

Claim 6 Sequencer authenticates himself to the DNA management system

Claim 7 Sequencer uploads the encoded DNA sequence into the DNA management system based on its authorization claim

Claim 8 Researcher authenticates himself to the DNA management system

Claim 9 Researcher make queries to the DNA management system about the encoded DNA sequence of the volunteer patient based on its authorization claim

4.2.3 Assurance of Claims in this Scenario

In this subsection, we discuss the mechanisms that are associated with determining the assurance level of each of these claims.

Claim 1, 2, 6, 8 The assurance level is mainly associated with the strength of the authentication mechanism used by the sponsor or investigator or Sequencer, the trust on the issuer, the channel through which the claim has been transferred, and if the statement has been generated using a tamper resistant module.

Claim 3 & 4 The assurance level is mainly associated with the the channel through which the claim has been transferred, and if the statement has been generated using a tamper resistant module.

Claim 5 The assurance level is associated with the strength of the authentication mechanism used by the patient, the channel through which the consent has been transferred, and if the consent and the cryptographic operations on the consent have been generated using a tamper resistant module and secure storage.

Claim 7 & 9 The assurance level mainly depends on how well the claim matches the authorization policy, the transfer channel, and the secure management of policies.

4.3 Use Case: Translational Research

4.3.1 Scenario Summary

This use case is about translational research (Movember). It aims at constructing a sustainable worldwide database for clinical, marker-related, and imaging data for scientific analyses and improvement of clinical guidelines. 14 Research Institutes and University Medical Centers in 5 regions of Australasia, Europe, UK, Canada, USA are involved in the activity.

Movember needs to deal with very sensitive data, and many different organizations and users (biostatisticians, technicians or radiologists) from different regions need to access it in a distributed system. As a result, it is crucial to ensure that proper user authentication and authorization, consent management and information trustworthiness etc are provided.

4.3.2 Claims in this Scenario

Identify a set of claims related to authentication, authorization and trust in this scenario.

Claim 1 The system claims that research institutions are able only to write data into their DMZ partitions.

Claim 2 The system claims that the research institutions are not able to read data from DMZ.

Claim 3 The system claims that only the Quality Checker (person or tool) is able to read data from DMZ.

Claim 4 The system claims that Quality Checker is not able to write data in the DMZ.

Claim 5 The system claims that Access to the "Globally accessible shared translation research data" is limited to those who have the knowledge to make statistical analysis.

Claim 6 Patient claims to consent for his medical information use in the translational research

Claim 7 Organization claims the landing zone to the system

Claim 8 Organization claims to the system that it can upload the anonymized data to their DMZ landing zone

Claim 9 Users authenticate themselves to the system

Claim 10 Users provide claims that they have the access to the relevant data of DMZ

4.3.3 Assurance of Claims in this Scenario

In this subsection, we discuss the mechanisms that are associated with determining the assurance level of each of these claims.

Claim 1-5 The claims are from the local system, so the assurance level are mainly associated with the secure storage, the cryptographic strength, and how the authorization policy is generated and maintained.

Claim 6 the assurance level is associated with the strength of the authentication mechanism used by the patient, the channel through which the consent has been transferred, and if the consent and the cryptographic operations on the consent have been generated using a tamper resistant module and secure storage.

Claim 7 The assurance level is related to the strength of the authentication mechanism, the channel through which the claim has been transferred, and if the landing zone matches the authorization policy etc.

Claim 8 The assurance level depends on the transfer channel, if the DMZ zone and the access 'upload' match the authorization policy, and cryptographic strength.

Claim 9 The assurance level mainly depends on the strength of the authentication mechanism, the trust on the issuer of the credential, and transfer channel.

Claim 10 The assurance level is mainly dependent on how well this claim match the system authorization policy.

4.4 Use Case: PACS

4.4.1 Scenario Summary

This use case is a medical imaging technology that provides storage of images from multiple modalities. The goal of this use case is to provide hard copy replacement, remote access capabilities of off-site viewing and reporting, an electronic image integration platform, and radiology workflow management.

As PACS needs to handle very sensitive data, and many different hospitals and users need to access it in a distributed system, it is crucial to ensure that proper user authentication and authorization mechanism is provided.

4.4.2 Claims in this Scenario

In this section a set of example claims related to authentication, authorization and trust for the PACS use case are presented

Claim 1 Patient claims to consent for his medical information use in PACS

Claim 2 Hospital B radiologist authenticates to the Hospital B PACS Broker

Claim 3 Hospital B radiologist requests patient data which resides in Hospital A PACS database

Claim 4 Hospital A PACS Broker evaluates the policies and attributes of the radiologist, and makes a claim about his/her access rights. It sends the claim to Hospital B PACS Broker as a security token.

Claim 5 Hospital B PACS Broker forwards the security token to Hospital B radiologist

Claim 6 Hospital B radiologist will use the security token to communicate with the Hospital A PACS databases thereafter.

4.4.3 Assurance of Claims in this Scenario

In this section, mechanisms that are associated with determining the assurance level of each of the above claims are listed.

Due to the distributed architecture of PACS use case, the assurance level of any transmitted claim, both during authentication and authorization, is strongly associated with the trust established when creating claims. Therefore, the cryptographic systems used for communication affect the assurance level of all subsequent claims.

Authentication system employed for the PACS collaborative scenario; as discussed in 3.2, there are different types of claims-based authentication systems. For example, anonymous credential system with a tamper resistant hardware will result in higher assurance level.

Database system used to store patient information; the type of database and the level of granularity, types of classifications and storage form.

Access control systems; for example, the granularity supported and the effectiveness in matching claims' attributes to ensure that claims are relevant to decisions

Mechanisms employed as part of the authorization to detect and process the geographical location and the time of transferring claims.

The security token generation, distribution and storage system.

4.5 Use Case: Biosecurity

4.5.1 Scenario Summary

The biosecurity incident response scenario and use case is based on using the CSIRO Collaboration Platform (CCP) and associated networked services infrastructure (henceforth the CCP System) for real-time, group-to-group, multi-site collaboration. The current state of the CCP System is that is does not provide strong authentication capabilities in entrusted unions. The goal of this use case, which is one of the two pilots of the project, is to add strong authentication and authorisation with interoperability mechanisms to the CCP System. The addition of these services will allow a user at a platform to securely authenticate the platform at a particular site to the CCP System's server. Because the scenario involves a group of participants at each site, the authentication and authorisation systems integrated to the CCP System will also allow the participants at each site to authenticate to the CCP server as attendees.

4.5.2 Claims in this Scenario

We present the main claims of this scenario next, focusing on the claims related to assurance of security in the use case.

Claim 1 Participating organisations provide the identities of experts to Animal Health Australia (AHA) through one person representing this organisation. This person is considered trustworthy in providing the information.

Claim 2 Each participating organisation produces an issuing claim for having a credential issued for each participating expert of the organisation.

Claim 3 AHA, when creating a new biosecurity incident, makes issuing claims for credentials for all people on the list of participants for this incident.

Claim 4 When creating a new biosecurity incident, the authorisation policy for this meeting is instantiated. This policy is the basis for the policy claim made by the CCP server side when a user wants to authenticate a platform or themselves.

Claim 5 A user at a platform, in order to fulfill a policy made through a policy provided by the CCP server, provides an authentication claim based on obtained credentials.

4.5.3 Assurance of Claims in this Scenario

We next discuss how sufficient assurance is ensured for the claims in this scenario.

Claim 1 The identity of individuals is retrieved in a way with sufficient security. The claiming is done through any mechanisms providing sufficient assurance for the purpose.

Claim 2 Using the cloud-based credential system for issuing cryptographic credentials signed with the private key of the respective organisation ensures integrity of the attribute data.

Claim 3 Using the cloud-based credential system for issuing cryptographic credentials signed with the private key of AHA ensures integrity of the attribute data.

Claim 4 This policy is the basis for and determines the authentication claim 5.

Claim 5 Using the cloud-based credential system, the user can make a claim based on her credentials. The CCP Server can verify the evidence related to this claim using the public keys of the organisation of the user and AHA. Verification implies integrity of the attribute information and thus assurance of attribute correctness.

4.5.4 Challenges

Relating to the misuse case of misusing the CCP Shared Workspace as discussed in D1.2.1, the claim-based authentication to authenticate collaboration platforms to the CCP Server and to authenticate each user as well helps prevent unauthorised persons to have access to confidential data. Also, each of the biosecurity incidents will be hosted as a separate shared workspace with a dedicated access control policy claim.

Secure document sharing of confidential documents related to an incident is a possible extension of the pilot, using the same infrastructure established for the authentication of workspaces and users to the CCP server. The claims made by users to access documents in this case will be such stating the permissions to access the shared documents for a biosecurity incident without revealing the identities of the users. The shared documents service is not interested in learning the identities of users accessing certain shared documents.

Theft, loan, or passing on an access card with fraudulent intentions to a third party poses a security challenge. E.g., a card could be stolen from a legitimate cardholder, or a legitimate holder could give their card in good intentions to another party with delegation intentions in case they cannot attend a meeting. Another authentication factor, e.g., knowledge or biometry can help resolve this challenge. A non-invasive, thus convenience-preserving approach would be automated comparison of a photo of the person logging on with a stored photo of theirs.

Having different organisations issue the identity credentials for their respective affiliated experts means that we encounter an interoperability challenge. Concretely, the problem is that different organisations can use different vocabularies to refer to concepts, e.g., first name vs. given name. Those different terms would be part of the issuing claims for the credentials issued to parties. Allowing to use the different terms with a verifier requires that the vocabularies of the verifier and of the issuers be aligned.

5 Proposed Solutions

This section will outline and discuss claims based security architecture design. First we identify and categorize claims used, then address how they are assured. Then we present how claims can be expressed and evaluated in different languages. Finally we discuss initial design choices for the claims architecture.

5.1 Identification of Claims in Security Architecture

We will be using the security architecture that has been developed in WP1, WP2, and WP3. In this section, we will identify the different types of claims that arise in this security architecture. For example, authentication claims, credential based claims and claims used for authorization. Note that the below-represented view of claims is very generic in terms of adopting a very generic view on claims to demonstrate the power of the claims-based architecture concept. In other architecture documents, the term claim is typically used in its more constrained meaning of authentication and authorization claim in order to not overuse the term.

Authentication claims

An authentication claim is a claim made by a person or other entity which makes an assertion about the party or other parties. In AU2EU, a prominent kind of authentication claim is the presentation token provided by users to relying parties in an authentication transaction. The presentation token includes cryptographic evidence that allows for assuring the integrity of the assertion. Another kind of authentication claim is a SAML token, e.g., used in AU2EU for authenticating the user to a user agent in a headless authentication setting, that is, an authentication setting without a user interface for a special-purpose scenario. One more kind of authentication claim in the current AU2EU use case discussion is a user identifier authenticated through an RFID token, e.g., a building access control token used for AU2EU pilot purposes. An important class of authentication claim is username/password, which may be employed as one of the authentication methods for authenticating users to user agents and, depending on setup, the issuers.

Authorization claims

An authorization claim is a claim made by the local system of the relying party to its authorization component for the purpose of obtaining an authorization decision. Such claim is derived from a validated authentication claim and is itself usually not cryptographically protected any more as it is claimed by the local system to the local authorization component and mutual trust exists between those components.

Request claims

A request claim has the semantic of being a generic request, with concrete examples being the following: request of a resource; request of a new credential; request of user consent for making an authentication claim.

Policy claims

An authentication policy claim is the (part of the) policy that is derived in a suitable manner from the authentication and authorization policy and then communicated to a requestor by a relying party as response to a request claim. The verifier identity is part of the policy claim as it is required for a secure authentication. The semantics of such policy claim is that in case the requestor fulfils it with a corresponding authentication claim, the relying party may grant the requestor access to the resource. An authorization policy claim is expressed through a policy, e.g., in XACML, that is specified by a party with the intention of having it enforced by the local authorization engine.

Consent claims

A consent claim is user consent given in response to a consent request claim. A consent claim can be viewed as a metadata attribute on an authentication claim specifying the user's consent.

Ontology claims

An ontology claim is a claim of a certain equivalence relation between concepts. Claim of an ontology made by the provider of the ontology. Trust in an ontology claim by certain parties, e.g., a relying party, is crucial for security of the system. An ontology may be integrity protected using digital signature schemes or trusted distribution channels, e.g., a secure, that is, encrypted and one-way authenticated channel.

Issuing claims

Claim by the issuer comprising the attributes of a credential to be issued to a user.

Environment claims

An environment claim is a claim about certain environment variables that is generically valid and may be consumed by the authorization system. A prominent example is date/time which may govern authorization decisions, e.g., access may only be given during business hours. Another example is the number of parties logged into a system, which should not be exceeded in a given scenario.

Having identified different categories of claims we now move to how we can be sure these claims are valid, ie. to assurance of claims. In the assurance we especially consider the categories authentication and authorization claims as they are key claims in Au2Eu.

5.2 Assurance of Claims in the Security Architecture

This subsection describes the aspects that need to be addressed when considering the issue of assurance of claims in the security architecture. First it will describe the general characteristics of assurance and then will consider how these assurance aspects are reflected in the generation and verification of authentication and authorization claims.

5.2.1 Assurance Characteristics

In general, the assurance level associated with a claim indicates the confidence a receiving entity can place on the claim that it receives from the other party such as a requesting entity. For example, if a claim is made by an entity A to an entity B, the assurance issue needs to determine how strongly B can believe the claim that has been made by A. There are several factors that affect the level of assurance that can be achieved on a claim. They can depend on a) the trust that a replying party can have in the entity who is making the claim, b) the channel(s) through which the claim is made, c) the content of the claim as well as d) the context in which the claim is made.

For instance, in the case of (a) if two entities are mutually suspicious, the use of trusted third parties vouching for the claim made by an entity helps to increase the level of assurance. (E.g. how much does the recipient trust the entity that is vouching for the claim). In the case of (b), if the security of the channel through which the claim is conveyed say by using cryptographic techniques to protect the claim and the entity vouching for the claim, and the protocols associated with the transfer are secure, then they lead to greater assurance. (E.g. trust on the unforgeability of the signature scheme used by the vouching entity). In the case of (c), it is concerned about whether the actual claim being made is relevant for the decision in question (e.g. how well the attributes in the claim itself match with what the recipient wants to determine), and (d) refers to the context attributes such as location and time associated with the claim which can enhance the assurance level.

In addition, the assurance level also depends on the trust on the processes that have been used to create the claims. For instance, whether hardware trusted computing technology such as TPM that is protected against tampering has been used to create the certified claims helps to increase the level of assurance. Similarly the use of secure software development process such as SDL helps to increase the level of assurance associated with generation and verification of claims.

5.2.2 Assurance of Different Types of Claims

The previous section Sec 5.1 outlined the different types of claims and the entities in the security architecture that are making the claims. These include authentication claims, request claims, policy claims and authorization claims. We will briefly now consider the assurance of authentication and authorization claims based on the assurance characteristics outlined above in Sec 5.2.1

Consider first a simple authentication claim involving users. Assume a user authenticates to a system (user agent) using attributes, username and password. Applying the four aspects outlined above to this case, for the user agent (system) to determine the level of assurance, the aspects that need to be considered include the strength of the authentication mechanism (user-password), the channel through which the attributes have been transferred and whether the attributes used are relevant for the context of the decision making. The assurance level determined in this case could be "low", which could be adequate for everyday less sensitive interactions such as normal email but would be inadequate for accessing online services provided by say government or financial institutions. Higher level of assurance can be achieved for instance if the trust associated with the entity making the claim can be enhanced by say using a government-issued eID token, and having a secure channel for the transfer of the token with appropriate signatures. The assurance level can be further increased if the attributes

such as the tokens and the cryptographic operations such as the signatures on these attributes have been generated using a tamper resistant module and secure storage.

Let us now consider a typical authorization claim. Assume an entity A makes a request to another entity B for a service. In the previous section, the request was treated generally as a claim in itself - a request claim. From the authorization point of view, the assurance associated with this decision involves whether (a) the entity which is making the request is the one who it claims to be (authentication claim) and then (b) whether the entity that is making the request is allowed to obtain the requested service. The authorization claim has as its pre-requisite an authentication claim (a) with its relevant attributes, similar to what we discussed above. With respect to (b), the issue to consider is how security of the authorization decision can be enhanced, which in turn can map to an increased level of assurance. For instance, verifiable attributes of the platform from which the request is made and/or the verifiable context such as the location or the time of the request can help to increase the level of assurance of the claim. The basis for the increase in the assurance level is to do with the relevance of the claim with respect to the authorization decision in question. For instance, only a user with a corporate PC or mobile device could have made that request (claim) and that is directly relevant to accessing corporate documents. Similarly only a user from this location could have made that request (claim) again making it relevant for the authorization decision making. Such an assurance approach of allowing for different levels of authentications and authorization mechanisms and context caters for the needs of the real-world applications.

With different types of claims and assurance methods of claims treated we now move to how to formally express such claims ie the claims languages.

5.3 Claims based Language and Policies

This subsection will describe the claims language and constructs that are used in the specification and representation of different types of claims identified in Sec 5.1. We propose to consider the claims language used in the design of architecture in WP2 and WP3 such as XACML as well as other claims based languages such as the ones based on logic. This section will also describe security policies involving such claims for different use case scenarios.

5.3.1 ABC4Trust-based claims

Multiple classes of claims based on ABC4Trust languages are used in the AU2EU security architecture. This comprises the authentication claims, policy claims and issuing claims. The ABC4Trust Architecture contains the full details of the specification of these claims [4]. We refrain from reiterating those details here and refer the reader to the original source for those, but we rather want to give an overview of the different types of claims here. For a usage description of these claims in the AU2EU context [6].

Authentication claims

Authentication claims (especially presentation tokens in the ABC4Trust and AU2EU context) can be composed of two parts, namely the attributes and values over which the claims are made and a cryptographic evidence for all the claims respectively. The attributes and values are claimed to come from a specific issuer and from a specific credential. Additional data can also be part of the authentication claims, e.g., verifier-given messages (e.g., terms of

conditions), or random nonces. The purpose of authentication claims are the authentication of the entity who wants to get access to a protected resource, service, etc. and in order for this she generates the claims. After successful cryptographic verification, the attributes and values from the claim can be used in the authorization system to make a decision about the success of the access request.

Policy claims

Policy claims (especially presentation policies in the ABC4Trust and AU2EU context) can be derived from or based on authorization policies (i.e., XACML policies) or created from scratch. The policy claims for authentication shall contain a claim for a need-to-know of every attribute and value constraint, which the user has to claim in her authentication claims (i.e., disclosure of attribute or a predicate over the value of it). They also have to contain information about from which credentials by which issuer these attributes have to originate. For the different methods how these policy claims can be created [7].

Issuing claims

Issuing claims (i.e., issuance policies and corresponding attributes in the ABC4Trust and AU2EU context) are composed of claims from the respective issuer. These claims are about the requestor's attributes which will be contained in the issued credential, i.e., their specification and the actual values. These claims have to be transferred in an initial step of the issuance protocol to the user in a secure manner. In advanced scenarios, the issuing claims can also contain additional claims for already existing attributes (which for example will be blindly carried over into the new credential) or claims for creating attributes jointly at random with the user. These advanced features will not be used in the AU2EU context as of current planning.

5.3.2 XACML-based Language and Policies

AU2EU expresses authorization policies defined at the parties using XACML [32]. These can be seen as claims made to the local system in terms of the policies that are in place and should be enforced.

As discussed earlier, the XACML authorization policies can be the basis of the ABC4Trust authentication policies, for details [7]. There are multiple solutions for achieving a proper integration between the two systems, as described in the aforementioned document. In order to achieve a practical integration, the modification of XACML itself is not necessary, but it may be advisable to make some small augmentations to the system handling the XACML requests.

The XACML authorization policies are a different kind of policy claims, and they can be composed of different types of other claims (e.g., environment claims).

5.3.3 Additional Logic-based Language for Claims

Logic based language can be used to express different types of claims. Logic based formalisms enable separation of policies from implementation mechanisms, and have well-defined semantics. Logic based authorization models have been studied by many researchers for the purpose

of formalizing authorization specifications and evaluations.

In this subsection, we present a logic approach based on extended logic programs (ELP)[21]. The extended logic programs (ELP), is based on nonmonotonic reasoning semantics, and has strong expressive power in the sense that it can deal directly with incomplete information in representation and reasoning. As incomplete information is common when it comes to access control policies, such policies are easier to specify in ELPs. For example, if one wants to express that a user is denied to access patient data if there is no knowledge that he/she is a doctor of the patient, the negation as failure (weak negation) is often the most direct way to express this intention. We will extend ELP with features needed for handling various claims.

Several design considerations

We first present the basic ideas in the development of a formal logic based framework for representing and evaluating claims. We discuss several design aspects such as attribute issuing and delegation correctness, authorization grant and delegation correctness, degrees of assurance/trust, and conflict resolution.

Administrative privilege delegation verification

In our formalization, administrative privilege delegation includes attribute administrative privilege delegation and access right administrative privilege delegation. It is about who can issue the attributes or who can grant the access rights. It is also about how this privilege can be delegated. We assume that System Security Officer (SSO) in a system is the first role that can delegate. Others have to be granted the privilege to delegate.

Attribute issuing and access right grant verification

For a particular attribute a, only the subject with some attributes that have been granted the administrative privilege on a can issue a to some subject. The subject can be individuals, agents or processes. We assume SSO can issue any attribute in the system.

The access right administrative privilege is in terms of a specific access right on a specific object with some attributes. Thus it is possible to say that a subject can only grant read, but not write, on an object to others. For instance, an admin may be able to grant "read" about a patient's medical file but not write. We also assume SSO can grant any privileges on any object in the system.

Degrees of trust and effective trust

There is a need for people to express the degree to which they believe that someone can have some attribute, can perform some action, or someone can grant some attribute/action. In our formalization, we allow a trust degree to be associated with each attribute issuing, access right grant, and each attribute/access right delegation.

As introduced in [35], a statement's effective trust degree should consider all the trust degrees on a delegation path to it, and all the delegation paths to it. In our framework, for any statement, if there exists only one path to it, then the product of all the trust degrees on the path is defined as its *effective trust degree*. If there exist multiple paths to it, then the greatest value of all the paths is defined as its *effective trust degree*. Effective trust degree can

help to dynamically control the administrative privilege delegation and role or access right assignment. For example, if the effective trust degree falls below a certain threshold, then, the system can reject the role assignment or delegation.

Conflict Resolution

A subject may receive conflicting authorizations from multiple administrators, or a user may receive conflicting authorizations due to having different attributes. For example, a radiologist working in both Hospital A and B may receiving conflicting authorizations about reading patient's record in Hospital A which says radiologist in Hospital A can read patient's reord in Hospital A, while radiologist in Hospital B cannot. Similar to [35], we use the effective trust degree to resolve the conflicts. This means that the authorization with higher effective trust degree will override. For example, by giving a higher trust degree to the positive authorization granted to Hospital A radiologist, the subject can read patient's record Hospital A. This method would also allow the administrative privilege delegator to control their delegations flexibly. For instance, by giving a delegatee a less than one trust degree, a delegator can keep a higher priority than the delegatee in their 'can grant' delegations. Therefore, when the delegator gives an authorization a full trust degree, this authorization will not be overridden by the delegatee's authorizations. This method will help to enforce the high level policies despite the delegations.

When the two conflicting authorizations have the same effective trust, the conflicts can be unresolved by using the simple negative-take-precedence solution or the other way around.

ABAP

We will call our logic framework as Attribute based authorization program (ABAP). It is an extended logic program which contains system built-in predicates and user defined predicates. Users can use ABAP to express complex security policies based on delegation and trust. Any ABAP consists of application rules and system rules. Application rules are defined by users to express the desired application related security policies, using user defined predicates as well as built-in predicates. System rules are defined to capture the features stated above such as attribute and access right administrative privilege delegation correctness, attribute issuing and access right grant correctness, and conflict resolution. The domain independent rules are the same for all applications. These general rules are combined with the domain-specific rules defined by users to derive the authorization decisions holding in the system. We will adopt well known answer set semantics for ABAP because it is more suitable for our purpose of handling authorization conflicts since it provides a more flexible way to deal with incomplete and contradictory information.

Syntax of ABAP

In attribute based authorization program (ABAP), variables are denoted by strings starting with lower case letters, and constants by strings starting with upper case letters. There are two authorization types denoted by $-, +$, where $-$ means *negative*, $+$ means *positive*. A negative authorization specifies that the access must be forbidden, while a positive authorization specifies that the access must be granted. A *rule* r is of the form:

$$b_0 \leftarrow b_1, ..., b_k, not\, b_{k+1}, ..., not\, b_m$$

where $b_0, b_1, ..., b_m$ are literals, $m >= k >= 0$, and *not* is the negation as failure symbol. An *Attribute Based Authorization Program*, ABAP, consists of a finite set of rules.

The predicate set P in ABAP consists of a set of ordinary predicates defined by users, and a set of system built-in predicate designed for users to express various claims about authentication, authorization, consent, policy, and environment etc.

Attribute administrative privilege delegation claim

For attribute assignment delegation, a special predicate $(g, gA, w, d)canIssue(s, sA, s', sA')$ is defined, where (g, gA, w, d) is called the issuer of the delegation. Intuitively, it means that a user g with attributes gA claims that the subject s with attributes sA can not only assign attribute sA' about s', but also further delegate this administrative privilege on sA' for the maximum delegation depth d, and g's trust degree on this delegation to s is w. If the depth is 0, s cannot further delegate. A depth of 1 would mean that s can further delegate to some role with maximum depth of 0 and etc. Here attributes is a set of pairs (key,value).

For example, in the PACS use case, the following predicate states that SSO says/claims Hospital B System Administrator can issue attribute radiologist in hospital B. The trust or assurance degree of this claim is 1, and the delegation depth is also 1, which means the system administrator can further delegate this administrative right once.

$$(g, \{(Role, SSO)\}, 1, 1)canIssue(g, \{(Role, SystemAdmin), (Hospital, B)\}, s, \{(Role, radiologist), (Hospital, B)\})$$

Access right administrative privilege delegation claim

Similarly, for privilege grant delegation, we have a special predicate $(g, gA, w, d)canGrant(s, sA, a, o, oA)$, which means that a user g with attributes gA claims that subject s with attribute sA can not only grant access right a on o with attribute oA, but also further delegate this administrative privilege to the maximum depth of d. And g's trust degree on this delegation to r' is w.

For example, in the PACS use case, the following predicate states that SSO says/claims Hospital A PACS Broker can grant access Read to patient data in Hospital A. The trust or assurance degree of this claim is 1, and the delegation depth is also 1, which means the PACS broker can further delegate this administrative right once.

$$(g, \{(Role, SSO)\}, 1, 1)canGrant(s, \{(Role, PACSBroker), (Hospital, A)\}, read, o, \{(Owner = patient), (Hosipital = A)\})$$

Attribute issuing claim

In terms of attribute assignment, a special predicate of the form $(g, gA, w)issue(s, sA)$ is defined. Intuitively, it means that a issuer g in attributes gA assigns attributes sA to subject s. g's assurance degree on this attribute assignment is w. The subjects can be individuals, agents or processes.

In the PACS use case, the following example states that Hospital B's system administrator

claims that Bob is a radiologist in Hospital B. The assurance degree about his claim is 1.

$$(g, (Role, SystemAdmin), (Hospital, B), 1) issue(Bob, (Role, radiologist), (Hospital, B))$$

Authorization claim / Policy claim

Similarly, we define a special predicate grant $(g, gA, w) grant(s, sA, o, oA, t, a)$ for authorization. It means that g in attribute gA says that subject s with attributes sA can/cannot (depending on the type t) do access/service a on object o with attribute oA, and the grant weight is w.

The predicate can also be used to express the policy claim sent to the requestor so that the requestor can use it in a multi domain environment.

In the PACS use case, for example, the following statement states that Hospital A PACS Broker says/claims radiologist from Hospital B can read Hospital A's patient's record. The trust or assurance degree of this claim is 0.9.

$$(g, (Role, PACSBroker), (Hospital, A), 0.9) grant(s, (Role, radiologist), (Hospital, B), o, \\ (Owner = patient), (Hosipital = A), +, read)$$

Authentication claim

We use the predicate $authen(s, cred)$ to express the authentication claim. It says that subject s is authenticated using the credential $cred$.

Request claim

$request(s, sA, o, oA, a)$ can be used to express the request that subject s with attributes sA wants to do access/service a on resource o with attributes oA. If the service does not need any resource access, the o and oA can be empty.

Consent claim

Consent claim can be represented by the *grant* predicate. A patient Bob consent the researcher to read his medical record can be represented as:

$$(g, (Role, Patient), (Name, Bob), 1) grant(s, (Role, Researcher), o, \\ (Owner = Bob), (class = medicalrecord), +, read)$$

Environment claim

Predicate $environ(eA)$ indicates that the environment variable setting is eA, where eA is a set of pairs (e, v) representing the environment variable e's value is set to v.

$$environ(\{(Location, A), (Date, B), (Time, C), (Platform, D)...\})$$

In the PACS use case, for example, the following statement states that Hospital A PACS Broker says/claims radiologist from Hospital B can read Hospital A's patient's record if they are in Hospital A. The trust or assurance degree of this claim is 1.

$(g,(Role,PACSBroker),(Hospital,A),1)grant(s,(Role,radiologist),(Hospital,B),o,$
$(Owner=patient),(Hosipital=A),(Loaction=HosipitalA),+,read)$

Assurance level claim

We use predicate $aol(o,oA,a,w)$ to express the sensitive level w the access a on object o with attribute oA has. If this level is high, then the assurance level of the required authentication and authorization must also be high.

System rules

The domain-independent rules are designed to formally achieve the design aspects described before on attribute and access right administrative privilege delegation, delegation correctness and conflict resolution. We give some examples here.

The next two rules are about delegation of attribute assignment. The first rule means that any delegation from SSO will be accepted, and is represented by predicate $canIssue1$. The second rule means that if an attribute has been delegated to assign roles for the maximum delegation depth d', then the attribute's further delegation with depth less than d' is accepted.

(r_1). $(g,\{(role,SSO)\},w,d)canIssue1(s,sA,s',sA') \leftarrow$
$(g,\{(role,SSO)\},w,d)canIssue(s,sA,s',sA')$
$hasAttribute(g,\{(role,SSO)\})$

(r_2). $((g,gA,w,d)canIssue1(s,sA,s,sA') \leftarrow$
$(g,gA,w,d)canIssue(s,sA,s',sA'),$
$(g',gA',w',d')canIssue1(g,gA,s',sA'),$
$hasAttribute(g,gA),d'>d$

The next two rules mean any attribute issuing from the System Security Officer or a grantor holding the right to issue an attribute is accepted, and is represented by predicate $assign1$.

(r_3). $(g,\{(role,SSO)\},w)issue1(s,sA) \leftarrow$
$(g,\{(role,SSO)\},w)issue(s,sA),$
$hasAttribute(g,\{(role,SSO)\})$

(r_4). $(g,gA,w)issue1(s,sA) \leftarrow$
$(g,gA,w)issue(s,sA),$
$(g',gA',w',d')canIssue1(g,gA,s,sA),$
$hasAttribute(g',gA')$

The next rule says a subject s has attribute sA if there exists a issuing claim saying s has attributes sA, with the issuing trust degree higher than some threshold W, and s has been authenticated.

$hasAttribute(s,sA) \leftarrow$
$authen(s,cred),(g,gA,w)issue1(s,sA),w>=W$

Pre-processing of the program

The special predicates defined above are not in the normal format of the predicates, but are more user friendly. We will conduct pre-process which will transform them into the normal format of predicates or rules based on their semantics.

Access Control Policy

A request is a four-ary tuple (s, sA, o, oA, a), which denotes that a subject s with claimed attributes sA requests access a over object o with attribute oA. The access control policy is a function f from (s, sA, o, oA, a) in $\mathcal{S} \times \mathcal{SA} \times \mathcal{O} \times \times \mathcal{OA} \times \mathcal{A}$ to $\{true, false, undecided\}$. Given a request (s, sA, o, oA, a), if $f(s, sA, o, oA, a) = true$ then it is granted. If $f(s, sA, o, oA, a) = false$ then it is denied. Otherwise, $f(s, sA, o, oA, a) = undecided$, and it is left to be decided by the implemented access control system. There is a distinction between $false$ and $undecided$. $false$ means denial in the stronger sense in that its negation exists, while $undecided$ means denial in the weaker sense in that it does not succeed.

According to answer set semantics, there may exist several authorization answer sets for a given ABAP, and they may be inconsistent with each other in the sense that they may contain conflicting literals. To maximize the accessibility, we adopt optimistic approach to deal with this problem. That is $f(s, sA, o, oA, a) = true$ if a subject s with attribute sA is granted to do the service/action a on o with attribute oA in one of the authorization answer set.

Having described ABAP which can be used to express and evaluate claims we are now ready to describe the initial security architecture.

5.4 Preliminary Outline of Design Choices for Assurance based Security Architecture

In this section, we consider the design choices for integrating assurance techniques into the security architecture for a distributed system. First we will outline a brief architecture for a distributed system such as the one that is being developed in our AU2EU project. Then we will outline the design of assurance based security architecture.

We begin with a brief outline of security architecture that is relevant for our discussion in this section. Essentially a user wishes to request a service from a service provider. The requester does not have upfront information as to what credentials and attributes are needed to satisfy the service provider's policy regarding the access to the service. The service provider communicates to the requester the attributes required for accessing that service. Depending on the attributes provided by the requester, the service is either granted or denied. Hence the approach is one of a progressive approach of attribute-based authorisation which is suitable in an open cross-domain distributed environment. Furthermore such an architectural approach has the potential to achieve privacy enhanced authentication and authorisation, as the requester only provides the attributes needed for accessing a specific service and the service provider only provides the policies associated with that particular service.

Now let us consider the design choices for the integration of assurance techniques into the security architecture. First consider the communication of the policy associated with authentication that the service provider communicates to the requester. In this case, the integration of assurance level into the architecture requires the service provider to have a specification of different assurance levels required for access to different services from the point of view of attributes and credentials needed for authentication. Then there must be mechanisms in the communication protocol to indicate to the requester not only the different attributes required to satisfy the policy but also indicate the level of assurance this would entail. Then in the interaction from the requester to the service provider, the requester

should have the flexibility to specify the different sets of attributes for different assurance levels. Then comes the authorisation decision which could be much more fine granular in terms of the rules and state of parameters used in the authorization policies to derive the decision. This will be dependent on the specific type of access request in terms of the specific operations on objects and resources involved in the service provision. The finer granularity of authorization decisions can require a greater level of assurance, as we saw in the example scenario above in terms of the context as well as the confidence associated with the attributes provided (in addition to the different types of attributes). Hence there would be a need to include information associated with the different assurance levels in the policy specifications associated with the authorization service.

In terms of the infrastructure, let us now consider a simple scenario involving cloud based services. In such a scenario, the authorisation components could be implemented as cloud services, e.g. policy decision points running as cloud services, with REST-based interfaces, which facilitate elastic scalability depending on system load and easy deployment. Similarly for the authentication platform, the cloud services can be run by the party using them or trusted third parties offering authentication services to relying parties. In terms of assurance, with respect to authentication attributes, the trusted third party could be trusted to provide certifications and verifications that can be used to determine the assurance level. The authentication platform could also be augmented with trusted computing technologies. With respect to authorization attributes and properties, in a general case, one would also need other trusted parties for certifying specific properties of the relying parties. Since these properties are dependent on the context and application, there could be a need for more than one trusted property certifiers. Such a cloud based approach to the design of assurance based security architecture can also help to enhance fine granular attribute based privacy.

In this document, we first presented the state of the art about various techniques for claim specification and verification. We then conducted AU2EU use case analysis in which we identified claims and assurance of claims related to authentication and authorization in each scenario. After this, we discussed possible solutions for identifying different types of claims from security architecture, techniques for generation and verification of claims and the levels of assurance that can be achieved. We described the claims language and constructs that can be used in the specification and representation of different types of claims, such as ABC4Trust-based claims, XACML-based policy claims and logic-based claims. Finally we present a preliminary outline of aspects that need to be considered when integrating claim assurance techniques into the security architecture. We plan to expand this in the development and implementation of the architecture in the subsequent stages.

References

[1] A. Abdul-Rahman and S. Hailes. Supporting Trust in Virtual Communities. In Proc. of the 33rd IEEE Hawaii International Conference on System Sciences, volume 6, page 6007, Washington DC, USA, 2000.

[2] M. Alam, X. Zhang, M. Nauman, T. Ali, and J.-P. Seifert, "Model-based behavioral attestation," in SACMAT '08: Proceedings of the 13th ACM symposium on Access control " models and technologies, pp. 175-184, ACM, 2008.

[3] Abadi, M., Burrows, M., Lampson, B., Plotkin, G.: A calculus for access control in distributed systems. ACM Trans. Program. Lang. Syst. 15(4), 706-734 (1993).

[4] ABC4Trust Consortium: H2.2 - ABC4Trust Architecture for Developers. Available at *https : //abc4trust.eu/download/ABC4Trust − H2.2_ABC4Trust_Architecture_for_Developers.pdf*, October 2013.

[5] AU2EU Consortium, WP1, D2.1.1 – Detailed descriptions of use cases, Work Package 1, 2014.

[6] AU2EU Consortium, WP2, D2.1.1 – Requirements and architecture for cloud-based authentication services, Work Package 2, including deliverables D2.1.1 and D2.2.1, 2014.

[7] AU2EU Consortium, D3.2.1 – Policy translation, Work Package 3, including deliverables D3.1.1, D3.2.1, D3.3.1, D3.3.2 and D3.4.1, 2014.

[8] Blaze, M., Feigenbaum, J. and Lacy, J. "Decentralized Trust Management". In Proceedings of IEEE Symposium on Security and Privacy, pages 164–173, Oakland, CA, May 1996.

[9] Blaze, M., Feigenbaum, J. and Keromytis, A. "KeyNote: Trust Management for Public-Key Infrastructures". In Proceedings of the 6th International Workshop on Security Protocols, volume 1550 of Lecture Notes in Computer Science, pages 59–63, Cambridge, UK, April 1998. Springer-Verlag.

[10] Stefan Brands, Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy. MIT Press, Cambridge USA 2000.

[11] J. Camenisch, S. Moedersheim, G. Neven, F. Preiss, and D. Sommer. A Card Requirements Language Enabling Privacy-Preserving Access Control. In Proceedings of the 15th ACM Symposium on Access Control Models and Technologies, 2010.

[12] J. Camenisch and A. Lysyanskaya, An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation, Advances in Cryptology - EUROCRYPT, Proc. of International Conference on the Theory and Application of Cryptographic Techniques, pp 93-118, Austria, May 6-10, 2001.

[13] J. Camenisch and A. Lysyanskaya, A Signature Scheme with Efficient Protocols. Proc. of 3th Conference on Security and Cryptography for Networks (SCN), LNCS 2576 pp 268-289, Springer 2003.

[14] J. Camenisch, M. Dubovitskaya, A. Lehmann, G. Neven, C. Paquin, and F. Preiss, Concepts and Languages for Privacy-Preserving Attribute-Based Authentication. IDMAN, volume 396 of IFIP Advances in Information and Communication Technology, page 34-52. Springer, (2013).

[15] M. Carbone, M. Nielsen, V. Sassone, A formal model for trust in dynamic networks . Proceedings of Int. Conf. on Software Engineering and Formal Methods, SEFM 2003. IEEE Computer Society.

[16] Patricia Arias Cabarcos, Florina Almenarez, Felix Gomez Marmol, Andres Marin, "To federate or not to federate: A reputation-based mechanism to dynamize cooperation in identity management infrastructures", Wireless Personal Communications, Special Issue on Advances in Trust, Security and Privacy for Wireless Communication Networks, vol. 75, no. 3, pp. 1769-1786, 2014.

[17] Consortium, OpenID Authentication 2.0. Dec 2007, $http://openid.net/specs/openid-authentication-2\_0.html$.

[18] D. Cooper and S. Santesson and S. Farrell and S. Boeyen and R. Housley and W. Polk, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. May 2008, $http://www.ietf.org/rfc/rfc5280.txt$.

[19] D. D. Clark and D. R. Wilson, "A comparison of commercial and military computer security policies," Security and Privacy, IEEE Symposium on, vol. 0, p. 184, 1987.

[20] Felix Gomez Marmol, Joao Girao, Gregorio Martinez Perez, "TRIMS, a privacy-aware trust and reputation model for identity management systems", Computer Networks, Special Issue on Managing Emerging Computing Environments, vol. 54, no. 16, pp. 2899-2912, 2010.

[21] M.Gelfond and V.Lifschitz, Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:pp365-385, 1991.

[22] Gurevich, Y., Neeman, I.: DKAL: distributed knowledge authorization language. In: Proceedings of the 21st IEEE Computer Security Foundations Symposium, pp. 149-162. IEEE Comput. Soc., Los Alamitos (2008).

[23] V. Haldar, D. Chandra, and M. Franz, "Semantic remote attestation: A virtual machine directed approach to trusted computing," in VM'04: Proceedings of the 3rd conference on Virtual Machine Research And Technology Symposium, (Berkeley, CA, USA), pp. 3-3, USENIX Association, 2004.

[24] D. Quercia, M. Lad, S. Hailes, L. Capra, and S. Bhatti , STRUDEL: supporting trust in the dynamic establishment of peering coalitions . Proceeding SAC '06 Proceedings of the 2006 ACM symposium on Applied computing Pages 1870-1874 ACM New York, USA (2006).

[25] Jajodia, S., Samarati, P., Subrahmanian, V.S.: A logical language for expressing authorizations. In: Proceedings of the 1997 IEEE Symposium on Security and Privacy, pp. 31-42. IEEE Comput. Soc., Los Alamitos (1997).

[26] T. Jaeger, R. Sailer, and U. Shankar, "PRIMA: Policy-reduced integrity measurement architecture," in SACMAT '06: Proceedings of the eleventh ACM symposium on Access control models and technologies, (New York, NY, USA), pp. 19-28, ACM, 2006.

[27] A. Nagarajan, V. Varadharajan, M. Hitchens, and S. Arora, "On the applicability of trusted computing in distributed authorization using web services," in 22nd Annual IFIP WG 11.3 Working Conference on Data and Applications Security, pp. 222-237, 2008.

[28] Li, N., Mitchell, J.C.: Datalog with constraints: a foundation for trust management languages. In: Proceedings of the 5th International Symposium on Practical Aspects of Declarative Languages, pp. 58-73. Springer, Berlin (2003).

[29] Li, N., Grosof, B.N., Feigenbaum, J.: Delegation logic: a logic-based approach to distributed authorization. ACM Trans. Inf. Syst. Secur. 6(1), 128-171 (2003).

[30] X.-Y. Li, C.-X. Shen, and X.-D. Zuo, "An efficient attestation for trustworthiness of computing platform," in IIH-MSP '06: Proceedings of the 2006 International Conference on Intelligent Information Hiding and Multimedia, (Washington, DC, USA), pp. 625-630, IEEE Computer Society, 2006.

[31] Microsoft: Microsoft Identity Model, Claims. $http://msdn.microsoft.com/enus/library/microsoft.identitymodel.claims.aspx$.

[32] OASIS, eXtensible Access Control Markup Language (XACML) Version 3.0, $http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en$, January 2013.

[33] J. Park and R. Sandhu, "Towards usage control models: Beyond traditional access control," in SACMAT '02: Proceedings of the seventh ACM symposium on Access control models and technologies, pp. 57-64, ACM, 2002.

[34] J. Poritz, M. Schunter, E. V. Herreweghen, and M. Waidner, "Property attestation-scalable and privacy-friendly security assessment of peer computers," tech. rep., IBM Research, May 2004.

[35] C. Ruan and V.Varadharajan, Dynamic delegation framework for role based access control in distributed data management systems. Distributed and Parallel Databases 32(2): 245-269 (2014).

[36] A.-R. Sadeghi and C. Stuble, "Property-based attestation for computing platforms: Caring about properties, not mechanisms," in NSPW '04: Proceedings of the 2004 Workshop on New Security Paradigms, (New York, NY, USA), pp. 67-77, ACM, 2004.

[37] R. Sailer, X. Zhang, T. Jaeger, and L. van Doorn, "Design and implementation of a TCG-based integrity measurement architecture," in SSYM'04: Proceedings of the 13th conference on USENIX Security Symposium, (Berkeley, CA, USA), pp. 16-16, USENIX Association, 2004.

[38] U. Shankar, "Toward automated information-flow integrity verification for security-critical applications," in Proceedings of the 2006 ISOC Networked and Distributed Systems Security Symposium (NDSS'06), 2006.

[39] Trusted Computing Group, TPM Main - Part 1 Design Principles, Version 1.2, Revision 103, July 2007, *http : //www.trustedcomputinggroup.org*

[40] Security Research Team at IBM Research-Zurich, Specification of the Identity Mixer Cryptographic Library, *http : //domino.research.ibm.com/library/cyberdig.nsf /papers/EEB54FF3B91C1D648525759B004FBBB1/File/rz*3730*revised.pdf*, April 29, 2010.

[41] Gines Dolera Tormo, Felix Gomez Marmol, Gregorio Martinez Perez, "Towards the integration of reputation management in OpenID", Computer Standards & Interfaces, Special Issue on Secure Mobility in Future Communication Systems under Standardization, vol. 36, no. 3, pp. 438-453, March 2014

[42] Woo, T., Lam, S.: Authorization in distributed systems: a formal approach. In: Proceedings of IEEE on Research in Security and Privacy, pp. 33-50 (1992).

[43] S. Yoshihama, T. Ebringer, M. Nakamura, S. Munetoh, and H. Maruyama, "WS-Attestation: Efficient and fine-grained remote attestation on web services," tech. rep., IBM Research, February 2005.