Dynamic Trust Negotiation for Decentralised e-Health Collaborations

A Dissertation
Submitted to the Department of Computing Science
at the University of Glasgow
in fulfilment of the requirements
for the degree of
Doctor of Philosophy

By
Oluwafemi Oluwafeyisayo Ajayi
June 2009

Abstract

In the Internet-age, the geographical boundaries that have previously impinged upon inter-organisational collaborations have become decreasingly important. Of more importance for such collaborations is the notion and subsequent nature of security and trust - this is especially so in open collaborative environments like the Grid where resources can be both made available, subsequently accessed and used by remote users from a multitude of institutions with a variety of different privileges spanning across the collaboration. In this context, the ability to dynamically negotiate and subsequently enforce security policies driven by various levels of inter-organisational trust is essential.

Numerous access control solutions exist today to address aspects of inter-organisational security. These include the use of centralised access control lists where all collaborating partners negotiate and agree on privileges required to access shared resources. Other solutions involve delegating aspects of access right management to trusted remote individuals in assigning privileges to their (remote) users. These solutions typically entail negotiations and delegations which are constrained by organisations, people and the static rules they impose. Such constraints often result in a lack of flexibility in what has been agreed; difficulties in reaching agreement, or once established, in subsequently maintaining these agreements. Furthermore, these solutions often reduce the autonomous capacity of collaborating organisations because of the need to satisfy collaborating partners demands. This can result in increased security risks or reducing the granularity of security policies.

Underpinning this is the issue of trust. Specifically trust realisation between organisations, between individuals, and/or between entities or systems that are present in multi-domain authorities. Trust negotiation is one approach that allows and supports trust realisation. The thesis introduces a novel model called dynamic trust negotiation (DTN) that supports n-tier negotiation hops for trust realisation in multi-domain collaborative environments with specific focus on e-Health environments. DTN describes how trust pathways can be discovered and subsequently how remote security credentials can be mapped to local security credentials through trust contracts, thereby bridging the gap that makes decentralised security policies

difficult to define and enforce. Furthermore, DTN shows how n-tier negotiation hops can limit the disclosure of access control policies and how semantic issues that exist with security attributes in decentralised environments can be reduced. The thesis presents the results from the application of DTN to various clinical trials and the implementation of DTN to Virtual Organisation for Trials of Epidemiological Studies (VOTES). The thesis concludes that DTN can address the issue of realising and establishing trust between systems or agents within the e-Health domain, such as the clinical trials domain.

Acknowledgements

I would like to thank all the people who have supported and encouraged me during my PhD studies. Though I may not mention all names I appreciate immensely the support I have received.

First and foremost, I wish to thank my supervisor Prof. Richard Sinnott for his numerous advice, unique guidance and motivation during my PhD. His support in every inch of the way in my PhD and life as a student has been tremendous. My sincere thanks also goes to Anthony Stell, John Watt, Susan McCafferty, Jipu Jiang, Christopher Bayliss and Gordon Stewart who are members of National e-Science Centre, Glasgow for their friendship and assistance at all times. Their support during my system development stages is deeply appreciated.

Secondly, I would like to thank my second supervisor Dr Peter Dickman for his ideas, encouragement and motivation that help shaped the direction of my research. I appreciate his suggestions and constructive criticism that guided me through my design.

Thirdly, I would like to thank my parents and brothers, Adewunmi, Bosun and Sola Ajayi for their love, encouragement and support that words cannot fully express. Thank you for believing in me and in my dreams. I also like to thank my wife, Beatrice Ajayi for her encouragement and understanding during my PhD. Her love, friendship and support are greatly appreciated. She made it easier for me to finish this thesis.

The work presented in this dissertation was supported by a grant from the UK Medical Research Council as part of the Virtual Organisations for Trials and Epidemiological Studies (VOTES) project. All the partners involved in the project are appreciated for their inputs. Finally, I would like to thank my friends for their timeless friendship and encouragement. Above all I give God the glory for all things, now and forever.

Contents

List of Tables

List of Figures

Glossary

AA	Attribute Authority
ABAC	Attribute-based access-control
ACL	Access Control List
ADF	Attribute Decision Function
AEF	Attribute Enforcement Function
API	Application Programming Interface
BGP	Border Gateway Protocol
CA	Certificate Authority
CHI	Community Health Index
CORBA	Common Object Request Broker Architecture
CVO	Clinical Virtual Organisation
CVS	Credential Validation Service
DAC	Discretionary Access-Control
DCOM	Distributed Component Object Model
DN	Distinguished Name
DoA	Delegation of Authority
DoS	Denial of Service
DTN	Dynamic Trust Negotiation
GSI	Grid Security Infrastructure
IBAC	Identity-based access-control

IdP	Identity Provider
LCAS	Local Centre Authorisation Service
LCMAPS	Local Credential Mapping Service
LDAP	Lightweight Directory Access Protocol
LSLV	Locally Stored Locally Validated
MAC	Mandatory Access-Control
MAC	Message Authentication Code
MLS	Message Level Security
NAS	Negotiated Attribute Store
OASIS	Organisation for the Advancement of Structured Information Standards
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PGP	Pretty Good Privacy
PIP	Policy Information Point
PKC	Public Key Certificate
PKI	Public Key Infrastructure
PMI	Privilege Management Infrastructure
RBAC	Role-based access-control
RDM	Role-based Delegation Model
RMI	Remote Method Invocation
RPC	Remote Procedure Call
RRLV	Remotely Retrieve Locally Validated
RSRV	Remotely Stored Remotely Validated
SAML	Security Assertion Markup Language
SOA	Service Oriented Arhitecture
SoA	Source of Authority

SP	Service Providers
TCP	Transmission Control Protocol
TIP	Trusted Intermediary Party
TLS	Transport Layer Security
TM	Trust Management
URI	Uniform Resource Identifier
VANGAURD	Virtual Anonymisation Grid for Unified Access to Remote Clinical Data
VO	Virtual Organisation
VOTES	Virtual Organisation for Trials of Epidemiological Studies
WAYF	Where Are You From
WS	Web Services
XACML	eXtensible Access Control Markup Language

1 Introduction

This chapter introduces the context, challenges, research motivation and contribution of this thesis. It concludes with a thesis statement and summarises the overall thesis outline.

1.1 Research Motivation

Access control in decentralised collaborative systems presents huge challenges when many autonomous entities such as organisations, humans, software agents from different security domains seek to dynamically access and share resources in a secure and controlled way. It is largely understood how to control access to resources within a given domain, however considerable challenges remain with regards to decentralised access control between collaborating autonomous remote entities as typified through Grid based collaborative research. The ideal solution would be a scalable distributed security approach where trust is easily discovered and realised and used to securely extend site autonomy to support collaborative work in a dynamic manner.

Currently there are numerous ways of controlling access to remote resources. Solutions based on a global schema for federation administration [1] or centralised policies have been used to provide access control to remote resources. Nowadays, however, access control across boundaries is an increasingly complex activity when collaborating partners are autonomous as regards to what is shared, how it is shared, how it is described or structured and who can access it [2]. This problem is further exacerbated when dynamic sets of heterogeneous resources are to be federated. The problem essentially is how to relate and exchange security attributes and other essential security information to support the federation from one organisation to the other without weakening any given site's security policies or infrastructure more generally [3]. Another way this problem has been explored is through semantic heterogeneity [4] of security credentials where the focus is on a credential's context realisation and how credentials relate with one another across boundaries [5, 6].

Often organisations are aware that certain resources exist in other organisations but usually will have to negotiate to obtain the required access rights or privileges with those target organisations to access their resources. In some cases, a target organisation publishes those resources they are willing to share with other organisations on a per collaboration basis. In other words an organisation or a user in an organisation will know that certain resources exist and yet lack appropriate credentials to access them. When this happens, the requesting organisation will initiate a negotiation process with the target organisation for privileges typically given as security attributes for resource sharing. These agreements are often difficult to reach because of different organisational security requirements and models. The agreement challenge is exacerbated when the number of organisations, users and resources involved in the agreement stage is large and dynamic. A fundamental cause of this is the lack of co-ordination and acceptance of agreements by the wider communities. A common approach for inter-site security policies is pre-exchange of security credentials between organisations. Among the disadvantages of this method are (a) credential revocation (b) credential re-distribution and (c) credential duplication/redundancy, which arise when a credential is revoked and/or re-distributed when security policies become invalid, extended or changed [7, 8]. Credentials may also become redundant or duplicated as new credentials are distributed.

The e-Health domain is no exception to this, and needs ways in which clinical researchers, health providers and associated IT staff can successfully and seamlessly share health information if they are to improve or make available quality healthcare services. One major area that aims to improve healthcare services is research into, and support of, epidemiological studies and clinical trials. Conducting such studies demands that detailed collaborative agreements between various healthcare providers, partners and researchers are in place. Usually, these collaborative agreements are given as specific agreements (protocols) governing who is involved, where and how collaboration is to be achieved, and importantly how the data collected can be used [9]. These collaborations require data to be shared between parties and across boundaries; hence the need to control access to shared resources.

In most cases, when an organisation advertises the availability of some of their resources (for resource discovery), they already have local access control policies in place that protect them from unauthorised use. For example, an NHS hospital might be willing to make available some statistical data to any organisation within the health services or health-research institutions. However, proving your identity (authentication) as a researcher from a health-research institution is not sufficient to guarantee access without proper privileges (authorisation) required for local security policies that may exist at a given hospital. In the health domain

there is an urgent need to share and collaborate, e.g. to identify potential clinical trial participants, seek consent to access patient information, recruit participants, collect data from on-going trials and manage on-going studies. All these will require access to electronic health information such as patient demographics, patient medical history, lab/test results, current treatment notes, past and current prescriptions. However, the lack of access to this geographically and autonomously distributed information may delay a trial or affect the success of a trial. Thus, the research undertaken are reported in this thesis has been conducted to address these issues.

1.2 Research Contribution

This thesis introduces a novel model called *Dynamic Trust Negotiation (DTN)* to address the heterogeneous and autonomous federation of credentials and policies. The model describes how trusted intermediary parties can provide multiple negotiation and delegation hops to help establish trust between strangers or non-collaborating institutions. The model prevents the disclosure of credentials and access policies, and reduces credential semantic issues that exist in decentralised systems [10, 11], such as the Grid.

In the context of access control, this thesis defines trust as the possession of *authentic* and *valid* credentials necessary for access control at an end point - typically a target with access control policies defined by the target resource providers. A credential is either *valid* and *authentic* or only *authentic*. An authentic credential implies a verifiable and un-tampered credential, while a valid credential implies a semantically correct credential that is acceptable, useable and tenable to an end point, e.g. a data service. Trust negotiation aims at delivering valid credentials that are authentic, and able to satisfy an access policy.

Dynamic trust negotiation (DTN) itself is the process of realising trust between strangers or two non-trusting entities, e.g. institutions, through trusted intermediary entities. Trust is realised when an entity delegates its digital credentials to trusted intermediary entities through which it can subsequently interact with previously non-trusting entities. These intermediary entities can in turn delegate to other intermediary entities resulting in *n-tier* delegation hops. Any entity can serve as a negotiator for other entities provided it is trusted by the two non-trusting entities or by their intermediaries.

DTN explores how credentials can be negotiated as the basis to support collaborative research between autonomous, distributed resources. It addresses the heterogeneity and autonomy of trust management credentials and policies in multi-domain environments. DTN negotiates

credentials between trusted parties also known as a *circle of trust (COT)*, who act as mediators on behalf of strangers and thus bridge trust gaps. This bridge also reduces the risk associated with disclosing policies to strangers.

DTN is based on a peer-to-peer model where organisations in a federation establish trust and negotiate security attributes both statically and dynamically with neighbouring organisations forming what is known as peer trust. Peer trust is a peer-to-peer trust model that exists between peers based on reputation, past experience or mutual agreements. This peer-to-peer trust model supports the establishment and subsequent interactions between trusted and non-trusted organisations, and serve as the basis for dynamic negotiations. In doing so it achieves the following objectives:

- it provides an alternative to the global attributes ontology approach that defines what attributes or credentials are, what they are used for, and where they are used;
- it supports inter-domain authorisation issues;
- it supports access to resources across organisational boundaries;
- it stays compatible with existing privilege management infrastructures;
- it enables the negotiation of security credentials through delegation of roles;
- and it dynamically supports the discovery and establishment of chains of trust.

This thesis's contribution to knowledge is in the area of discovering and realising trust in an open and decentralised environment such as the e-Health environment. The originality of the work includes:

- the design and development of Dynamic Trust Negotiation (DTN);
- the application of routing algorithms for trust discovery;
- the application of trust contracts for trust negotiation;
- and the application of trust contracts and negotiation hops to trust realisation.

1.3 Supporting Publications

The following publications have so far been made as a result of this research and other related work:

- O. Ajayi, R. Sinnott, and A. Stell. Towards Decentralised Security Policies for e-Health Collaborations. In Proceedings of 2nd International Conference on Emerging Security Information, Systems and Technologies, (SECURWARE), Cap Esterel, France. IEEE Computer Society, Aug. 2008.
- O. Ajayi, R. Sinnott, and A. Stell. Blind Data Aggregation from Distributed, Protected Sources: The Future Model for Security-oriented Collaborations. Workshop of the UK E-Science All Hands Meeting, Edinburgh, UK, 2008.
- O. Ajayi, R. Sinnott, and A. Stell. Dynamic Trust Negotiation for Flexible e-Health Collaborations. In Proceedings of 15th Mardi Gras Conference, Baton Rouge, USA. ACM Digital Library, Feb. 2008.
- O. Ajayi, R. Sinnott, and A. Stell. Trust Realisation in Multi-domain Collaborative Environments. In Proceedings of 6th IEEE International Conference on Computer and Information Science, ICIS07. IEEE Computer Society, July 2007.
- O. Ajayi, R. Sinnott, and A. Stell. Formalising Dynamic Trust Negotiations in Decentralised Collaborative e-Health Systems. In Proceedings of the 2nd International Conference on Availability, Reliability and Security, (ARES07), Vienna, Austria. IEEE Computer Society, Apr. 2007.
- O. Ajayi, R. Sinnott, and A. Stell. Trust Realisation in Collaborative Clinical Trials Systems. In HealthCare Computing Conference HC2007, Harrogate, England, Mar. 2007.
- R. Sinnott, O. Ajayi, A. Stell, and A. Young. Towards a Virtual Anonymisation Grid for Unied Access to Remote Clinical Data. In 6th International HealthGrid Conference, Chicago, USA, June 2008.
- R. Sinnott, J. Watt, J. Jiang, and O. Ajayi. Shibbolth-based Access to and Usage of Grid Resources. In Proceedings of IEEE International Conference on Grid Computing, Barcelona, Spain, Sept. 2006.
- R. Sinnott, J. Watt, J. Jiang, A. Stell, O. Ajayi, and J. Koetsier. Single Sign-on and Authorization for Dynamic Virtual Organizations. In 7th IFIP Conference on Virtual Enterprises, PRO-VE 2006, Helsinki, Finland, September 2006.
- R. Sinnott, A. Stell, and O. Ajayi. Development of Grid Frameworks for Clinical Trials and Epidemiological Studies. In Challenges and Opportunities of HealthGrids - Proceedings of Healthgrid 06, volume 120, June 2006.

- R. Sinnott, A. Stell, and O. Ajayi. Initial Experiences in Developing e-Health Solutions across Scotland. Workshop on Integrated Health Records: Practice and Technology, Edinburgh, Mar. 2006.
- A. Stell, R. Sinnott, and O. Ajayi. Secure, Reliable and Dynamic Access to Distributed Clinical Data. In Proceedings of Life Science Grid Conference, Yokohama, Japan, Oct. 2006
- J. Watt, R. Sinnott, O. Ajayi, J. Jiang, and J. Koetsier. A Shibboleth-Protected Privilege Management Infrastructure for e-Science Education. In Proceedings of 6th International Symposium on Cluster Computing and the Grid, CCGrid2006, Singapore, May 2006.

1.4 Thesis Statement

This thesis asserts that it is possible to discover and realise trust through circles of trust and exchange security credentials in a collaborative e-Health environment. Trust realisation enables decentralised access control for various autonomous and heterogeneous e-Health organisations. The dissertation assumes the existence of a means of authentication across e-Health domains either by federated or centralised authentication mechanisms. The focus in this dissertation is on security credentials or attributes that are used for trust negotiation and access control.

1.5 Dissertation Outline

This dissertation presents dynamic trust negotiation (DTN) and how it is used for the discovery and realisation of trust between heterogeneous and autonomous entities, thereby supporting decentralised authorisation in collaborative environments. Chapter 2 reviews background literature in the area of security, access control and trust management. The chapter concludes with a review of security frameworks in the context of e-Health collaborations. Chapter 3 presents DTN objectives and gives an overview of the design of a DTN system. Chapter 5 describes the formal model of DTN in decentralised collaborative e-Health systems. Chapter 6 presents the syntax and semantics of the DTN policy language. The chapter describes in detail various policies that are used in DTN, including policies based on trust contracts. Chapter 7 describes the details of two protocols used in the DTN framework for discovery

and negotiation. Chapter 4 describes the overall DTN Architecture explored in this thesis and shows how it uses a discovery system and negotiation system. Chapter 8 discusses the area of clinical trials and presents scenarios that have served as the basis for testing the DTN framework described in this thesis. Chapter 9 describes the DTN implementation undertaken in the Virtual Organisation for Trials of Epidemiological Studies (VOTES) project and also discusses DTN performance. Finally, chapter 10 presents and discusses the conclusions of the dissertation and identifies potential areas of future work.

2 Literature Review and Related Work

This chapter reviews background literature in the area of security and other related work loosely associated with supporting e-Health collaborations. The discussion focuses on access control and trust. As will be discussed trust is a fundamental base for security. The first section discusses security in decentralised and open environments while the second section reviews existing access control models and approaches. These lay ground work for any security contributions. The third and fourth sections review privilege management infrastructures and security standards from the Grid and open systems perspective respectively. Negotiation and trust management are discussed in the fifth and sixth sections that serve as basis for this thesis. Decentralised access control from the Semantic Web perspective is discussed in the seventh section. The chapter concludes with reviews of other work or projects that have tried to address security in the context of e-Health collaborations, since the health domain is the context for this thesis and these projects raise issues that this thesis aims to address.

2.1 Security in Decentralised and Open Environment

Security can be viewed from a centralised or decentralised perspective. Each of these perspectives introduces various security challenges since their properties and environments differ. One of the main properties of a centralised security system is that a single pivotal point exists, from which security can be marshalled, co-ordinated and managed. In a decentralised security system, however, a single pivotal point does not exist. Indeed many such points will co-exist.

Two types of environment can be classified, closed and open environments. A closed environment is one in which tight control exists over a number of issues such as systems, users, resources and infrastructure. An open environment can be seen as a more liberal environment where each component in the environment is to an extent free of one another. For example, the Windows operating systems, can be regarded as systems designed in a closed environment with little or no input from external sources. The Linux operating systems are designed in

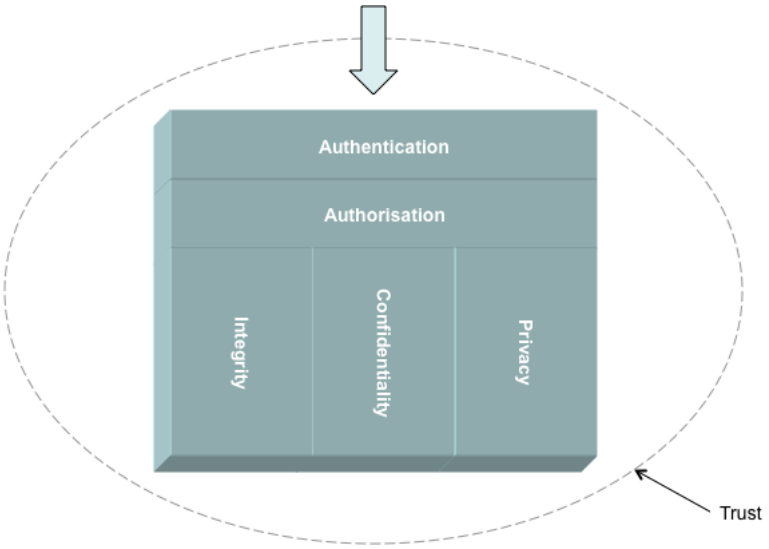


Figure 2.1: Security Concepts

open environments where anyone from anywhere can, in principle, input into the design and direction of the system. A key advantage of open environments is their collaborative nature. This is not to say that collaboration does not exist in closed environments.

Arguably, security, from a centralised perspective, could exist in closed or partially closed environments, whereas security from a decentralised perspective that could exist in either closed or open environments. The security challenges facing decentralised and open environments are enormous [12, 13]. Overcoming some of these challenges are the basis for this work. One prime example of an open environment requiring a decentralised security model is the Grid paradigm. Grids exemplify the challenges of security in decentralised and open environments, related works in security of Grid systems and are discussed in this chapter.

2.1.1 Security Concepts

Security is a broad topic of research but some principal concepts are worth mentioning here. This is not to disregard other concepts or other aspects of security, but rather to emphasise those that underpin this thesis. The concepts shown in Figure 2.1 include: Authentication, Authorisation, Confidentiality, Integrity and Non-repudiation. The concept of trust is also discussed in this section, since it underpins each of these concepts.

Authentication: Authentication is the identification and assurance that a subject is who they claim to be. It is the assertion of the ownership of an identity. A subject's identity is usually verified when a proof of identity is provided. For example, an identity may be proved or verified when a username and pass phrase are presented and successfully validated against a stored phrase or key. Alternatively an identity may be verified when a valid digital certificate is presented along with data signed by the subject [14, 15]. Usually the certificate is signed by a mutually trusted third party and the certificate binds the identity of a subject to their public-key. A parallel example of a digital certificate is a drivers license or passport.

Authentication models in use today include: trusted third party and web of trust [16]. Unlike the trusted third party model, the web of trust is not hierarchical in nature. The web of trust model allows a party to verify an entity's identity by verifying any one of the many signatures used to sign a message. The signature that is trusted the most is used for the verification. In the UK e-Science community, a centralised certificate authority (root CA) has been adopted as a trusted third party used for the verification of certificates. The main advantage of centralised over decentralised CAs is its single point of control and reduced ambiguities, thus allowing support for single sign-on. However, the main disadvantage of centralised authentication is its single point of failure, where a compromised root CA affects verification and causes downtime or, in the worst case, compromise of any site where CA certificates have been used. In contrast, decentralised authentication does not suffer when one of the CAs are compromised or invalidated. Rather only those sites that have used certificates from that CA are affected. The main advantage of decentralised authentication is its flexibility and scalability. Since there is no single point of failure and as multiple sites participate in the up keeping of authentication data. Whilst the main disadvantage is that it seldom takes a while for the changes or updates to propagate or for notifications to be received.

Authorisation: Authorisation is the validation that a subject has the required privileges to access a resource. Usually authorisation is achieved through some sort of access control policies, restricting access to protected resources for privileged users or entities. An access policy primarily indicates what actions a subject is authorised to perform on an object or the capabilities of a subject in a system. It typically defines the context, attributes and constraints that must be satisfied before access can be granted to an object. Today, numerous authorisation models are in use often based on one or more policy specifications. Some of the policy specifications that exist today include: discretionary, mandatory and role-based policies. These policies are discussed in more detail in [1], while [13] compares

and contrasts centralised and decentralised authorisation models. An assumption made by most authorisation models is that a subject's identity has been validated prior to the system deciding access control. Thus it is always required that a subject is authenticated before he or she can be authorised.

Confidentiality: Is the assurance that information either stored or in transit can only be accessed by authorised entities. Using cryptographic techniques, confidentiality can be improved to protect against man-in-the-middle attacks [14], but can be compromised where the shared key or private key are exposed, for example. In some cases, both shared key and public-key cryptography are used to achieve better performance. Today, technologies such as HTTPS using Transport Layer Security (TLS) [17] provide secure point-to-point connection through which data can be transmitted securely between endpoints. However, when intermediary parties such as proxies need to access and work on data being transmitted, TLS does not ensure end-to-end security. End-to-end security ensures that a message or parts of a message are encrypted and can only be viewed by the intended recipient regardless of the nature of connection or intermediaries that are required to work on parts of the message. As such, technologies using Message Level Security (MLS) [18] are often preferred for confidentiality since they ensure end-to-end security.

Integrity: Is the assurance that an unauthorised modification of data has not occurred in transit or generally. This implies that anyone should be able to read or make use of the data with certainty that the data has not been tampered with or altered by an unauthorised entity. Encrypting data with a private-key ensures that only the person with private-key can modify it while others with the associated public-key can read or open it. In practice, for performance reasons, it is normally the case that the digest of the message is taken and encrypted with the private-key. The recipient receives the digest and decrypts the encrypted digest with the public-key. The decrypted digest is then compared with the received message to verify the message integrity. Message digests are created using a checksum or one-way hash algorithms such as MD5 (128-bit hash value) [19] or SHA-1 (160-bit hash value) [20]. Reliability of these hash algorithms against attacks depend on the size (in bits) of the hash values.

Non-repudiation: Non-repudiation is the assurance that transactions once performed by a subject are undeniable. This is a requirement that cryptography by itself cannot satisfy. Non-repudiation requires the generation, verification, storage and tracking of evidence and

facts in order to resolve disputes that may arise. Usually the process of dispute resolution involves trusted third parties that validate tracked and stored evidence such as certificates, signatures, transaction details and time stamps.

Trust: Trust is the underlying phenomenon of security concepts. Trust is built on the concept of limiting expected behaviour [16]. It is associated with an assurance measurement. The level of confidence in limiting behaviour within a security context determines the level of assurance. From an authentication point of view, trust defines the level of assurance that should be associated with identity. From an authorisation point of view, trust defines the expected behaviour of an entity in possession of security credential, i.e. what the entity should have access to or what privileges they can have. From a confidentiality point of view, trust is the assurance or confidence associated with behaviour, e.g. the confidence in certain entities to keep information secured and protected. From an integrity point of view, it is the assurance of an expected behaviour. For non-repudiation, it is the assurance that an action or behaviour is undeniable.

2.1.2 Security Threats

A threat is a hostile entity or agent that can intentionally or otherwise disclose or modify the information managed by a system [21]. Threats can include anything e.g. events that violate the security of a system. Security violations of a data system, for example, may include improper access, modification or deletion of data. Confidentiality, information integrity and system availability are three main categories where the consequences of violations can be observed.

- *Confidentiality* – preventing/detecting/deterring the improper disclosure of information. Confidentiality violation is caused when data is accessed either intentionally or otherwise by improper users or systems. In some cases this can be caused when users who are authorised to access certain information can infer unauthorised information often termed statistical disclosure [22].
- *Data Integrity* – preventing/detecting/deterring the improper modification of data. Violation of data integrity can be caused through improper modification of data. Data modification can occur even in situations where the data itself cannot be read. That is data integrity can be violated regardless of whether confidentiality is violated.
- *System Availability* – preventing/detecting/deterring the improper use of a service.

Violation often affect system availability. This can be caused by actions that prevent a system or service from being accessible by authorised users. One example of this might be denial of services [23].

There are two classes of users that can give rise to security threats: authorised users and hostile agents [21]. Authorised users can abuse their privileges and authorities to violate the security of the system. Hostile agents can include internal users and external users who carry out improper actions to violate a system for a variety of reasons. Internal users can be unauthorised users who despite their lack of privileges carry out improper actions to access data they have not been authorised to use. In some cases hostile agents use applications (legally or illegally), programs and codes such as viruses to attack and violate systems.

2.1.3 Cryptography and Public Key Infrastructure

Public Key Infrastructures (PKI) [24, 25, 26] are technologies based on the use of public-key encryption and/or digital certificates for security implementation in distributed applications. Cryptography (or Encryption) is a method or technique of concealing information from eavesdroppers. It dates back to time immemorial and it has evolved over the years. Two well known cryptographic methods today are symmetrical and asymmetrical cryptography.

Symmetrical cryptography is a method where both parties at the end of the communication share and keep a key. The key is used for both encrypting and decrypting information. The strength of the method is that it is fast but the weakness is that the key must be kept secret. Examples of this method include algorithms like DES [27], Triple DES [27], Blowfish [28] and AES [29].

Asymmetrical cryptography also known as public-key cryptography is a key-pair method - often based on a public key and a private key. It is the basis for PKI. The keys are different, but are mathematically constructed so that if you encrypt data with one key you can only decrypt it with the other. In practice, the owner of the keys gives the public key to whoever they want to communicate with whilst they keep the private key. More than one person could have the public-key, hence why it is called public-key, but only one person should have the private-key. If a message is encrypted with a public-key, only the person with the private-key can access it. But if a message is encrypted with a private key, only those with the public-key can access it. The benefit of this method is that the receiver of the information may be sure of the sender (as long as the private-key has not been compromised). The issue of this method is: how do we know if the private-key has been compromised? This is an issue of trust and

the basis of digital signatures - that is identifying whose signature it is. Other issues are: how do we know that the sender is who they claim to be? In practice this is resolved by the existence of a trusted third party such as a CA. The sender provides a certificate that has its name and public-key amongst other things. This certificate is signed by the CA before the certificate is issued. The CA acts like a cheque guarantor, who is expected to have validated the content of the certificate.

Current PKI standards use X.509 standard [30] which describes the structure used for digital certificates. With PKI, confidentiality can be achieved through encryption, data integrity through a digital signature and mutual trust through CA infrastructures. However PKIs are not without their own issues and they do not solve all domain security needs, e.g. those of the e-Health domain. From the end-users perspective, PKIs are difficult to implement [31] and keys are difficult to be kept secret. For instance, users are often required to transform their certificates to other useable forms; to configure their systems to use these certificates; store their certificates in a secure place, as well as not storing multiple copies of their certificate which often happens when backing up their systems. In addition, users often write down their *strong* private key passwords in unsecured places because of the difficulty in remembering their passwords. Lastly, where a private key has been compromised it is difficult to detect and this can only be done retrospectively. In the UK e-Science and other Grid communities, PKI has been widely accepted and it now underlies most security solutions used in e-Infrastructures. This is partly because it is supports the notion of single sign-on. The root CA used by the community is widely seen as a basis for trust but only in the area of authentication and not authorisation (i.e. access control).

2.2 Access-control Models

Access-control models provide the formal representation for describing and implementing security policies. Security policies in the context of access control refers to the set of rules and practices that an organisation defines to manage who can have access to what resource [16]. Access-control models provide a means of analysing whether a security system conforms to an abstract model in its design and implementation, and satisfies the same properties that the abstract model satisfies. Today, a number of access-control models exist and have been in use to analyse access control policies. These models can be mainly grouped into two categories: mandatory and discretionary models.

2.2.1 Mandatory Access-control

Mandatory access-control (MAC) models manage access for information based on the classifications of subjects and objects in a system. Subjects are entities that access objects and objects are information entities such as resources and services. Mandatory policies drive these models, which classify subjects and objects based on the sensitivity of the information. Subjects are initiators like users or system processes, while objects are targets like database tables and actions are activities like read or write. The classification given to a subject expresses the level of trust assigned to the subject. The object classification expresses the risk level assigned for that object. To access an object, a subject must belong to the object security class.

Mandatory policies are predefined and remain static. They are invariant and cannot be changed dynamically. Policies are defined by administrative officers that set up who has access to what resource based on the sensitivity of the resource. Resources that belong to the same classification have the same access control applied to them. This implies that a subject with a security classification label can access all resources that have the same confidentiality label. A lack of a discretionary property is mostly unacceptable in todays commercial environment, as fine-grained access control (in terms of subject, objects and actions) are often preferred, e.g. database management systems. In addition to this issue, MAC does not allow access rights to be transferred from one subject to another.

Mandatory policies are suitable for certain kinds of environments such as the military, where users and resources can be classified. MAC models provide a level of protection against hostile agents due to its flow-control nature. Information flows are prevented from objects with higher classification to lower classification. For example, a user at a specific trust-level is prevented from accessing resources or processes at a higher trust-level. Even so, malicious code embedded in a program running at a specific classification can (in principle) be used for disclosing information to a different classification.

2.2.2 Discretionary Access-control

Discretionary Access-control (DAC) models manage access to information based on the identity of subjects and rules that are specific to each subject and object in the system. The rules which are defined by discretionary policies specify the types of access a subject is allowed for the object. [21] defines discretionary as *"the possibility that exists for owners to grant and revoke access rights on some objects"*. Discretionary policies are owner-centric and require

each object in the system to be owned by one or more entities. These owner entities define who can access the object and in what mode it can be accessed. A subject is granted access to an object where the subject satisfies authorisation rules that have been defined for the object.

Today, a DAC policy is widely adopted and used as the basis for most implemented policies because of its ease, simplicity and how it relates to many access situations. The ability to own, grant and revoke access rights enables the decentralisation of administrative control and is one of the reasons for its wide adoption. However, one issue with DAC is that it does not protect the flow of information and can thus give rise to containment issues. Similarly, the propagation of access rights that DAC supports is unbounded and often difficult to determine in systems.

Though discretionary and mandatory policies are different, they are not mutually exclusive. They can easily be combined to compliment each other in addressing vulnerabilities in providing access control. Mandatory policies can be used to provide authorisation control while discretionary policies can be used to provide access control. For instance, can user X request resource Y? If he can, what kind of action - read/write, can he perform on resource Y? Any request that satisfies the discretionary control will also satisfy the mandatory control.

The access matrix model [32] is one of the most widely used security models. The model can be used to represent most policies including discretionary and mandatory policies. However, it is more widely applied for discretionary policies.

2.2.3 Access Control List vs. Capability

Access Control Lists (ACL) and Capabilities [16] originate from the Access matrix model [32]. The access matrix is a conceptual model that details access rights that each subject has for each object. The access matrix shown in Figure 2.2 contains a row for each subject and a column for each object. Each cell of the matrix contains the access right for that subject on that object. The problem with implementing an access matrix is that they are large and often difficult to maintain since they describe a whole system. Similarly many cells may be redundant. Other issues are how to classify and/or categorise objects along with what type of access rights they require.

One of the main reasons for access control is to prevent unauthorised access to objects and resources. Discretionary and non-discretionary (mandatory) access controls [16] are two common mechanisms for achieving control. Discretionary access controls are based on the

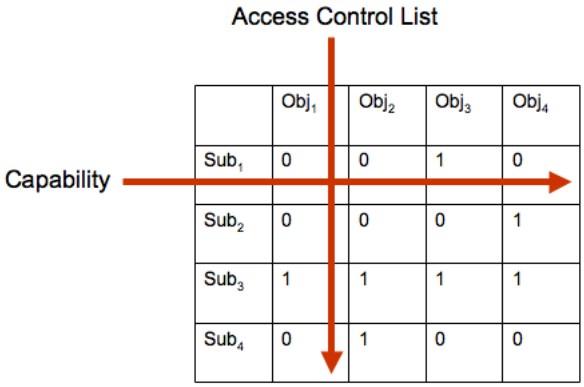


Figure 2.2: Access Matrix

access matrix model and are often implemented using an ACL and/or Capability. The method is called Discretionary because access rights to objects are at the discretion of the object owner or provider. Non-discretionary access controls are based on access classes consisting of two parts: sensitivity level and sets of categories. Non-discretionary access controls are mostly use by the military and are not generally implemented in non-military systems. The lattice security model [33] is a common model for mandatory access controls.

An ACL assumes the columns in an Access Matrix, which for each object consists of a list of subjects with their associated access rights. The Capability assumes the rows in an Access Matrix and provides a list for each subject that consists of objects, the location of objects and the access rights that the holder has for each object. Both ACL and Capability have to be protected to prevent authorised alterations of access rights known as containment and thus prevent unauthorised access to objects (resources). [34, 35] discusses containment issues in respect to both ACL and Capability systems.

An ACL defines who can do what on an object and is widely implemented in systems today because they are both easy to implement and provide better performance compared to Capability systems [21]. Each object is provided with a list that shows the access rights for each subject in the list. With ACL, access to objects can easily be reviewed and access to objects can easily be revoked. On the contrary, access rights of subjects can not easily be reviewed or access to subjects revoked since the system will have to iterate through all ACLs of every object. An example of the ACL is the Grid *mapfile* in globus-based systems. A Grid *mapfile* holds a list of subjects who can access a resource along with the access rights the subject

possess [36, 37, 38].

A Capability defines a subject's rights for an object. Each subject is defined with a list (a list of capabilities) that details what objects they can access and what rights they have for those objects. Capability systems assume that the use of capability is both necessary and sufficient for access to an object by subjects. Apart from implementation and performance issues of capability systems, capability problems stem from the ability to revoke all access rights to an object and also with the ability to review access to an object, since all of the subject capabilities will have to be visited to review or revoke access to an object. An example of a capability in a Grid system is an attribute certificate [39]. An attribute certificate stores all of the attributes or access rights that a subject holds for various resources (objects). Similar to capabilities in a capability based system, attribute certificates are signed to avoid tampering. In addition, capability systems either restrict the rights and/or shrink the size of a capability list, a process known as refinement. Refinement is analogous to a system where a subject makes available some of their attributes for authorisation.

In Grid systems today, solutions like PERMIS [40] combine both ACL and Capability security models to provide a secure access control system. Although PERMIS is a role based policy system, it leverages the benefits of discretionary access control by providing policies that list subjects and granted actions on specific objects similar to an ACL. At the same time PERMIS can be configured to make use of subject attributes based on policies to make access decisions similar to Capability systems. Shibboleth [41], on the other hand, provides a means of exchanging attributes (credentials) between domains. These attributes are in essence capabilities of what a subject is allowed to do in a domain. Shibboleth could be used to support Capability systems across domains where security attributes have been pre-agreed.

2.2.4 Identity-based Access Control

Identity-based access-control (IBAC) is a model that manages access to information and resources based on a subject's identity. Identity is the possession of proof that uniquely describes an entity or subject. Identification is the process of establishing the identity of a user or entity while authentication links the identity of a subject to a user/entity. Identity-based access control manages the access to information based on the identity of a subject that wants to perform an operation on an object [42]. Identity management, albeit digital identity management, is the process of managing, creating, using and verifying identities.

Security policies used in IBAC systems define the permissions associated with a resource for

an asserted identity. Most identity-based access control systems in use today are modelled based on the access control list discussed in Section 2.2.3. Using ACL, if a subject entry exists in the list, and the operation to be performed is part of the entry for an object, then access to the object is permitted for that subject. A typical example of an IBAC that uses ACL is the grid-*mapfile*, which is discussed later in Section 2.3.4. As in ACL, IBAC supports discretionary policies, which requires object ownership – each object in the system is owned by one or more entities. Usually, object ownership lies with the creators of the object.

However, in some environments resource ownership does not lie with subjects but rather with the organisation. ACL in these environments have been modelled to use a "group" as an entry instead of a subject. A group in this case is a collection of subjects. So if a subject's identity is found in a group and an entry exists in the ACL for that object, the subject is permitted to perform the operation defined in the ACL.

2.2.5 Role-based Access Control

Role-based access control (RBAC) manages the access to information based on the role of a subject that wants to perform an operation on an object. The notion of "group" is mostly similar to the notion of "role". A role is the representation for the collection of users and a user can be a member of one or more roles. In RBAC, a group is equivalent to a role if [42]:

- there are no restrictions on the number of groups that could exist;
- there is no upper limit on the number of users that could be members of a group;
- there are no restrictions on the number of groups that a user can simultaneously be a member of.

Many RBAC models [43, 44, 45, 46] have been described but [44] is widely accepted as the primary model. The main elements of RBAC are users, roles and permissions. Users are assigned to roles and roles are given permissions. Permissions specify the privileges permitted on objects. Permissions given to roles are permitted to all users in that role. It is important in RBAC that users are assigned to roles that reflect specific privileges they require for their task. This requirement is well known as the least privilege principle [21, 47].

RBAC is largely used today because it specifies and enforces security policies in a way that reflects the organisation structure. An ACL can be used to model RBAC if groups are used fundamentally as roles in the ACL model. Because of this, ACL have been widely used for the implementation of RBAC in organisations. The use of roles makes it easy for users to switch

between roles thus dynamically changing the permissions they have. In a large organisation with many dynamic users, this notion of roles makes RBAC scalable and simplifies policy administration.

In [44], roles are allowed to contain other roles, which introduces role hierarchies. A role hierarchy is defined [42] *as a strict partial ordering on the set of roles.* In a role hierarchy, a higher role inherits the permissions of lower roles, through the inheritance relation that exists between the two roles. Hierarchies between roles further simplifies policy administration.

The RBAC model also introduces role constraints and combines role hierarchies with role constraints. Through constraints, RBAC can support the principle of separation of duty [48] through mutual exclusion of roles. In addition to constraints, other RBAC-related models have introduced role delegation and revocation [49, 50, 51, 52]. Some have also introduced parameterised roles and obligations [53] to optimise the number of roles and to provide more fine-grained access control. The fundamental benefit of RBAC is that it is based on user attributes and not on identity. This attribute concept is useful in the area of trust management where the identity of a user is initially or often unknown. For example the decision to allow a person entry into a building is often not based on their identity but on whether they hold a required credential – an access card, key or fob. The benefit of this is that the entry system is manageable since the access decision is not based on the identity but access attributes that can easily be assigned, person to person.

2.3 Privilege Management Infrastructures

Privilege Management Infrastructures (PMI) are based on the use of privileges to enforce access to resources. Privileges are essentially attributes that determine what type of access or permission a user or entity has. One common form of attribute is descriptive attributes which include roles, access levels and group membership. Privilege management [54] entails the definition, assignment, presentation, delegation and revocation of attributes, and their enforcement by authorisation infrastructures.

A PMI is analogous to a PKI where a PKI uses user certificates for authentication, so a PMI uses attribute certificates for authorisation. Most PMI-based systems include an attribute issuing service and a policy engine. The software components of a PMI can be centralised or distributed depending on the application and often support a push or pull model approach for attribute retrieval/delivery. A policy engine comprises, amongst other things, a Policy Decision Point (PDP) and a Policy Enforcement Point (PEP) [55]. The PDP

makes authorisation decisions by checking through collections of rules that are associated with a target, while the PEP ensures that all requests to access a target are authorised through the PDP. Various PMI infrastructures have been developed and some of these are outlined here.

2.3.1 Privilege and Role Management Infrastructure Standard Validation

The Privilege and Role Management Infrastructure Standard Validation (PERMIS) [40, 56] is an authorisation infrastructure that uses role based hierarchy policies to achieve fine grained access control. It can work with most authentication systems such as Kerberos [57], PKI, Shibboleth [41] and username/password systems. The infrastructure provides both an attribute/policy issuing service and a policy engine. Other PERMIS components include:

- authorisation policies, which are signed by source of authorities (SoA);
- attribute tokens, which are X.509 based attribute certificates;
- servers such as LDAP servers, which are used to store policies and attribute certificates;
- Attribute Certificate Manager (ACM) tools, formerly known as the Privilege Allocator, used for issuing and managing attribute certificates.

PERMIS policies are written in XML and comprise sub-policies [54] thus requiring policy decomposition by the PDP. The policies provide information on subjects, source of authority (SoA), roles, targets and actions that can be performed. These policies are typically stored in LDAP servers. The PDP shown in Figure 2.3 makes its decision based on stored policies and presented user attributes communicated by the PEP. PDP decision response is either a deny or grant or an insufficient information. The PEP subsequently enforces this decision on access to the resource.

PERMIS gets its attributes in pull mode from attribute authorities (AA) but can also get attributes through push mode from AA or from the PEP. When a user makes a request for a resource in the push mode, the PEP for the resource picks up the request and contacts the PDP with the pushed user attributes while PERMIS pulls the policies for the particular resources. In the pull mode, PERMIS pulls user attributes from LDAP servers, which could be remote LDAP servers. The pull mode requires PERMIS to be configured with the location of LDAP servers where user attributes can be pulled from. This configuration requirement potentially limits PERMIS in the pull mode, which potentially makes pull mode less preferred to the push mode in supporting some virtual organisations (VO). Another downside for the

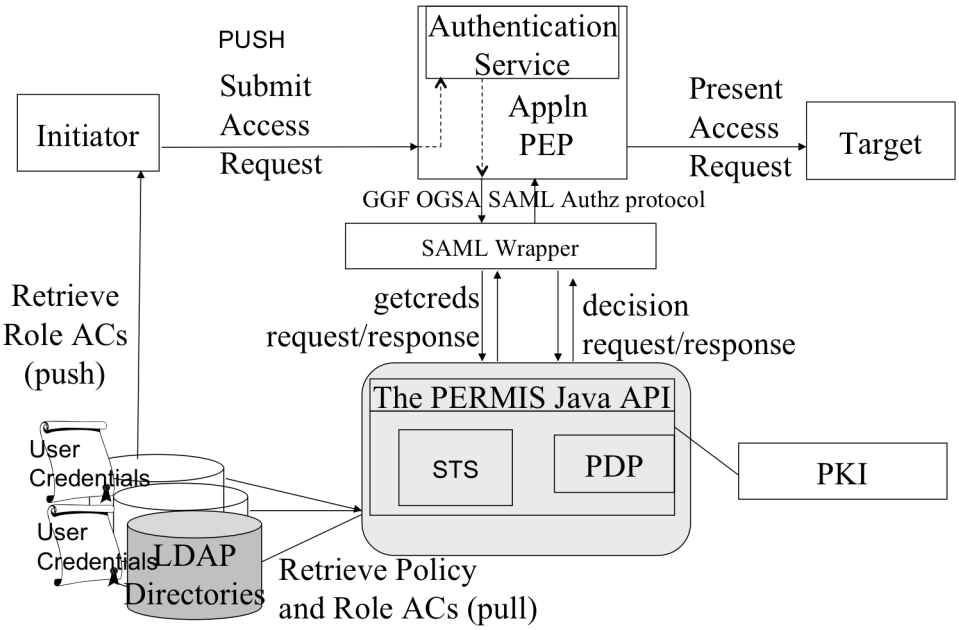


Figure 2.3: PERMIS Authorisation System [from JISC Middleware Security Workshop, 20-10-2005]

pull mode is that the user's privacy is not protected because the identity of the user, e.g. their Distinguished Names (DN) would have to be known for all attributes of the user to be retrieved. How user's attributes are retrieved in the push mode depends on the application implementation. For instance, a user could forward their attributes along with their request for resources or the application could use a Shibboleth [41] style approach for retrieving and forwarding user attributes. With Shibboleth, releasing a user's distinguished name (DN) could be prevented.

PERMIS supports dynamic delegation of authority, where remote SoAs are trusted to issue attribute certificates (AC) to their users and potentially to delegate the trust to their subordinates. However, roles assigned by remote SoAs must conform to a role assignment subpolicy that is provided by the root SoA. PERMIS also supports distributed credential management, where numerous attribute authorities (AA) are empowered to allocate credentials to users.

2.3.2 Virtual Organisation Membership Service

The Virtual Organisation Membership Service (VOMS) is a service designed to manage authorisation in a virtual organisation [58]. VOMS is a centralised attribute authority and it relies on other infrastructures such as the Grid Security Infrastructure (GSI) [59, 37] for authentication services. It also relies on other infrastructures such as Local Centre Authorisation Service (LCAS) [60] and the Local Credential Mapping Service (LCMAPS) [61] to provide a framework for Authorisation Decision Functions to grant or deny requests based on provided VOMS attribute assertions.

VOMS components shown in Figure 2.4 include user clients, servers aka VOMS server [54, 62], administration clients and an administration server. The user client identifies a subject's identity to the user server which are often at the subject's home location (origin). The user client queries the VOMS server using the subject's GSI proxy that it creates and the roles it wishes to use. The VOMS server in response extends the proxy certificate with the VOMS related attributes. These attributes are signed by the VOMS server as an assertion to the resource provider. This extended proxy certificate is provided to the PDP of the resource provider that then makes a decision based on the VOMS assertions. With LCAS/LCMAPS, this decision is about account should the user in this VO with this role be mapped to. VOMS is currently being rolled out across the UK National Grid Service (NGS) and is a common approach accepted by the UK oriented Grid community for specification of attributes. The VOMS assertion usually includes the VO name, assertion validity, and groups or roles or privileges. The administration client and administration server are used for VOMS server management. VOMS systems are an example of an IBAC based-systems Section 2.2.4.

VOMS is primarily an issuing service while PERMIS is both an issuing service and a policy engine. VOMS-signed proxy certificates can be used by non-VOMS aware authorisation decision functions since VOMS does not modify the body of an X.509 user certificate but only the extensions. In addition, VOMS provides a privacy mechanism for subjects in that subjects can choose how much information about themselves they are willing to expose [58]. One known vulnerability of VOMS is that it lacks a revocation mechanism and VOMS attributes can only be rendered invalid after their validity period. The VPMan [63] is a project that combines VOMS and PERMIS to improve security of resources such as the NGS.

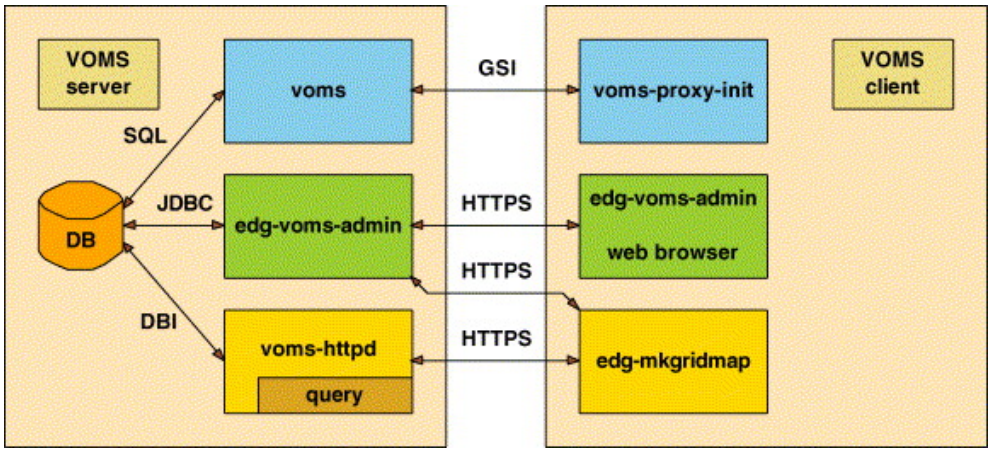


Figure 2.4: The VOMS system [62]

2.3.3 Community Authorisation Service

The Community Authorisation Service (CAS) is an authorisation service used to control access to resources in a VO [64]. CAS provides central control over resources in a given VO, similar to an administrator that manages resources in a domain. It is also analogous to a certificate authority (CA) in a PKI. In essence organisations delegate some privileges to the VO and the VO grants these privileges to its users.

CAS uses a push based authorisation model. As shown in Figure 2.5, a client makes a Security Assertion Markup Language (SAML) request (AuthorisationDecisionQuery) to the CAS server for access to particular resources together with their intended actions. The request is made up of resources and required actions. The CAS server identifies the user and determines the privileges of the user based on the VO policy. The CAS server sends a signed SAML assertion response (AuthorisationDecisionStatement) back to the client. The communicated assertion contains the user's identity, known as the Subject and some or all of the user's requested actions. The user communicates with the resources with the signed assertions and the resource validates the assertion with its local policy and access is allowed but restricted to the validated actions.

CAS only implements PDP engine. The resource providers are required to implement their own PEP engines. Resource providers trust the CAS server to authorise users on its resources. CAS does not use roles or groups thus the management of users' access rights is neither maintainable nor scalable. That is CAS does not issue ACs [58], instead it suggests the permissions a user has as it relates to the resource. Similarly, it is the CAS administrator

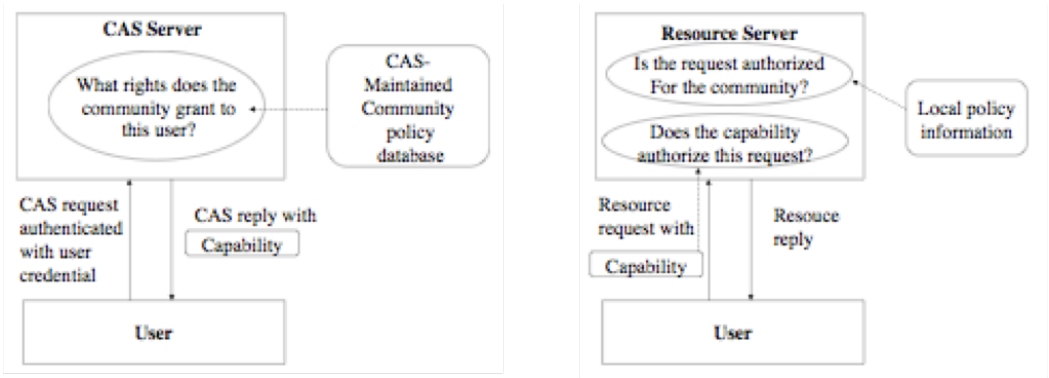


Figure 2.5: The CAS system [64]

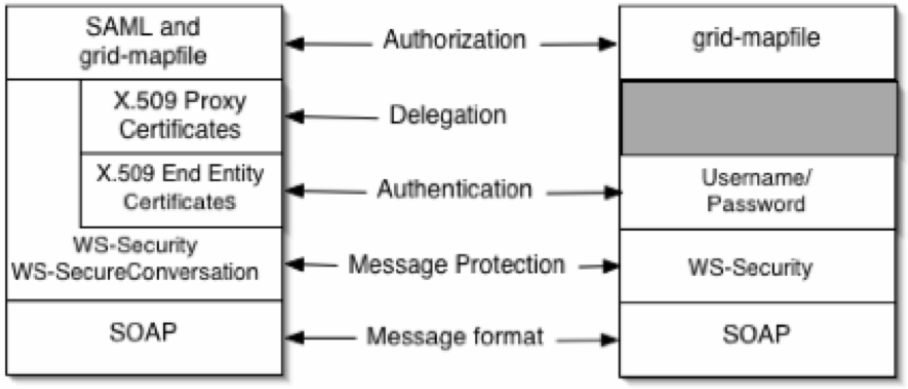


Figure 2.6: GT4 GSI Overview [37]

rather than the SoA, that decides who has access to what resources. CAS thus presents a centralised approach to authorisation which is not really suitable for the Grid. Currently, CAS only supports Globus [59] GridFTP.

2.3.4 Grid Security Infrastructure

The Grid Security Infrastructure (GSI) [37] is the security framework underpinning the Globus Toolkit middleware [59] and provides various security functionality to support Grids. The current incarnation of the Globus middleware is based on Web Services technologies. GSI is the most commonly used security infrastructure on the Grid. It contains four primary functions: authentication, authorisation, delegation and message protection.

As shown in Figure 2.6 Authentication is performed using X.509 digital certificates or username/password pairs as credentials. X.509 credentials allow for the use of advanced security features such as delegation, confidentiality and data integrity. The username and password option does not support such features. However, through other solutions such as MyProxy [65], username and passwords can be used to access MyProxy repositories to create short lived proxy (X.509) certificates.

GSI implements an Access Control List (ACL) in the form of a grid-*mapfile*, which is used to control access to resources. A grid-*mapfile* contains a list of mappings between local user accounts and Distinguished Names (DN) of pre-accepted X.509 entity certificates. The list determines who can access services or resources on the host/container and the privileges of the host's mapped local accounts. This controls access on a per container or per Unix account privileges and not on a per service or per service-method basis. In some cases this kind of access control is sufficient, but is not in many other cases such as the service-oriented cases. The current version of GSI uses the SAML standard for providing AuthorisationDecision assertions as a means of exchanging user's attributes from clients to services, this is used by most CAS clients. It also uses the AuthorisationDecision protocol of SAML to support integration of other authorisation decision services, such as PERMIS.

GSI provides message protection by communicating SOAP messages [66] over Transport-Level Security (TLS) or encrypting portions of SOAP messages using WS-Security standard or the WS-SecureConversation specification for Message-Level Security (see Section 2.4.4). In contrast to TLS which uses X.509 credentials to establish secure transport layer connections, MLS uses X.509 credentials to either establish a session key or uses the associated keys of the sender and receiver's X.509 credentials for message protection.

Whilst widely accepted, GSI has several issues including a lack of scalability owing to the use of grid-*mapfiles* and lack of inherent fine-grained access control. However, fine-grained access control could be achieved with the use of third party authorisation services such as PERMIS in combination with GSI.

2.4 Security Standards in Federation Systems

This section reviews some evolving security standards that underlines the exchange of security information in decentralised open environments. Key to this is federation. Organisation for the Advancement of Structured Information Standards (OASIS) considers Federation

[67] as a term for organisations that collaborate based on agreed standards and that combine business and technological practices to enable access to resources and services across boundaries in a secure and trustworthy way. A federation is built based on trust, standards and agreements [68, 69]. Today, a federation is often associated with single sign-on (SSO) across organisational boundaries [70, 71]. SSO enables a user to authenticate once at the beginning of a process or computation and then to be automatically authenticated to every other process or computation that the user or process initiates. Users do not have to keep multiple passwords and they do not have to keep authenticating themselves every time access is required to other processes, services or resources.

2.4.1 Security Assertion Markup Language

Security Assertion Markup Language (SAML) [72] is a standard developed by OASIS to provide a means of exchanging security-related information between parties over the network. The standard provides a communication framework written in XML for the exchange of assertions between (virtual) domains. This framework does not define new mechanisms for authentication or authorisation but enables existing security mechanisms to interoperate across boundaries. One of the SAML components called Assertions provides security information consisting of user authentication (identity), attributes and entitlements. SAML allows domains to collaborate and make decisions based on signed assertions. As shown in Figure 2.7, SAML components include Protocols [70], Bindings [73] and Profiles [74]. The XML representation of SAML makes it *in principle* interoperable and easy to integrate into existing applications; flexible and extensible with other standards such as Shibboleth, WS-Security [75] and PERMIS.

The SAML specification defines how to construct, exchange, consume, interpret and extend security assertions for various needs. Key benefits of the specification include: interoperability; loose coupling of resources; improved end user experience; risk transference, and a reduced administrative costs for service providers [76].

SAML contemporaries include Liberty Alliance [77] and WS-Federation [69]. To bridge the gap between these different federated protocols, federation gateways [78] are being suggested to act as brokers between the different evolving federated identity management protocols. The word “evolving” is to indicate the incompatibilities that exist presently between protocol versions.

SAML V1.0 became an OASIS standard in November 2002 and V1.1 [79] was released in

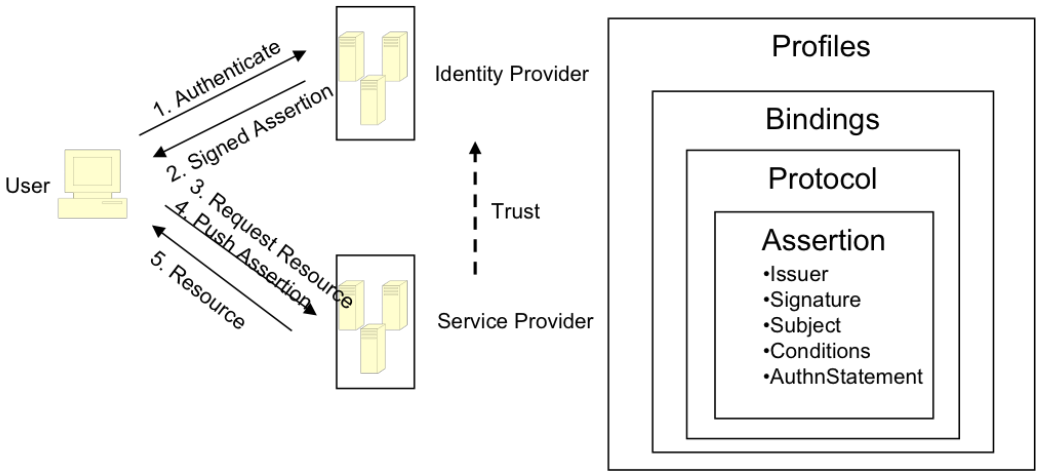


Figure 2.7: SAML Overview - Push Scenario

September 2003. SAML 1.0 addressed how identity information and be communicated from one domain to another. While SAML 1.1 adds support for "network identity", secure exchange of user security information between organisations and introduces signing of SAML assertions by the use of digital certificates. In March 2005, SAML V2.0 [70] was approved as a standard and appears to be a step closer to full convergence of federated identity standards. SAML V2.0 comprises input from Shibboleth and Liberty Alliance's Identity Federation Framework (ID-FF) 1.2 [71]. SAML 2.0 consolidates protocols for single sign-on, policy management and delegated administration. SAML 2.0 enhancements include support for federated identity, global sign-out and session management. Typical uses of SAML includes [80]:

- Web-based single sign-on (Authentication) including communication of authentication assertions from one site to another;
- Attribute-based authorisation and use in Shibboleth for attribute exchange;
- Web Services security as used within SOAP messages to transfer security information.

2.4.2 Shibboleth

Shibboleth [41] is an Internet2 project that describes an open standard protocol for secure exchange of attributes between trusted collaborating partners for the purpose of authentication and authorisation. Shibboleth was initially a higher education initiative for actualisation of federation in schools, but is now gaining acceptance outside academic communities. Shibboleth builds on trust between collaborating partners and between institutions within a federation.

Shibboleth defines a model and architecture for cross-institutional exchange of authentication and authorisation information. Similarly, it defines a means for establishing and managing trust between institutions. It provides a protocol for the secure exchange of attributes required for authorisation decisions between partnering sites. Shibboleth depends upon and indeed pushes the need for standard schema definitions required for secure inter-site communication. It also provides privacy mechanisms that allow users to retain control over the release of their attributes. The model is usable with most third party security standards. As shown in Figure 2.8, the main Shibboleth components include: Identity Providers (IdP) also referred to as origins; an identity provider discovery service also known as Where Are You From (WAYF) service, Shibboleth Handle Indexical Reference Establisher (SHIRE), Shibboleth Attribute Requester (SHAR) and Service Providers (SP) also known as targets.

Shibboleth extends and bridges local authentication and authorisation infrastructures between institutions. Typically, a user that wishes to access a remote resource is redirected back to their home institution for authentication (via a WAYF) established for the federation and uses the home site authentication infrastructure to authenticate themselves. Using a push model, the home institution forwards user attributes to the service provider (in the future, users will be able to select which attributes they want to make available to service providers). The target in turn makes authorisation decisions based on the received attributes using its own authorisation infrastructure. The key to this model is that the semantics and structure of the exchanged attributes are agreed and understood between sites. It also relies on trust between the two sites, that is, where the target site delegates authentication to the home institution.

The main benefits of Shibboleth to the Grid community include simplicity for end users, secure exchange of attributes and single sign-on. Users log in once and their session information allows access to many resources without re-authentication assuming the browser session is maintained. Today many projects [81] that use portals as a Shibboleth target have X.509 certificates created via a portlet for MyProxy [82] services and are used to invoke

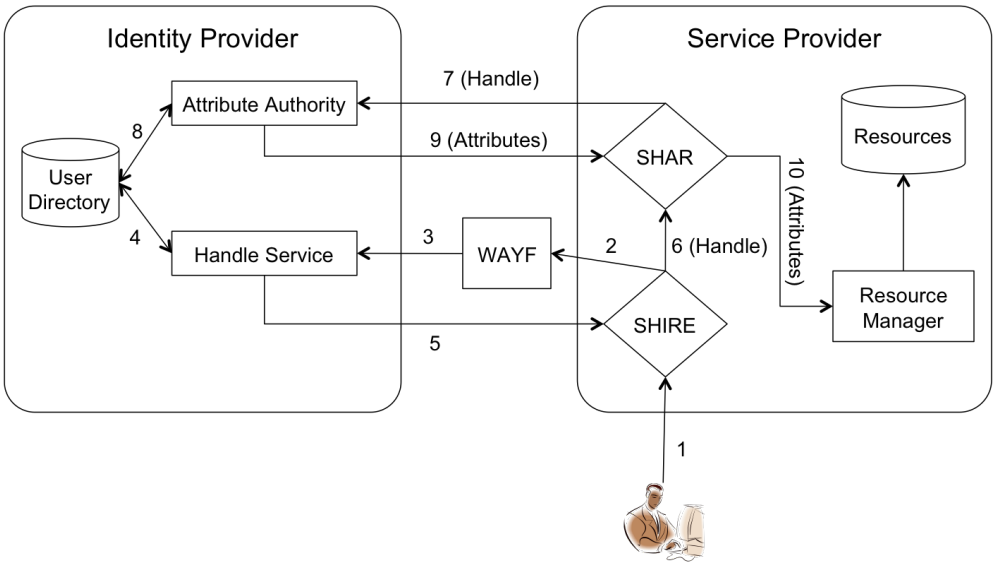


Figure 2.8: Shibboleth Architecture Overview

Grid services on behalf of a user. This is primarily because many users are not willing to maintain and use X.509 certificates due to the associated difficulty in maintaining and using them [83]. Through Shibboleth a user is shielded from X.509 certificate management issues, though (except via MyProxy) it implies that more than one user may be using one X.509 certificate at any given time to invoke Grid services. Although the many user to one certificate approach is widely discouraged, it is often used as an alternative in environments where users are reluctant to managing their own certificates [81].

2.4.3 eXtensible Access Control Markup Language

eXtensible Access Control Markup Language (XACML) [84] is a privilege management language. It is an OASIS [85] standard that can be used to design access control policies and make resource requests. The standard describes both an access control policy language, e.g. who can do what and when, and a request/response language which expresses queries/responses for access to a resource. The language is extensible and can be adapted to suit various purposes. Some extension points include new functions to express new algorithms, data types and logic combinations. The new XACML specification v2.0, can use other open standards for specification of its requests and responses. For instance, XACML generic responses include *Permit, Deny, Indeterminate* or *Not applicable*. It can also be

extended to express responses such as SAML [72] assertions or XACML-based attributes as SAML attributes and requests as SAML requests [80].

XACML, unlike PERMIS, does not provide a policy engine but devolves the task to other actors. In XACMl, the SoA or AA are responsible for the PDP while the Resource Provider (RP) is responsible for the PEP. As opposed to PERMIS, XACML provides a language for defining, assigning and presenting policies as well as attributes that are used by both the PDP and PEP. XACML also provides the specification for locating policies that are required in evaluating requests. In a typical scenario, the PEP receives a request from users and pushes the request and attributes to the PDP. The PDP receives the request and queries for policies that are required for deciding on the request. The PDP, like every other PMI, responds to the PEP with a decision token, which can be *allow* or *deny*. Unlike every other PDP, the XACML based PDP is extensible to fit into various scenarios with various decision algorithms that can be applied to these scenarios. XACML also allows various combining logic (algorithm for evaluating more than one policies) to be applied to the policies or policySets (set of policies).

At the moment XACML does not support features for delegation of authority [86] but does support multiple roles and role based policies albeit with little support for role hierarchies [87]. Although XACML is more versatile and less mature than PERMIS, it has a wide appeal because of its international consensus and the support of OASIS.

2.4.4 Web Services - Security

Web-Services Security (WS-Security) [75, 88] is an OASIS specification that ensures the integrity and confidentiality of SOAP [66] messages. SOAP messages are based on the remote procedure call (RPC) [89] paradigm similar to other distributed architectures like CORBA [90] and DCOM [91]. The WS-Security specification proposes SOAP header extensions, which can be used to underpin many other WS-Security standards. The goal is to provide a complete security framework that is customisable and extensible to individual/specific domain needs. One primary benefit of the standard is that it helps to structure the content of a SOAP header, where a header is a placeholder for contents generated by supported WS-Security standards.

WS-Security uses different security token profiles to carry identity information. One of these profiles is the SAML token profile which uses SAML assertions to provide message security [80]. Other profiles use other token formats such as X.509 public key certificates, Kerberos tickets and encrypted username/password pairs. The specification also supports multiple

signature and encryption formats.

As shown in Figure 2.9, there are numerous evolving Web Services specifications some of which are designed to address a range of security issues. These specifications include: XML-Encryption [92], XML-signature [93], WS-SecureConversation [94], WS-Policy [95] and WS-Trust [96]. WS-SecureConversation is a message level security equivalent of https. Like https, WS-SecureConversion defines a protocol for agreeing a shared session key that can be used for securing a conversation. XML-Encyption is an established technology that uses well-known cryptography technologies to encrypt XML messages. As part of the specification, the identity of the encryption algorithm used is carried in the message. Ws-Policy is used to specify access rules that an authorisation engine can use for evaluating security tokens that are carried as part of a message. WS-Trust is a specification that defines methods for issuing, renewing, and validating security tokens. It also defines ways by which participants in a secure message exchange can establish, assess the presence of, and broker trust relationships.

The issue with all these standards is that they increase the possibility of incompatibility between systems as there are so many developments, some evolving, some real and some proposed. WS-Interoperability (WS-I) [97] is an initiative designed to promote interoperability across all Web Services specifications. WS-I issues profiles (WS-I Basic Profile and Basic Security Profile) that provide interoperability guidance for resolving incompatibility issues that may exist between sets of standards as well as their different versions.

2.5 Trust Management

Trust is an important concept that underpins information security. Understanding trust and how trust can be realised or at least transferred affects how authorisation is viewed in centralised and decentralised open environments. In [98] OASIS defines trust as the characteristic that one entity is willing to rely upon a second entity to execute a set of actions and/or to make set of assertions about a set of subjects and/or scopes. Four trust properties were presented in [15].

- The first trust property is that trust is transitive depending on the context. For example, if Bob trusts Alice and refers patient X to Alice for diagnosis, and Alice trusts Jane and refers patient X to Jane for further diagnosis, then Bob may be willing to accept Jane's diagnosis of patient X.
- The second trust property is that trust cannot be shared - that is if Bob trusts Alice

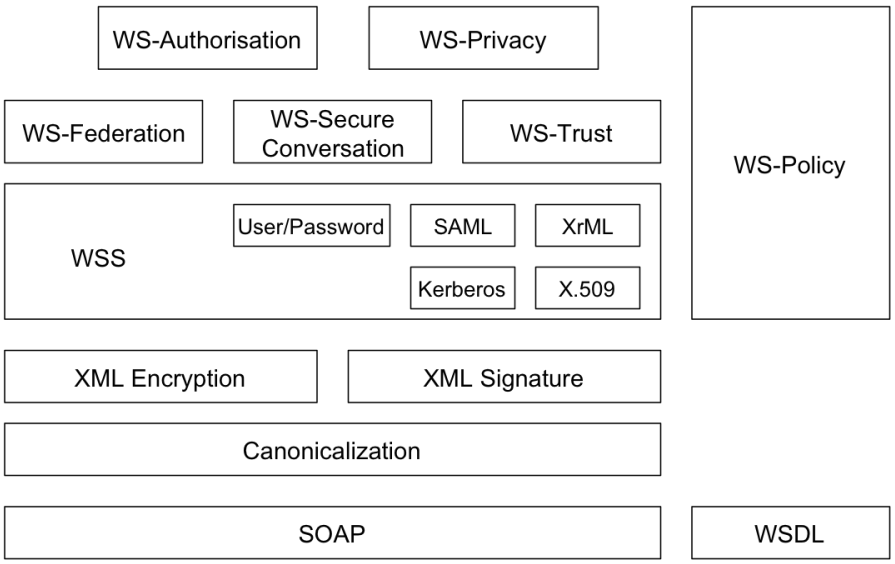


Figure 2.9: WS-Security Evolving Specifications

and Bob trusts John, it does not imply that Alice will trust John.

- The third trust property is that trust is not symmetric, that is Alice trusting Bob does not mean that Bob trusts Alice.
- The last trust property is that trustworthiness cannot be self-declared, that is, Bob saying to Alice that he is trustworthy does not mean that Alice will trust him. This property is the basis for reputation and experience as a means of deciding who to trust [99].

Trust Management (TM), a term first introduced in [100], is an authorisation mechanism that provides a unified approach to how security policies, credentials and their relationships are formulated and interpreted. Trust management from a distributed system points of view is expected to support expressive and extensible policies, local control and decentralised administrative tasks.

2.5.1 PolicyMaker/KeyNote

PolicyMaker [101, 100] was a pioneering example of trust management systems. It was the first tool to process signed requests based on trust management principles [100]. Its credentials and policies were fully programmable and could be written in any programming

language that was "safe" for the local environment. Its design choice was to accommodate any policy language and to enable the reuse of its compliance-checking algorithm. However, in general, the language openness to situations where policy compliance checking was undecidable and in monotonic cases require plenty of time to solve [2]. PolicyMaker was designed to be minimal and analysable, so a fair amount of responsibility was placed on the calling application including the responsibility for credential gathering and cryptographic verification of credentials signatures.

KeyNote [102, 103] was similar to PolicyMaker in that its design was based on the same principles as PolicyMaker. However, it differs in two ways. The first was that KeyNote assigned more responsibility to the trust management engine in that cryptographic signature verification was done by the KeyNote policy engine unlike by the application in PolicyMaker. The second difference was that a specific language compatible with KeyNote's compliance checker was required for writing credentials and policies. In both PolicyMaker and Keynote, credentials and policies were referred to as assertions. Policy assertion differs from credential assertion in that the issuer field of the policy are locally trusted and so does not need a signature.

In KeyNote, authorisation requests are passed, by calling applications, to a KeyNote policy engine for authorisation. The calling application passes the public key of the requester, a list of credentials (signed by foreign parties), local policies and an action attribute to the policy engine. The policy engine replies with an authorisation decision that, in the simplest case, may be "permit" or "deny" the requests. Keynote does not enforce policies; it only provides decisions based on the security requirements of the request to calling application. It is the responsibility of the calling application to interpret and enforce the decision reached by the KeyNote policy engine.

2.5.2 SPKI/SDSI

The Simple Public Key Infrastructure / Simple Distributed Security Infrastructure (SPKI / SDSI) is an IETF SPKI working group project [104, 105], developed to provide an internet standard for certificate formats and authorisation protocols including key acquisition for authorisation purposes as opposed to authentication. Originally SDSI [106] and SPKI were separate infrastructures developed concurrently. They were motivated by the perception that X.509 public-key infrastructure was restrictive and too complex. This perception was emphasised by inadequate global name spaces of public-key certificates. Significantly, SDSI dealt with the issue of decentralised name spaces while SPKI focused on authorisation

specifications. These infrastructures were later merged into a single framework and widely called SPKI/SDSI or just SPKI.

In [104] an algorithm for resolving certificate chains given a set of credentials was presented. The algorithm in the worst case performs at a polynomial-time, which was a good achievement considering the challenges of linking certificates together. This achievement can be attributed to the SPKI/SDSI naming scheme. In SPKI/SDSI, a local name space can be associated with a public-key thus enabling the use of local names to work both locally and globally. Similarly, SPKI/SDSI introduced the notion of linked local names, which expands the expressive power of SPKI/SDSI. This has a wider application appeal since application name spaces can be supported.

In addition to its naming scheme, SPKI/SDSI defines two classes of certificates, name certificates and authorisation certificates. Authorisation certificates differ from name certificates in that they specify permissions granted to an entity. To access a resource, an entity must provide a set of certificates. If a certificate chain exists, access to the resource would be permitted.

SPKI/SDSI and KeyNote are similar in that they are both capability-style TM systems. These systems use credentials to delegate permissions in where each credential is used to delegate permissions from its issuer to its subject (holder). A chain of credentials is like a capability, granting a set of permissions to the holder of the last credential in the chain.

2.5.3 Role-based Trust-Management Framework

Role-based Trust-management Framework (RT) [107, 108] is a language based framework for role-based trust management. In RT, roles are viewed as attributes. An entity is a member of a role if the entity has the unique attribute identified by the role. [107] argued that authorisation in collaborative environments can be simplified in systems where attributes are used for access control decisions. Attribute-based systems provide the flexibility needed to support the distributed nature of authority in decentralised environments - unlike capability-style systems, which cannot express authorisation statements that simplify decision making for access control in a decentralised environment where distributed authorities exist. The four attribute based access control requirements supported in RT_0 [107] are:

- Decentralised attributes in which an entity asserts an attribute that is held by another entity.

- Delegation of attribute authority in which an entity delegates the authority it has on an attribute to another entity.
- Attribute inference in which an attribute can be inferred through another attribute.
- Attribute-based (in contrast to identity-based) delegation of an attribute authority in which authority over an attribute is delegated to another entity based on attributes that the entity possess.

[107] defined four additional RT components: RT_1, RT_2, RT^T, and RT^D. RT relies on the use of credentials to manage roles, role delegation, linked role and parameterised roles. Parameterised roles were introduced in RT_1 to extend RT_0. RT_2 extends RT_1 by grouping logically related objects such as resources together. Threshold and separation-of-duty policies was supported by RT^T, which introduces the concept of manifold roles. Similarly, the support for delegation of role activation was introduced by RT^D.

RT languages use Datalog [109] to represent credentials and policy rules. Using Datalog, the notion of well-formed credentials was introduced, by which safe Datalog rules are used to represent RT_1 credentials. The problem of credential chain discovery is not uncommon to trust management systems. This problem was addressed first in [108], in which goal-directed algorithms were presented for credential chain discovery. These algorithms can be used in scenarios where credentials are either stored with the subject or with the issuer.

2.5.4 Trust Negotiation

Credentials provide a means of transferring trust between entities. The need to exchange credentials between unknown entities introduces the concept of trust negotiation (TN) commonly known as Automated Trust Negotiation (ATN) [110]. ATN is an approach for trust establishment between strangers through the exchange of sensitive information such as digital credentials. Trust is established through an iterative but cautious bilateral disclosure of credentials [111, 112]. Digital credentials, which are analogous to paper credentials are digital assertions about a credential owner signed by the credential issuer. Currently digital credentials are widely implemented using X.509 certificates [30]. A credential is signed using an issuer's private key and the signed credential is verified with the issuer's public key. A credential contains attributes that describe properties of the holder asserted by the issuer. Credentials also contain the public key of the credential owner through which the owner can demonstrate its ownership by the corresponding private key. Negotiating these sensitive credentials without any human intervention is the basis of trust establishment [113, 112].

Several approaches [114, 115, 116, 113, 117] to trust negotiations have been proposed to support access control policies in an open decentralised environment. Some approaches have investigated a trust negotiation framework in the context of a peer-to-peer environment. [117] introduced a Locally Trusted Third Party (LTTP) which acts like a cache and mediator between two entities for the purpose of successful trust negotiation. Similarly [114] introduces a *sequence prediction module* that caches and manages trust sequences used in previous successful trust negotiations. [116] proposes a trust chain based negotiation strategy (TRANS), which dynamically constructs trust relationships using a *trust proxy* that can cache common credentials or partial trust chain information from previous negotiations.

ATN is not all about credential disclosure but also about access policy disclosure. [118, 113, 119, 112] all present models for negotiation strategies to protect the disclosure of sensitive credentials. However, for a negotiation to succeed the negotiating peers must operate using the same strategies. [120] discusses the use of interoperable strategies for credential exchange and why every entity should be free to use whatever strategy they choose before or during negotiation. Two strategies are said to interoperate if trust negotiation succeeds whenever it is possible. Arguably, if a trust negotiation succeeds, access policies would have been disclosed. In some context, these access policies are sensitive information that needs to be protected.

Various ATN systems have been developed, they include Trust-X [114] and TrustBuilder [121]. Trust-X is a framework that provides an XML-based language that is used to encode policies and certificates for trust negotiations. It also provides a peer-to-peer architecture used for negotiation management. TrustBuilder is an architecture that focuses on negotiation strategies. The architecture verifies credentials and checks policy compliance – by verifying which credential should be involve in a negotiation. Other systems like Traust [122] have been developed to augment TrustBuilder to provide interaction between applications or systems that offer trust negotiation services.

The focus of this thesis is how credentials can be negotiated as the basis to supporting collaborative research between autonomous, decentralised domains. In sensitive domains such as e-Health, it is often impossible to deal with strangers owing to the risk involved. This makes it much more difficult to support automated trust negotiations. When an intermediary party is introduced that is known to both parties (initially strangers), then the associated risks are reduced since credentials are not perceived to be disclosed to strangers. The model introduced in this thesis negotiates credentials between known parties who can then act as mediators on behalf of strangers.

2.6 The Semantic Web

The Semantic Web introduces concepts extending the current Web in which information is structured and given well-defined meaning. The semantic Web applies semantics to the Web, expressing content in a machine processable form to enable co-operation between computers and people [123]. The Semantic Web's goal is to provide automatic discovery, integration and reuse of information on the Web by linking information based on meaning and not solely on syntax. The Semantic Web introduces the notion of ontologies as a possible solution to semantic interoperability as discussed in [124].

An ontology is a detailed description of a shared vocabulary or conceptualisation of a specific subject matter. Its use for security promises an inter-institution authorisation method for collaboration between organisations sharing resources across boundaries. In [5] a security-focused interoperable model was presented that uses a shared ontology for implementing access control between collaborating healthcare institutions. The model contains a mediator shown in Figure 2.10, which acts as a security broker that maps shared security ontology information to a local security ontology, which is understood by the local authorisation infrastructure for access decisions.

[125, 6] described ontology mapping as a possible solution to inter-institution authorisation issues. The solution uses matching techniques such as multi-strategy learning and relaxation labelling to match one ontology to the other. Relaxation labelling is a technique use for assigning labels to nodes in a graph, given a set of constraints. The technique considers the labels of neighboring nodes, the percentage of nodes in the neighborhood that satisfies a certain criterion, and whether a set of constraints is satisfied or not. For instance, in Figure 2.11, a Medical Science Division of a University participating in a clinical trial might have roles of Senior Lecturer, Reader and Professor which might have equivalent functionality (and be mapped) to Consultant or Clinical Lecturer to Registrar in the NHS domain. This labelling is not simply a naming issue: the role is directly associated with privileges. Thus a Professor is able to perform duties beyond those of a Senior House Officer for example. The solution in [125] has advantages over [5] in that the latter depends on an agreed and shared ontology while the former does not. However, having an agreed and shared ontology eliminates the need for matching techniques because static rules are defined by each organisation to map between the shared and local ontology.

The application of Semantic Web technologies in Grid computing has led to the concept of Semantic Grid technologies. The Semantic Grid presents an opportunity for information and services to have well-defined meaning, so one can build intelligent and secure Grid services.

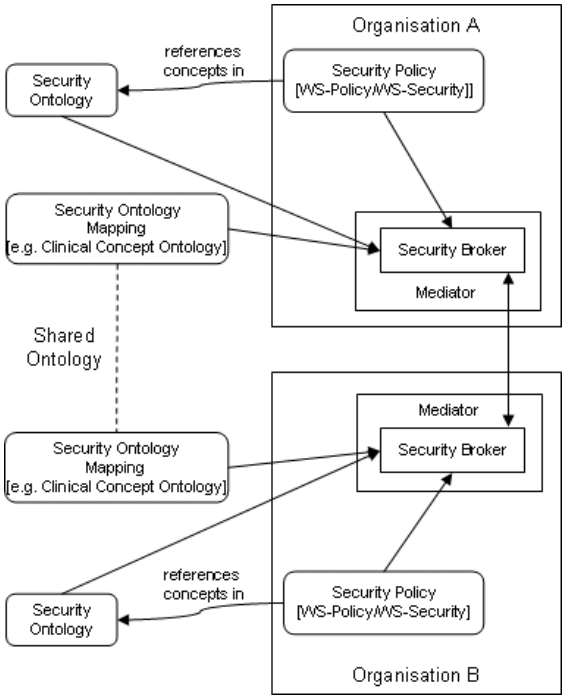


Figure 2.10: Shared Ontology Architecture

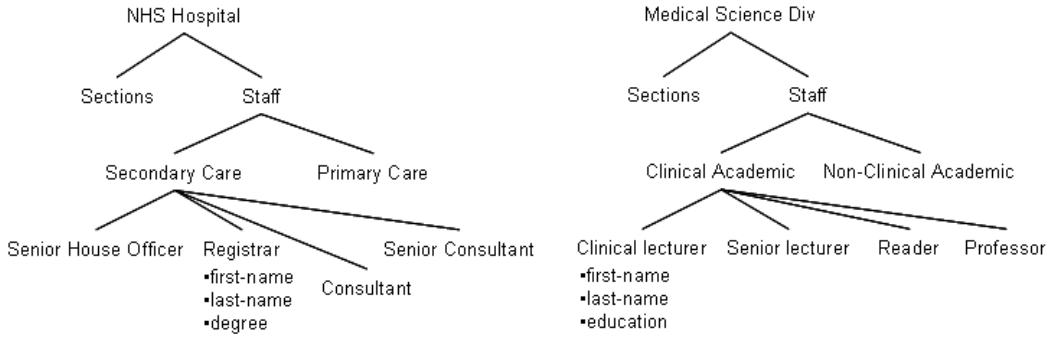


Figure 2.11: NHS Hospital/Medical Science Division Ontologies

Hopefully, the Semantic Grid for data or service integration will spawn semantic security solutions. Especially in scenarios where non-trusting entities and providers wish to share data.

2.7 Context of e-Health Domain

This section reviews some healthcare projects that were developed to provide system interoperability, data access, integration and management in decentralised autonomous e-Health environments. There are many projects that could be reviewed in this domain space, those reviewed here were selected because of their similarities to the MRC-funded VOTES project [126, 127], i.e. they are Grid-based e-Health scenarios.

2.7.1 Integration Broker for Heterogeneous Information Sources

The Integration Broker for Heterogeneous Information Sources (IBHIS) [128] is a collaborative project developed by three Universities in the UK: Durham, Keele and UMIST[1]. IBHIS is an information broker that serves as a trusted intermediary between autonomous organisations to provide access to distributed data services in heterogeneous forms. The architecture provides live access to distributed data that are protected by local access rules. IBHIS was designed based on the concept of federated database systems [129, 130].

As shown in Figure 2.12, the architecture provides services that address issues of query formulation, data access service (DAS) discovery and enforcement of access rules. Each DAS maintains a semantic description file that describes the data it contains; the access control policy for the data it provides and the access control mappings that can be used to relate local roles to remote roles. Queries are formulated using an ontology service that maps clinical concepts between a broker and data services. In addition to the broker, the IBHIS model includes an audit service that keeps records of all actions, which may subsequently be used for evidence corroboration in ethical and legal cases. Similarly, the model includes a semantic registry, used to identify the required DAS when a query is submitted.

A user creates a query using a global ontology and, based on their access profile, submits the query to the broker. The broker processes the query consulting the semantic registry to identify which DASs are needed for the query and to decompose the query into sub-queries. Based on the identified DAS and sub-queries, attribute authorisation is performed to match

[1] UMIST is now known as the University of Manchester

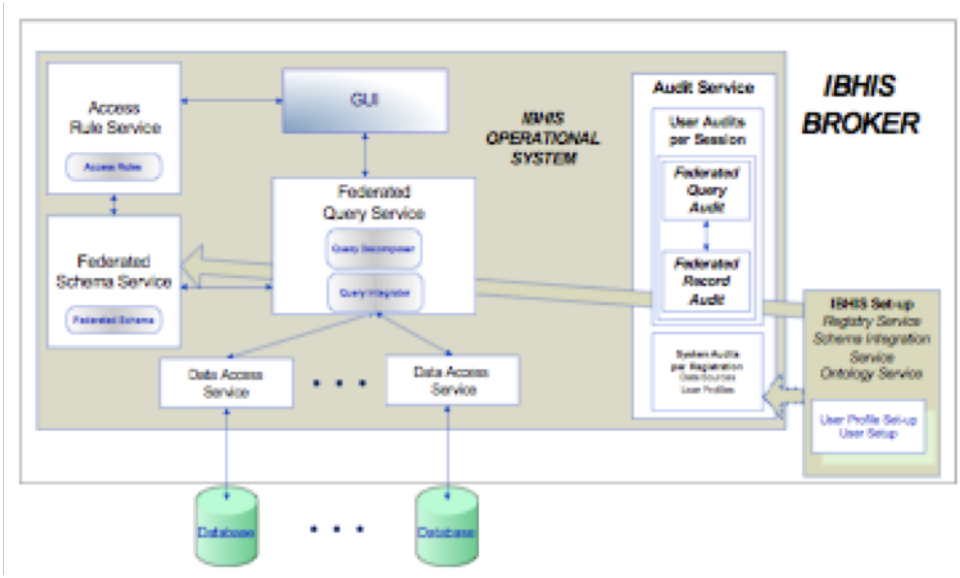


Figure 2.12: The Architecture of IBHIS Operational System [128]

the sub-query to the data access permissions received from the DAS. The broker views data fields as attributes for the purpose of attribute authorisation and in some cases attributes are dropped from the final sub-queries. DAS executes sub-queries; does authorisation and returns results to the broker.

In the final stage, the broker is expected to filter results through inference rules, but it is not clear how these rules are formed or the algorithms used for the inference checking. Also, besides semantically correct global query issues, the centralised nature of the model where the broker acts as a link between users and data providers, is the single point of failure of the entire system. To address this potential failure, it is important for each DAS to be able to discover and connect to one another besides the centralised broker. However, for this to work, each DAS must be able to trust the another, i.e. realise trust, and they must be able to map local roles between one another. Trust realisation and relating local roles across domains are some of the issues this thesis aims to address.

2.7.2 Clinical e-Science Framework

The Clinical e-Science Framework (CLEF) [131, 132] is a UK e-Science project that is being developed by a host of UK universities to deliver an interoperable infrastructure that

supports access and integration of various operational electronic patient records in order to support clinical and bioscience research. One of the goals of the project is to develop a manageable repository of information histories that can easily be linked to genetic and genomic information.

In the framework, the repository often called a 'Chronicle' is the central component of the framework. All data workflows pass through the repository, thereby making it possible to build an historical information view of consenting patients. In order to preserve privacy and confidentiality various safeguards are designed in the framework. First, patients records comprising of structured and unstructured data sets are pseudonymised in the repository to anonymise patients. Patient's records are tagged with a CLEF entry identifier which can only be reversed by the data provider. Second, data is depersonalised during access to conceal or remove any potentially identifying information. Third, statistical disclosure control techniques are employed to eliminate or reduce any identity inference that may occur.

Another design goal of CLEF was for it to be built on or extend Grid middleware and so leverage Grid security frameworks. In addition to Grid authentication, CLEF access control services are being developed with PERMIS [56] to provide scalable authorisation. In retrospect, CLEF depends on numerous technologies, especially technologies that are proven for e-Science to deliver and maintain its interoperable repository. However, it is not clear how CLEF hopes to achieve robustness and availability of its 'centralised' repository.

CLEF's aim is to "develop a high quality, safe and interoperable information repository, derived from operational electronic patient records to enable ethical and user-friendly access to the information in support of clinical care and biomedical research" [131]. However, to achieve this in the long term, the security model will have to be extensible and scalable. The centralised model will have to be reviewed and the issue of trust realisation between collaborating members will have to be addressed. This thesis focuses especially on trust realisation from a clinical research and collaboration perspective.

2.7.3 ARTEMIS

A typical healthcare provider has many heterogeneous healthcare information systems that support patient care delivery. Providing interoperability between these healthcare systems was the motivation behind ARTEMIS. ARTEMIS [133, 134] is a project involved in the development of a semantic web-service driven peer-to-peer infrastructure for healthcare information systems. ARTEMIS fundamental objective is to support communication of electronic

healthcare records across organisational boundaries. With this objective an infrastructure that supports interoperability between healthcare information systems was developed.

The key components of the ARTEMIS infrastructure includes: clinical concept ontologies; semantically annotated security and privacy policies, and mediators, Figure 2.13. Clinical concept ontologies are derived from healthcare information standards such as HL7 [135], CEN TC251 [136], ISO TC215 [137] and are used to describe Web Services functionality semantics and the data or documents that are exchanged through Web Services. Semantically annotated security and privacy policies are defined by healthcare providers. These policies are based on agreed organisational requirements. Mediators are super-peers that broker and reconciles semantic differences between healthcare organisations. Mediators reference agreed healthcare standards to reconcile organisational semantic differences. The peer-to-peer architecture of the mediator components provides scalability and enables discovery of other mediators.

ARTEMIS models healthcare organisations as peer nodes, which are connected in a peer-to-peer network structure. Typically healthcare providers are autonomous. Each healthcare provider develops and semantically annotates security policies that suit their organisation. Using mediators or super-peers, there security policies are able to be mapped and translated between organisations, Figure 2.14. A mediation [5] is achieved between a requester and a provider by linking organisational role ontologies with clinical concept ontologies. The idea of role mapping using clinical concept ontologies introduced a possible solution to information access interoperability. However, it introduces semantic interoperability issues that are common with open network and federated systems [3, 138]. To address this, this thesis investigates an alternative to the single ontology paradigm by considering how localised security ontologies can be folded into one another through trust contracts.

2.7.4 PsyGrid

PsyGrid [140, 141] is a UK e-Science project developed by a collaboration of universities in the UK to address issues that affect healthcare systems, such as data gathering and aggregation. Its aim is to develop Grid middleware and applications that could support epidemiology and clinical trials in the mental health domain. It also aims to make available resources on first episode psychosis that researchers and clinical scientist can use. Essentially, it is a data collection system for longitudinal studies.

Data is collected from eight different geographic areas in the UK and stored in a data repository. Each of the areas is autonomous and corresponds to a hub of the Mental Health

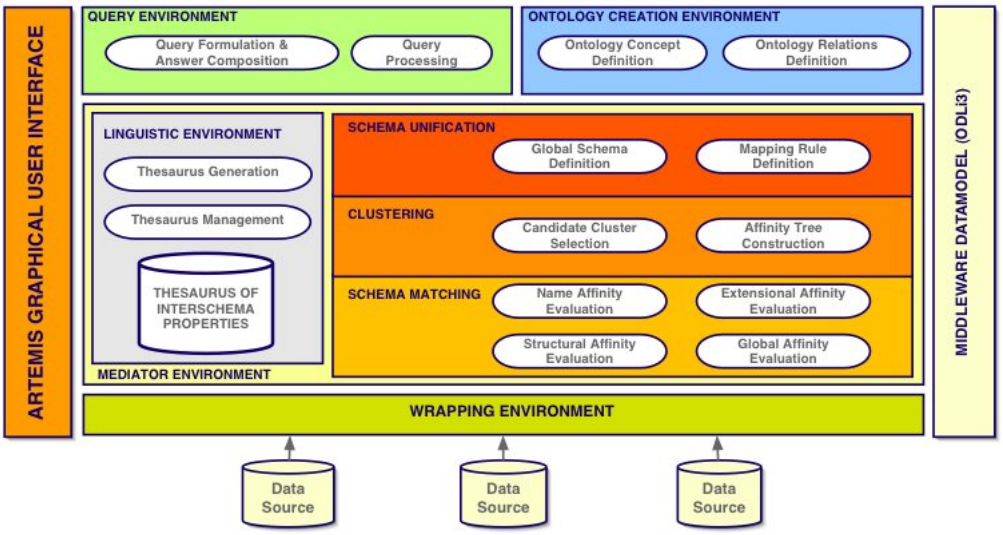


Figure 2.13: The ARTEMIS Architecture [139]

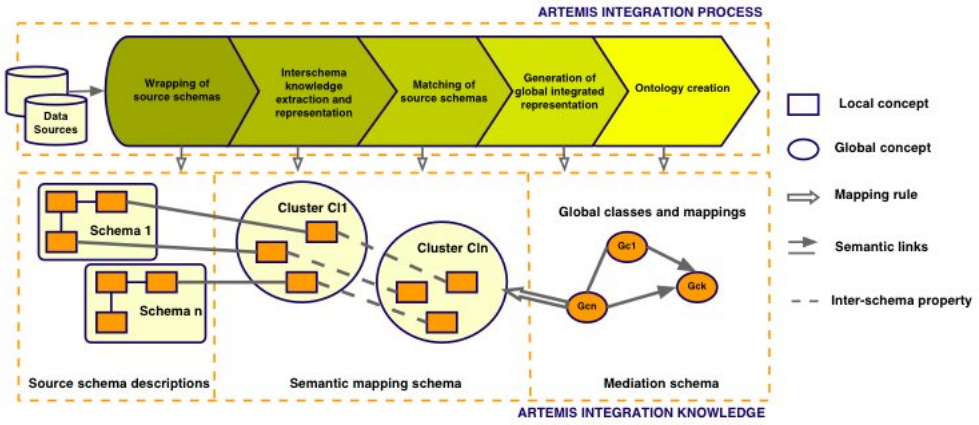


Figure 2.14: The ARTEMIS Process [139]

Research Network (MHRN). One of the design principles of PsyGrid is to produce a reliable, customisable and secure infrastructure that clinical researchers can use. Two key components of PsyGrid are a central data repository for storing clinical data and a security sub-system that uses SAML [70] and a role-based access control system. As opposed to message level security, TLS [17] in conjuction with PKI is used for secure communication. Since the framework is not designed to support message collaboration, where parties need to secure different parts of a message. PsyGrid also provides audit trails and anonymisation support on data imported and exported.

For the implementation of the security layer, technologies like SAML, LDAP, MyProxy [65] and OGSA-DQP [142] are used for the inter-organisation federation. Policy authorities in an organisation trust SAML assertions signed by another organisation's attribute authority. A policy authority is expected to be able to verify the signature on a SAML assertion of the attribute authority it trusts, thereby enabling privileges of a user from another organisation to be used for access control. This is made possible in PsyGrid as inter-organisation federation that requires participating organisations to agree on semantics and meaning of privileges that will be used in the federation in advance. This assumption is not always valid and means a lack of scalability and flexibility. Also, the requirement of reaching agreement in advance is a limitation since agreements are not always easy to reach and maintained in an open environment. This thesis examines pair-wise agreements in the form of trust contracts since organisations are able to reach and maintain agreements with another on a peer-to-peer basis.

2.7.5 Cancer Biomedical Informatics Grid

The cancer Biomedical Informatics Grid (caBIG) [143] is a program and designed to create a national-scale infrastructure that connects data, research tools, scientists and organisations, to leverage their combined strengths and expertise in an open federated environment, so as to precipitate in the development of effective patient therapies for cancer. The underlying infrastructure for caBIG is referred to as caGrid. caGrid provides the technology that enables collaboration between institutions and it supports sharing of information and analytical resources securely. caGrid makes available virtualised datasets through its open standardised service interfaces and communication mechanisms. caBig emphasises data modelling and semantic interoperability across heterogeneous resources, achieved by use of controlled vocabularies, common data elements, published information models and well defined programming interfaces.

caGrid infrastructure is built upon the version 4.0 of Globus Toolkit (GT4) [59]. This enables caGrid services to be based on the Web Services Resource Framework (WSRF) version 1.2 of the standard [144]. Developed on top of the Grid Security Infrastructure (GSI) [37] is the Grid Authentication and Authorisation with Reliable Distributed Services (GAARDS) [145]. GAARDS provides enterprise services and administrative tools that can be used to administer and enforce security policy in a Grid environment.

GAARDS services for authentication include: Dorian Services, Authentication Services, Credential Delegation Services (CDS) and Grid Trust Services (GTS). Like GSI, GAARDS require Grid credentials in order for a user/application to communicate with secure Grid services. To obtain Grid credentials, a user is expected to be registered with a Dorian service or have an existing account in another domain that is a Dorian trusted security domain (identity provider). An Authentication Service provides a framework for identity providers for issuing SAML assertions that may be used by a user to obtain Grid credentials from a Dorian Service. However, in the case where a user is registered at a Dorian domain (IdP), the user can obtain their Grid credentials directly by authenticating at the Dorian domain. Once a Grid credential has been obtained, a user can then invoke secure services. A secure service supports authentication by validating Grid credentials presented by a user against a Grid Trust Service (GTS).

GTS maintains a federated trust fabric of all trusted digital signers such as Dorian and grid certificate authorities. The importance of GTS is in how it maintains the trusted digital signers as these signers are dynamic, growing in number, and regularly publish new trust information. The importance of GTS is that it enables Grid services to remain updated and to know which CA certificates or credentials to trust. The traditional approach to trusted CA certificates is to have a CA directory in the server/service configuration directory that is used to identify which credentials are trusted (LSLV). This implies that trust is configured at the container level instead of the service level. LSLV approach is supported out of the box in Globus, but its scalability and maintainability drawbacks make it unsuitable for enterprise Grids that require fine-grained and flexible access control for each Grid service that exists in a container.

In addition to GTS managing trusted certificates, Grid services can retrieve trusted certificate from GTS and perform credential validation locally (RRLV) or they could send received credentials to a GTS for validation (RSRV) [7].

Once a user has been authenticated at a GTS, a secure service may then proceed to determine if the user is authorised to perform what they requested. GAARDS services for

authorisation include: Grid Grouper services [145] for managing and enforcing Access Control Policies. The Grid Grouper service is a group-based authorisation service that can be used to enforce authorisation policy based on memberships to groups[2]. Groups provisioned by the Grid Grouper are defined and managed at the Grid/virtual organisation level. This enables applications and Grid services to enforce authorisation policies based on membership to groups at the Grid level. Groups in Grid Grouper are organised into namespaces, which are also called stems and a stem may contain a set of child stems.

In addition to the Grid Grouper, the caCore Common Security Module (CSM) [146] can also be used for authorisation as it provides a centralised approach to the management and enforcement of access control policy. A Grid service can enforce its local authorisation policies by asking the Grid Grouper directly if the requester (user) is a member of a given group. Similarly, a Gird service can make use of the CSM as its PDP to decide if a requester is authorised to perform a given action. The CSM in the case of group-based access policy, can enforce access control based on groups that are provisioned by the Grid Grouper.

In order to support federated queries, distributed workflow execution and integration with web applications, GAARDS 1.2 provides Credential Delegation Service (CDS) and Web Single Sign-On (WebSSO). Using the CDS, external users (delegates) or services can assume the identity of another entity (delegator) for the purpose of authorisation with remote services or applications. With the CDS, a delegation policy can be expressed, which entitles delegates to assume the identity of the delegator for a limited period. WebSSO on the other hand compliments the CDS in that it enables web users to make use of both web and Grid services without the need to provide login credentials. WebSSO provides a mechanism that automates the delegation and retrieval of a user's Grid credential by using the identity of the currently logged in web user.

Although the concept of Grid Grouper and CDS is significant, it is not clear how the model will scale with increase in number of Grid Groupers. Similarly, as many more users become available, many more delegators will be required. To address this scalability concern, it would be valuable if components like Grid Grouper and CSM could negotiate trust, i.e. group namespaces (security attributes) with one another. In addressing this, this thesis investigates trust negotiation in decentralised environments.

[2]The similarities between groups and roles are discussed in 2.2.5

2.7.6 UK BioBank

UK Biobank [147] is a long term study that began in 2007 in the UK to investigate genetic predisposition, environmental exposures including lifestyle, nutrition and medication in the causes of diseases. The study targets half a million UK based participants aged 40–69, which will be followed up for about 25 years from their enrolment. During the study, data such as blood samples, DNA, disease events, drug prescriptions, deaths of participants are stored in a centralised repository [148]. Currently, participants are being recruited and data collected from about 39 regional centres in the UK. It is expected that some years from now, researchers will apply to use the database and be able to compare a sample anonymous participants that developed a particular disease such as heart disease with a sample of participants that did not. With the sample, they will hopefully be able to consider the benefits, effects and interaction of specific genes, lifestyles and medications.

The main components of the biobank system architecture include: appointment service, data archive, a laboratory information management system (LIMS) and a core data repository [149]. The appointment service which is controlled centrally, makes use of an NHS register of people aged 40–69 to identify potential participants. In order to recruit $500,000$ participants, an estimated 5 million primary invitations are needed using the NHS register. Collected data is stored in a data archive. Data samples like urine and blood samples are stored in a sample archive, which is managed by the LIMS. Access to the sample store are through store operators based on access privileges assigned to user profiles. In addition, orders for sample retrieval can only be made through the LIMS which is subject to an approval process and cannot be initiated by a store operative. The core data repository which is the architecture central component, can only be accessed by a selected number of named UK biobank staff under controlled conditions.

UK BioBank is relevant to this work as it presents an example of a large, country-wide study, with serious implications if anything goes wrong. It is important to note that the UK BioBank approach to security is a closed and centralised approach so as to limit and curtail security risks. For example, the aspect of patient recruitment required the National Health Service to provide the register of patients aged 40–69. This was made possible by the fact that one of the sponsors of the study is UK Department of Health and thus has the advantage of NHS services [149]. However, if patient records were not obtained centrally, that is if each of the 39 regions were expected to separately find and recruit patients from their locality, it would present a huge security challenge and several risk implications could follow. Similarly, it is yet to be seen how access to ongoing participants records and data linking would be

controlled and monitored over the coming years. Perhaps the NHS would still be responsible for providing follow up data as they become available. To address these issues, projects like VOTES and the Scottish Health Information Platform (SHIP) [150] are investigating various security models that BioBank can build on.

2.8 Summary

This chapter has reviewed the associated background literature in the area of security and presented other relevant work associated with supporting e-Health collaborations.

The first section of this chapter covered security in decentralised and open environments by introducing security concepts including authentication, authorisation, data confidentiality and integrity, and non-repudiation. The discussion showed that trust underlines security concepts and that security threats are related to the violations of these concepts. From an authorisation point of view, trust is often modelled using digital credentials. Related to this, cryptography and public key infrastructure were discussed as they highlight the issues of trust realisation. The discussion showed that using PKI, confidentiality can be achieved through encryption, data integrity through a digital signature and centralised trust through certificate authorities.

The second section of this chapter reviewed the various types of access control models since they underpin the work described in this thesis. The models included mandatory and discretional access control; access matrix; identity-based and role-based access controls. These models provide a means of analysing whether a security system conforms to an abstract model in its design and implementation, and whether it satisfies the same properties as the abstract model. The discussion showed that no one security model fits all by highlighting their relevant strengths and weaknesses.

The third section of this chapter focused on various Grid-oriented privilege management infrastructures including PERMIS, VOMS and CAS. Using these infrastructures as examples, the challenges of security in decentralised and open environments were discussed. The discussion showed that the definition, assignment, presentation, delegation and revocation of security attributes are core security challenges in decentralised environments. This section concluded with a review of Grid security infrastructures and associated framework, and emphasised the need for fine-grained access control systems.

The fourth section of this chapter discussed some of the evolving security standards and

languages, such as SAML, Shibboleth, XACML and WS-Security. The discussion showed some of the in-roads made towards solving key security challenges. The chapter described how solutions such as SAML provides a mean of exchanging security-related information between parties, whilst Shibboleth provides a standard protocol for the secure exchange of attributes between trusted collaborating partners for the purpose of authentication and authorisation. XACML defines a standard for writing access control policies and provides one way to describe the semantics for trust realisation. WS-Security on the other hand provides a set of specifications for realising security concepts through Web Services. To achieve this, the WS-security standard has proposed SOAP header extensions and profiles for exchange of security tokens.

The fifth section of this chapter introduced trust management. Four fundamental trust properties were discussed: that trust is transitive; that trust cannot be shared; that trust cannot be mirrored, and that trust cannot be self-declared. All these properties underline the notion of trust in this thesis. They serve as the basis for the credential equivalence discussed in Section 5.6. Some existing trust management models were presented in this section including: PolicyMaker, SPKI/SDSI, RT, and ATN. The limitations of these models highlighted the need for a trust discovery and realisation framework, especially between non-trusting domains. The sixth section reviewed the area of the semantic web (and the notion of ontology) and its application to Grid systems.

The last section of this chapter addressed the context of this research by focusing in particular on the e-Health domain and on the various projects that have been conducted in respect of interoperability, data access, data integration and management in decentralised autonomous e-Health environments. IBHIS was reviewed and it was described how it uses a broker for data sharing and access. However, for IBHIS to work in a decentralised environment it must be able to realise trust and map local attributes between domain brokers. The CLEF system was reviewed and it was shown how it uses a centralised repository for sharing data between parties. Given this model, it was identified that it is not clear how CLEF hopes to achieve robustness and availability of its centralised repository. ARTEMIS, another e-health framework was described and it was shown how it relies on the use of clinical ontologies and semantically annotated security policies. However, ARTEMIS also introduces semantic interoperability issues in decentralised environments where for example different ontologies may exist. Other e-Health projects reviewed in this chapter included PsyGrid, caBIG and UK Biobank. For each of these it was discussed how they each have their own open security challenges and could benefit from trust realisation and discovery.

3 DTN Objectives and Design Overview

This chapter presents Dynamic Trust Negotiation (DTN). It describes the overall objectives of DTN and the design of the DTN system. The design of DTN includes the formulation of policy statements, the application of policy language, the DTN protocol, and the architecture. The design is broken down into components and this chapter describes each component in relation to other components. It concludes with a typical scenario for applying DTN.

3.1 Design Objectives

Trust is an important component in collaborations and this is especially so in e-Health collaborations. As collaboration in the e-Health research community continues to grow, so does the need to realise and establish trust between systems or agents representing the involved parties. Dynamic Trust Negotiation (DTN) [10, 151] is designed to address the issue of realising and establishing trust between systems or agents within the e-Health domain such as the clinical trials. To address this issue, the DTN design objectives are as follows:

- **To support access to resources across organisational boundaries:** the primary goal of DTN is to enable seamless access to resources across boundaries without compromising separate organisation security measures that are in place, e.g. to ensure confidentiality and integrity.
- **To be compatible with existing privilege management infrastructures:** In order not to compromise the existing security infrastructure of each organisation, DTN has to be deployable and compatible with the existing PMI in place. These might include security standards such as WS-Security [88], Shibboleth [41], SAML [72] and XACML [84].
- **To provide an alternative to the single global policy or attributes approach that defines what attributes or credentials are, what they are used for and**

where they are used: Existing solutions for decentralised access control like Shibboleth require the use of agreed attributes such as the eduPerson attributes [152, 153] to be used for authorisation across boundaries. Another solution is that used in Artemis [134] where local ontologies of authorisation attributes are mapped to a global attribute ontology or vice versa for access control. In contrast to the global attributes approach, DTN should enable the use of foreign attributes or credentials to be acceptable, useable and tenable for local policies without the need for global attributes.

- **To negotiate security credentials through delegation of roles:** DTN should allow the negotiation of attributes used for access control between organisations through the use of credentials so that local roles can be assigned to entities with credentials issued by trusted foreign parties.
- **To support the discovery and establishment of chains of trust:** Chains of trust are analogous to certificate chains, which show how a remote certificate can be trusted through an order list of certificates through intermediary certificates to a root trusted CA certificate. In this thesis, chains of trust are viewed as an order list of entities that are connected together through trust relationships. DTN can be considered as supporting the discovery of these trust relationships and when necessary, establish the chain of trust between entities.

In addition to the above objectives, the following are some e-Health collaboration challenges that DTN aims to address.

- **Control And Autonomy:** Each organisation in the community is independent and controls access to, and use of, their own resources. Tackling this demands the need to address the challenges of distributed access control including policy definition, enforcement, conflicts and heterogeneity. In e-Health environments, organisations typically design and enforce their own access policies with no notion of external collaboration or sharing information with external organisations. In some cases, situations arise where access policies are tied to applications, making it difficult to share information with other applications within the same organisation, a unique feature of health organisations.
- **Credential proliferation:** The use of credentials for access control has been around for a considerable time including applications in health organisations. However, a common occurrence in most health organisations is the proliferation of credentials. Some of these credentials are tied to applications and so it is possible to have users with similar credentials for different applications. Applications such as the Scottish

Care Information Store (SCI-Store) [154], the General Practice Administration System for Scotland (GPASS) [155], and the Picture Archiving and Communications System (PACS) [156], for example, provide access to data resources that in most cases are related to one another, thus users are required to present different credentials to different applications for access to related data.

- **Different and numerous policies:** Since a centralised access management control model does not exist, organisations are at liberty to create policies that meet their specific needs. These policies, more often than not, are written in different languages. Similarly, as numerous health applications exist within the same organisation, different policies exist with the same goals.
- **Policy disclosure - not with strangers:** This is one of the main challenges of e-Health collaborations. Policies can exist that may not be disclosed to external organisations. This is partly due to government directives such as [157] that prevent data disclosure. In some scenarios where policies disclosure is permitted, they are only disclosed to trusted third parties that possess the necessary access and disclosure agreements (contracts).
- **Government guidelines:** The e-Health environment is an area where government guidelines, acts [158], and directives [157] exist to ensure adequate protection for patients and healthcare providers. They raise requirements that must be met and systems developed in this area are required to satisfy these requirements, e.g. privacy protection.

3.2 Architectural Overview

The overview of the DTN layered architecture is shown in Figure 3.1. The architecture is made up of three main components: a protocol interface; a trust enforcement engine and an access control engine. Two protocols are defined in DTN, the first is for trust-path discovery, through which a trust-pathway (chain of trust) can be established between entities. The second protocol deals with trust negotiation, through which trust (credentials) can be negotiated between trusted entities on behalf of two non-trusting entities. Without discovery of a trust-path, trust negotiation on behalf of non-trusting entities cannot be achieved. These protocols are discussed in more detail in Chapter 7. In addition to negotiating security credentials, discovery and the establishment of trust-pathways, these protocols also ensure that policies are not disclosed to strangers (non-trusted entities).

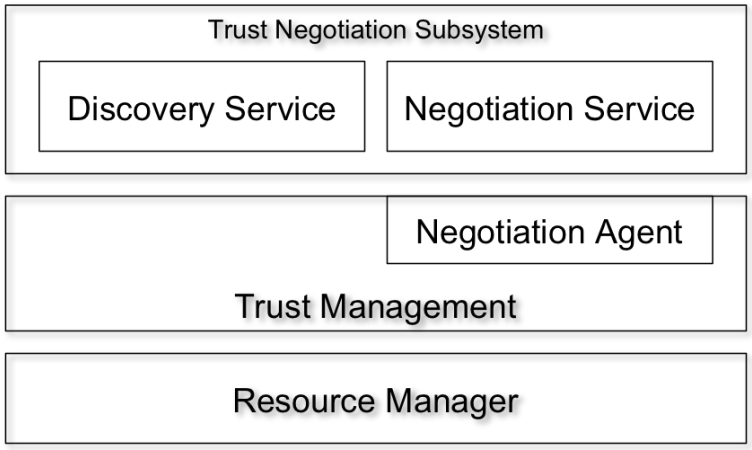


Figure 3.1: Decentralised View of DTN Architecture

As shown in Figure 3.2 The trust enforcement engine is responsible for trust negotiation between itself (e.g. the local organisation) and other trusted entities (e.g. remote organisations). It participates in negotiations that are initiated by other intermediary nodes (e.g. remote/intermediate organisations) and can also initiate negotiations with other intermediary nodes. It receives foreign credentials and, where the negotiation is successful, it releases credentials of its own to other entities that it trusts, on behalf of the initiator. During negotiation, when a credential is received, it verifies and also validates the credential against its policies through the access control engine.

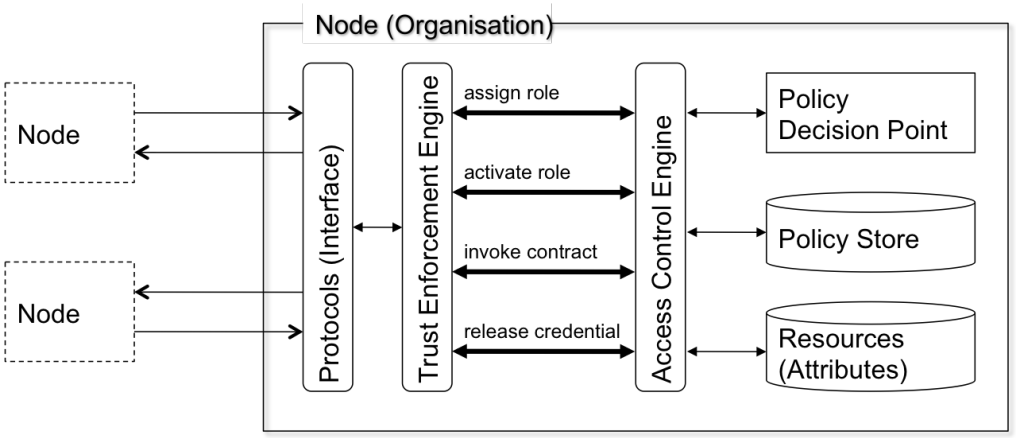


Figure 3.2: DTN Negotiation Components

The access control engine itself provides authorisation services for the trust enforcement engine. As credentials are negotiated, the acceptance and release of credentials have to be authorised against various policies. Similarly, in the case of it being the resource target, i.e. the service provider, foreign credentials that are to be negotiated for local attributes, such as roles, are authorised against trust contract policies. These policies are discussed in detail in Chapter 6. Some authorisation requests that are made to the access control engine are discussed in subsequent sections.

3.3 Protocol and Trust Enforcement Engine

The trust enforcement engine works with the negotiation protocol. When a negotiation request is received, the protocol parses the request and, if it is identified as the service provider, it forwards the received credentials to the trust enforcement engine. If it is only an intermediary node in the negotiation, it forwards the received credentials along with the list of *next-hop* nodes to the trust enforcement engine. The list of *next-hop* nodes are generated and maintained through a discovery protocol on an on-demand basis. The list of *next-hop* nodes is made up of nodes that are in its *circle-of-trust*. Credentials and a list of next-hops are used by the trust enforcement engine to construct authorisation requests that are made to the access control engine.

The trust enforcement engine ensures that policy rules governing the acceptance and release of credentials are enforced. Often these rules embody constraints that are placed on certain entities and their credentials. For instance, in Figure 3.3, node D might be willing to release its General Practitioner (GP) role at the request of organisation C to organisation A and not to organisation B because B has no trust contract for C's GP role, unlike A that has one. A *trust contract* is an agreeable arrangement made between two mutually suspicious entities to trust each other, which includes agreement on identity and key management, credential mappings and delegation of access attributes such as roles. These agreements can be static and made offline by Attribute Authorities. The list of nodes that a node has a trust contract with constitute the node's *circle-of-trust*. The *Circle-of-trust* and *trust contract* are discussed in more detail in Chapter 5.

The government guidelines, acts and directives mentioned above can be enforced using trust contracts. Similarly, the trust enforcement engine ensures that negotiations do not occur with entities that are not in its COT thus ensuring that policies are not disclosed to "strangers". In addition, since trust negotiation goes through multiple negotiation hops, it makes it less

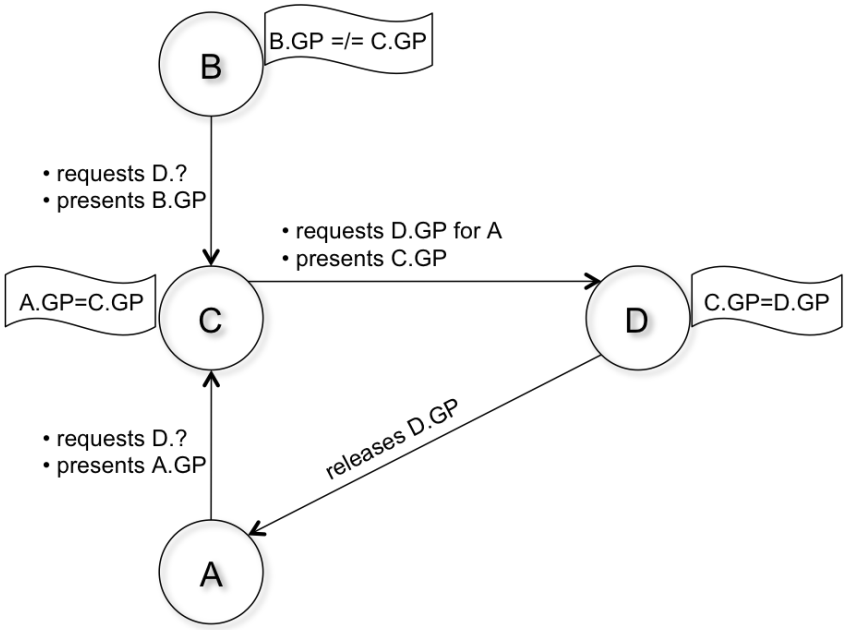


Figure 3.3: Use of Trust contract

likely for the requester to infer the policies that control access to a resource. The ability to infer is far less likely as multiple trust-paths may exist between the requester and the service provider.

3.4 Trust Enforcement Engine and Access Control Engine

During negotiations, the trust enforcement engine makes various authorisation requests to the access control engine depending on whether its domain is the service provider for the service request or just a trusted intermediary.

Example 3.1 *Bob in organisation A initiates a service or data request to Organisation C for patient records. C is not in A's* circle-of-trust *as no trust contract exists between them. However, B is in A's* circle-of-trust *and C is in B's* circle-of-trust*. As shown in Figure 3.4, A negotiates with organisation B for B's credential that can satisfy C's access policy. B in this case is a trusted intermediary while C is the service provider. Each of these organisations A, B, C have their own trust enforcement engine and all participate in the trust negotiation.*

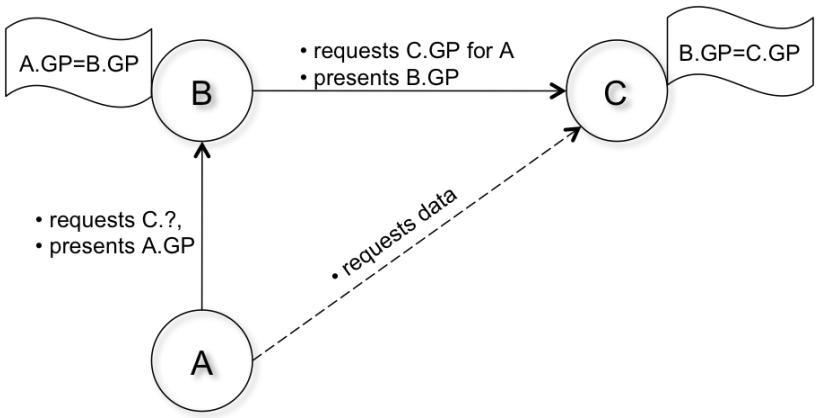


Figure 3.4: Use of Trust contract and Circle-of-trust

Upon receiving an authorisation request, the access control engine invokes its policy decision point (PDP) that in turn makes access deductions using authorisation policies to reach a decision. An authorisation decision is essentially a logic deduction based on the request and policy rules. Four essential authorisation requests include: assigning and activating a role, invoking a contract and releasing a credential.

3.4.1 Assigning a Role

In the case of an organisation being the service provider, the trust enforcement engine needs to ascertain if it can assign a role to a foreign entity based on the negotiated credential. A foreign entity can access a resource only after they have been assigned a local role with permissions for the resource requested.

Example 3.2 *Based on Example 3.1, a credential issued by organisation B that asserts that Bob is a* clinician *has to be assigned to C's local role before a decision can be made for Bob to access C's resources that he requested. For instance, C has a resource called PatientMaster table and the Nurse role has been granted permissions to perform select and update actions on PatientMaster. So if Bob wants to select or read datasets from PatientMaster table at C, he would need to be assigned C's Nurse role.*

A typical assign request would be:

> Assign *Bob* who is issued a credential from *B* that asserts *Bob* has a *Clinician* role with the *Nurse* local role granted with permission *p* such as {*select* action

on *Patientmaster* table}. This can be written as a predicate:

$$assign((credential\ issued\ by\ B\ that\ asserts\ entity\ has\ Clinician\ role), Nurse, Patientmaster(select))$$

In Example 3.2 Bob is the entity and only when the assign predicate is evaluated to be true by *C*'s PDP that Bob can be assigned a nurse role. The specification of such predicates are described in Chapter 6.

3.4.2 Activating a Role

A role can be viewed as an attribute that is associated with a set of permissions. In RBAC [43, 44], roles are assigned to users and given permissions through roles. It is important that roles are assigned to users as this allows for the specific privileges they require for their task. Once a user is assigned a role, the user can enjoy the benefits of the role by activating its role membership or by acting in the capacity of the role. In practical terms, a user can activate a role by presenting the attributes that says he/she is a member of a role to a policy enforcement point (PEP).

Role-based access control is largely used today because it specifies and enforces security policies in a way that reflects organisational structures. In ABAC [107, 119], roles are simply attributes that are used to distinguished between capacities [21]. As is shown in [39] role attributes can be exchanged as attribute certificates which are widely used in Grid systems.

Example 3.3 *Bob might be registered at a hospital as both a* Consultant *and as a* Patient*. Bob's attributes showing he is a* Consultant *might include his General Medical Council (GMC) Code and his specialisation attributes. As a* Patient *he might have a patient identification attribute. As a* Consultant *he may be able to administer prescriptions to Patients and as a* Patient *he might himself receive prescriptions. In most systems he would not be allowed to activate both roles at the same time as control policies may be in place to ensure separation of duty [48].*

In DTN, role memberships are described using predicates, *can_activate* and *has_activated* predicates. The *can_activate* predicate describes if an entity can act in the capacity of a role, while *has_activated* describes whether an entity is currently acting in the capacity of a role. A typical activation request may be:

> Has *Bob* activated a *Consultant* role? If the answer can be deduced to be true then *Bob* can administer a prescription. If false, the question "can *Bob* activate a

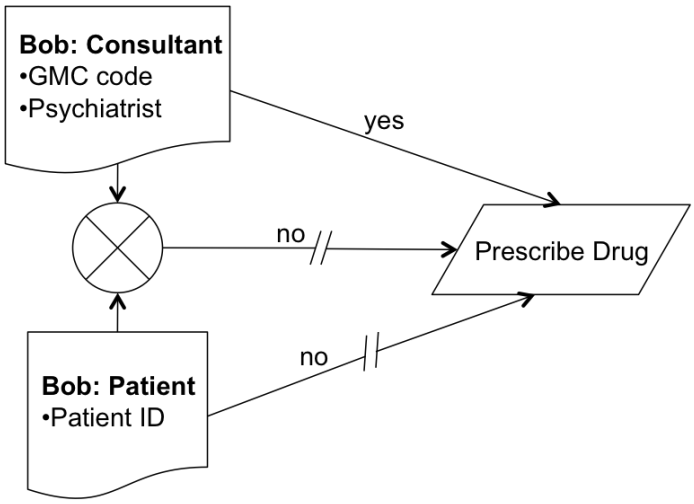


Figure 3.5: Separation of duty

Consultant role" will be asked. If that answer is deduced to be true then *Bob* can, in principle, go ahead to administer a prescription to a patient. The predicates for *has_activated* and *can_activate* can be written as follows respectfully:

$$has\_activated(Bob, Consultant(g19099, cardiologist))$$
$$can\_activate(Bob, Consultant(g19099, cardiologist))$$

where g19099 in both predicates might be Bob's GMC code and cardiologist his specialism.

As another example, if Bob wants to activate the *Consultant* role. If the predicate $has\_activated(Bob, Patient(1943567))$ can be deduced to be true, then the access policy rule that says an entity cannot have activated both a patient and a clinician, e.g *Consultant* role, will be triggered preventing Bob from activating both roles. For instance a policy rule that controls the activation of a *Consultant* role at a given PEP may be expressed as:

Entity e can activate a *Consultant* role if a *Consultant* role at a given PEP has been assigned to e and e has not activated the *Patient* role at that PEP.

The specification of such predicates and policy rules are described in Chapter 6.

3.4.3 Invoking a Contract

At the heart of DTN are trust contracts. Trust contracts are agreements between domains, which are invoked during trust negotiations. A remote domain invokes a contract it has reached with another domain in order to get the benefits of the interoperability. When a credential is received from a remote domain, the first request to the access control engine by the trust enforcement point is a contract invocation request. If an applicable contract does not exist, the trust enforcement engine will prevent the domain from participating in the trust negotiation.

Before a local role can be assigned to a foreign entity, the credential presented by the entity must (through appropriate deductions) be able to invoke a trust contract for that local role. A foreign entity should not be allowed to activate a local role, only a local entity can. Similarly, an entity should not be able to invoke a contract that exists in its local domain since no contract between a local entity and its domain should exist[1]. For instance, based on the example in Section 3.4.1 Bob can be assigned a *Nurse* role only after Organisation B has invoked the contract for the *Nurse* role. In this case, C would have a trust contract with B that says B's *Clinician* credential is required for C's *Nurse* role.

The contract invocation request made by the trust enforcement engine to the access control engine can be expressed as follows:

> Can B invoke a contract for the *Nurse* role by releasing a credential it issued that asserts a *Clinician* role? That is, can the following predicate be deduced?
>
> $$can\_invoke((credential\ issued\ by\ B\ asserts\ entity\ has\ Clinician\ role), Nurse)$$

If the request is not granted then the domain can not take part in the negotiation. If the request is granted and the domain is the service provider, then it is possible to proceed to assigning the role to the entity assuming no issues of separation of duties exist. If the request is granted and the domain is a trusted intermediary, a credential release request will be made and evaluated before it can proceed with the trust negotiation.

A request can also be made to deduce if a contract has already been invoked. This deduction can be used to prevent two mutually exclusive contracts from being invoked concurrently by the same entity. This request can be expressed as follows:

[1] A local entity should not be able to invoke a contract for a local role where both are in the same domain and/or sub-domain. In the context where both are in the same domain but in different sub-domains then a local entity may be required to invoke a contract for a local role.

> Has B invoked a contract for the *Nurse* role in the current negotiation session? This can be expressed as a predicate: $has\_invoked(B, nurse)$

The specification of such predicates and policy rules are described in Section 6.

3.4.4 Releasing a Credential

If a domain is participating in a trust negotiation as a trusted intermediary, the trust enforcement point would first make a contract invocation request and, if the request is granted, it would subsequently make a release credential request to the access control engine before a credential can potentially be released to remote entities during negotiation. This is because, in trust negotiation, credentials are sensitive resources that have to be protected. For instance, based on the example in Section 3.4.1, organisation B's PDP would need to deduce if the credential that asserts a *Clinician* role can be released to C. It is only when the deduction is true that B's trust enforcement engine can release the credential to C during the trust negotiation process.

The release credential request ensures that credentials are only released to approved domains i.e. known and trusted organisations. Suppose organisation B from the example in Section 3.4.1 has organisations D and E in its *circle-of-trust*. B could negotiate, i.e. release its credential to D and not to E. This may, for example, be a scenario where D is a healthcare institution and E is a social care institution, and the credential being negotiated are mainly healthcare related. In some other scenarios, it may just be that C is not comfortable releasing that role credential to E.

DTN supports policies that control the release of credentials, for an extra layer of protection. The release credential request can be expressed as follows:

> Can B release a credential that asserts a clinical role to domain C if entity A has invoked a contract for the clinician role? That is, can the following predicate be deduced from B's policy?
>
> $can\_release\_cred(C, (credential\ issued\ by\ B\ asserts\ entity\ has\ clinician\ role), A)$

If the request is granted, the credential is released to the trusted domain, i.e. C in this case, and negotiation at B would be regarded as successful. If the request was denied, the negotiation requester domain, i.e. A in this case, would be notified with a negotiation failed response. The specification of such predicates and policy rules are described in Section 6.

3.5 Access Control and Policy Evaluation

Access control relies on the evaluation of access policies with respect to access requests. Policy evaluators, such as a PDP, are used by access control engines to deduce if access to resources should be granted or denied. A policy is made up of rules which contain the core logic of a policy. Policy rules can be expressed as Boolean functions or predicates.

Programming logic lends itself well to policy specifications as policies are written in terms of the goals and not in terms of actions sequences. Today many policy languages are based on Datalog [109] and in some cases Datalog with constraints [159, 107]. Constraint logic programming offers an easy way of introducing access control restrictions, e.g. that a GP can only access medical records of patients that are registered with a particular GP practice. The fact that one can express natural conditional statements into policy rules makes constraint logic programming attractive.

The policy language used to express policy rules and credentials in this thesis are based on Datalog with constraints similar to that expressed in Cassandra [2]. For DTN policies, four main predicates are introduced and mentioned in Section 3.4. These predicates are discussed in detail in Chapter 6 along with how these predicates can be used to express policies in a distributed environment.

However, there are several drawbacks to implementing Datalog inference engines [107]. For example, it is not easy to integrate a Datalog inference engine to an application. To avoid these drawbacks, DTN policies are implemented using XACML [84] policies since XACML policy evaluation engines are relatively easy to implement and integrate with existing applications. Appendix A shows some DTN rules that have been expressed in Datalog and implemented as XACML policies.

3.6 Scenarios

To show the applicability of DTN, this section outlines some typical scenarios. These are explored in more detail in Chapter 8.

A trial administrator with the responsibility of recruiting patients for a cardiovascular research trial executes queries that span multiple health organisations. These queries involve linking records across all of the datasets provided by service providers in these organisations. Access to data is based on trust and access rights that each of the service (data) providers

have granted to the administrator based on: their role; their home organisation; the current clinical trial study, the context for access; the origin of the data; the general patient consent or patient consent for a study; and the sensitivity of potential result sets. An example of a successful query might be to return patients suffering from cardiovascular disease, aged between 40 to 50, residing in Glasgow, who have given general consent for their medical records to be used for clinical trial studies. Or it may return patients who have not given any consent that the trial administrator would like to subsequently solicit consent from (possibly through the patient's GP).

Similarly, suppose a researcher from the University of Dundee named Alex wants to recruit patients for a leukaemia cancer trial from three healthcare providers: Glasgow, Edinburgh, and Aberdeen. He decides to submit query requests to each provider. The following idealises the transaction between Alex@Dundee and Gartnavel hospital in Glasgow.

1. **Authentication at Gartnavel:** Alex needs to identify himself at Gartnavel. Since Alex does not have a local ID at Gartnavel, Gartnavel cannot authenticate him locally. However, Gartnavel is part of an Identity Federation which Dundee University is also part of. Based on the basic authentication trust relationship between IdPs in the federation, Gartnavel was able to verify that Alex has authenticated at Dundee University. With this authentication information identity, a request session is created for Alex.

2. **Authorisation at Gartnavel:** Alex submits a query request to Gartnavel for patients that satisfy a certain criteria. Although the identity of Alex has been verified, his researcher role at Dundee University cannot be used for authorisation at Gartnavel since no authorisation policies exist recognising a researcher role from Dundee University, i.e. no access agreement exists between Gartnavel and Dundee University for data access. However, Gartnavel does have access agreements with other healthcare providers. Gartnavel notifies Alex that his query request cannot be authorised but are willing to reconsider his request if his authorisation credentials can be successfully negotiated through other healthcare providers.

3. **Trust Negotiation Initiation:** Since Alex cannot negotiate his credentials directly with healthcare providers, he makes a negotiation initiation request to his Institution's attribute authority asking that his credentials be negotiated with healthcare providers that Dundee University has access agreements with.

4. **Trust Negotiation:** Dundee University negotiates Alex's research role with healthcare providers that they have access agreements with. One of the healthcare providers that Dundee University has access agreements with is Tayside Childrens Hospital, Ninewells.

Tayside Childrens Hospital had agreed to have Dundee's research role recognised as an honorary consultant role in line with their NHS contract policy [160]. Gartnavel has access agreements with Tayside Childrens Hospital, which has Tayside consultant roles recognised as Gartnavel consultant roles in line with [160]. That is, Tayside honorary consultants are recognised as Gartnavel honorary consultants just as Tayside consultants are recognised as Gartnavel consultants.

5. **Data Access:** Based on Alex's request session, Gartnavel is able to collate all negotiated credentials that they have access agreements with including having a Tayside consultant role. Authorisation decisions are made based on Alex's negotiated credentials and requested data. Since the Tayside consultant role is permitted to access anonymous leukaemia patient records at Gartnavel, Alex is allowed to have a list of anonymised consented leukaemia patients that satisfy his criteria.

This is an elementary scenario where only one negotiation hop gives Alex the permissions he needs. In most cases, several hops are needed in order to achieve access agreements between institutions.

The above scenario is also idealised since federated access control and authorisation system are not yet in place at these hospitals. Such scenarios are currently being explored in Connecting for Health (CfP) / National Programme for Information Technology (NPfIT) [161] and will be explored further in the Wellcome Trust funded Scottish Health Information Platform (SHIP) [150] amongst others.

3.7 Summary

In this chapter the objectives of DTN were presented. To meet these objectives and the associated challenges of e-Health collaborations, the overarching DTN design objectives were formulated and discussed. Based on this an architectural overview was presented that included the three main components, namely: a protocol interface; a trust enforcement engine and an access control engine.

The chapter described the function and purpose of the architectural components. Discussion on assigning and activating roles, invoking contracts and releasing credentials were introduced and explained. The evaluation of access policies vis-a-vis policy rules and credential requests used in the research were presented. The advantages and disadvantages of technologies such as Datalog were discussed.

The chapter concluded by describing various scenarios utilising the application of DTN.

4 The DTN Architecture

This chapter describes the DTN architecture. The architecture is made up of two systems: a discovery system and negotiation system. The first section describes the discovery system and the components that make up the system, while the second section describes the negotiation system and its components. The last section describes the flow of data in the DTN architecture during a typical negotiation process.

4.1 Discovery System

The discovery system is used by domains to realise trust-pathways between themselves and other nodes that are not members of their circle-of-trust. As shown in Figure 4.1, the discovery system is made up of five components: the protocol interface; the controller; the protocol data processing; a management interface, and a routing information handling. These components are discussed in detail in the following sections. The routing algorithm described by these components is presented in Chapter 7.

4.1.1 Protocol Interface

The discovery system provides an interface through which routing messages, i.e. route request and responses can be exchanged between a node and nodes in its *COT*. The interface is implemented as a web service and has an endpoint reference [162]. It is this endpoint reference that nodes use to communicate with each other. By exposing the discovery system as a GT4 or Web Service, the discovery system can easily be implemented to extend any trust management infrastructure, such as the PERMIS.

The endpoint reference of a node is created using the URI address of the node. Whenever a node decides to change its URI address, it notifies the nodes in its COT of the change. Route request and route response described in Section 7 are two primary interfaces of the

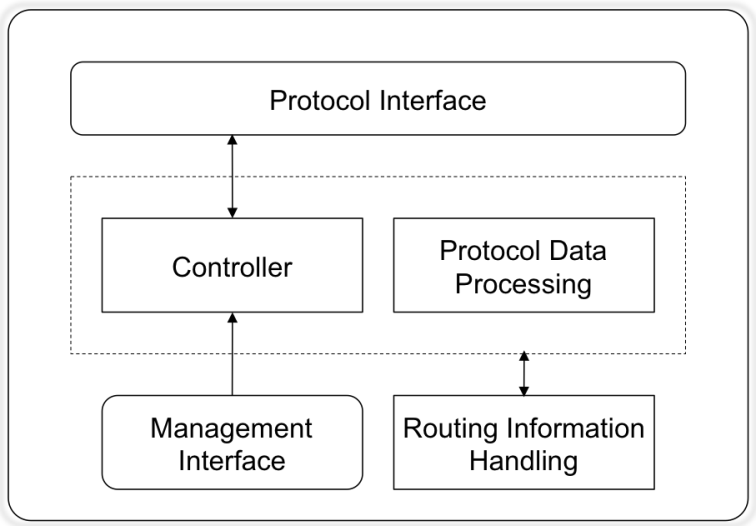


Figure 4.1: Discovery System Overview

discovery web service. Communicating via WS implies that routing messages are exchanged as SOAP messages [66]. Similarly, it implies that nodes can take advantage of WS-Security as discussed in Section 2.4.4 for route message exchange.

The protocol interface is responsible for transcoding routing messages between itself and the controller component. Message transcoding is node dependent as nodes are free to implement the discovery protocol as they see fit, so as to enable easy integration with their local infrastructures.

4.1.2 Controller

The protocol processing unit is made up of two components: a controller and a protocol data processing component. The controller component controls and maintains the state of messages during the discovery process. When a routing message is received, the controller decides how the message should be processed and, when necessary, creates corresponding messages that are communicated via the protocol interface.

A controller authenticates the sender of a message and validates a message before it is processed by the protocol data processing component. Security keys, which are managed by a trust management system, are used by the controller for MAC validation and also for sender authentication through key verification.

The controller monitors and ensures message security. The controller guards against vulnerabilities and attacks such as spoofing and replay attacks, as discussed in Section 7.4.1. The controller along with the protocol data processing component ensures that routing messages received or sent adhere to the discovery protocol and are secure.

4.1.3 Protocol Data Processing

The protocol data processing component processes routing messages. Every message received has to conform to the structure defined for routing messages. Received messages are decoded, interpreted and actions taken based on the message.

After validation by the controller, route request (RREQ) and route response (RREP) messages are processed by the protocol data processing component. The data is analysed and passed on to the routing information handling component for subsequent reuse as described by the discovery algorithm presented in Section 7.2.3.

4.1.4 Routing Information Handling

The routing information handling (RIH) component stores routing information for discovered nodes. Routing information includes next-hop nodes and nodes that exist in a COT. During trust negotiation routing information is communicated via the controller and through the management interface to the negotiation service. Frequently the controller must access the RIH for its routing information and retrieve or update the COT information. The endpoint of a discovery process needs to update the routing or next-hop information.

4.1.5 Management Interface

The management interface is used to access and manage data on the controller. Through the interface, a node's COT information can be updated and managed. The interface can also be used to update the number of trust contracts a node has with other nodes. The interface can compliment any existing trust management infrastructure through the use its API and it can be used by the negotiation service to retrieve next-hop nodes.

As discussed in Section 7.2.2 key management is pivotal to the discovery process. The management interface provides a medium through which security keys managed by a trust management infrastructure are accessible by a controller. A security key can be a shared

key that has been agreed between two nodes or it might be a public-private key pair. It is necessary that key management is separated from the discovery system as existing infrastructure may already exist for managing security keys. An example of a trust management infrastructure for key management is a PKI [26] system – discussed in Section 2.1.3.

4.2 Negotiation System

The negotiation system shown in Figure 4.2 is used for trust negotiation. The system comprises a negotiation service, agent and a SAML *plus* component. The negotiation service interacts with the negotiation agent and is used to enforce the decision of the agent. The negotiation service also interacts with the discovery system in order to identify other trusted domains (next-hops), when it is acting as an intermediary domain. Each of these components are described in this section.

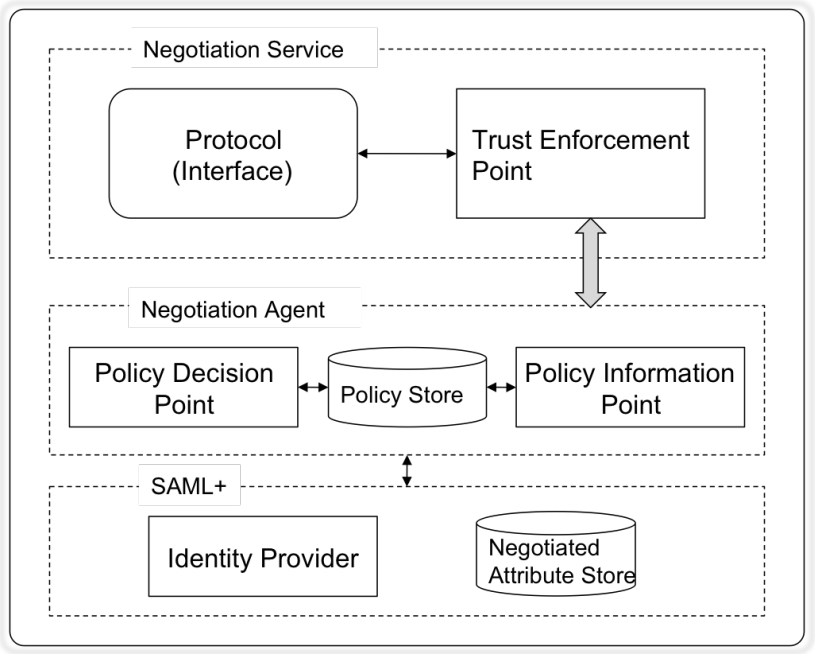


Figure 4.2: Negotiation System Overview

4.2.1 Negotiation Service

The negotiation service is the point of contact for node interactions during trust negotiation. It provides a secure interface through which domains can request and exchange credentials. The negotiation service consists of two main components: the interface to the negotiation protocol, and the trust enforcement point. The protocol interface is the point of interaction where negotiation request or responses are received or sent. The protocol interface is also the point of interaction between the negotiation service and the management interface of the discovery system.

4.2.2 Trust Enforcement Point

This component interacts with the negotiation agent and is used to enforce the decisions of the agent by communicating responses or by interacting with the SAML module. It communicates responses through the negotiation service to a requester or to intermediary domains. Based on the negotiation agent's decision, it initiates a negotiation request with other next-hop domains on behalf of the requester.

4.2.3 Negotiation Agent

An agent must understand the protocol used for trust negotiation as in [122] and manage the negotiation session. An agent validates a negotiation request and checks that access and release policies are not violated. The attribute assertions received are validated against access policies and checked against trust contracts that may exist between domains. Depending on the negotiation strategy in use, further requests can be made for more attribute assertions from the request domain. The agent checks the release policies upon validating the accept policies. If these policies are satisfied, attribute assertions are issued for further negotiation with other intermediary domains or for interaction with the SAML module.

Negotiation requests made by the trust enforcement point are processed by the negotiation agent's access control components. The four main negotiation requests are role assignment, role activation, contract invocation and credential release. These negotiation requests are also viewed as authorisation requests and are described in Section 3.4.

4.2.4 Policy Information and Decision Points

A Policy Information Point (PIP) is a component used by the negotiation agent to prepare the request context that is sent to the policy decision point (PDP). Request context includes attributes for the subject, actions and resources. In DTN, subject attributes include role assertions required for access decisions by foreign entities.

The prepared request context is presented as an XACML request [84]. XACML was preferred as it is a standard widely accepted for access control policies and, as such, has wider acceptability in open systems. Existing PIP components in various organisations can be used to prepare DTN authorisation requests. An example of an XACML DTN authorisation request is shown in Appendix A.

A Policy Decision Point (PDP) is a component used by the negotiation agent to make authorisation decisions. A PDP based on the request context will retrieve corresponding policies that are necessary for it to reach a decision. In DTN, as with the PIP, the PDP use the XACML standard. If corresponding policies cannot be found, an indeterminate result is returned [163]. If corresponding polices are found and are valid, the access decision can either be a deny or permit response.

4.2.5 SAML plus

Component The SAML *plus* is a component invoked when a resource/ domain is the service provider for the negotiation session. Even though intermediary domains have SAML *plus*, they do not make use of it during negotiation unless they are the service provider. It is called SAML *plus* because it extends both the Identity Provider (IdP) and the Service Provider (SP) components used by SAML.

Negotiated Attribute Store

A Negotiated Attribute Store (NAS) is introduced to augment a service provider. This is a requirement for DTN as a NAS is used to store attributes that have been negotiated during trust negotiation. These negotiated attributes are bound to the negotiation session and thus link the resource requester to a local attribute. As opposed to retrieving attribute assertions from an IdP during a SAML request assertion session, attributes stored in the NAS are retrieved instead. This redirection step is one of the SAML extensions for DTN.

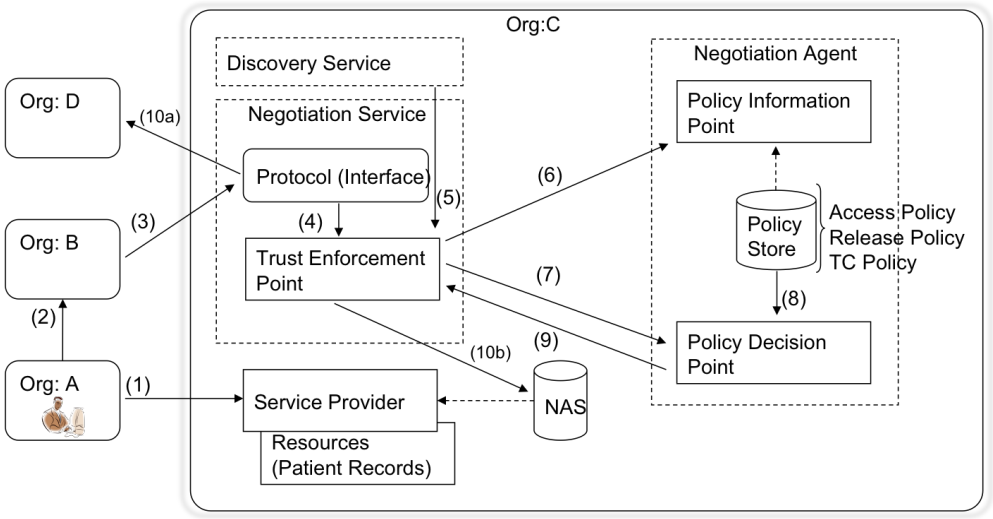
Extended Identity Provider

In SAML 2.0 when a service provider makes an assertion request e.g. *<samlp:AttributeQuery >* to an IdP, the IdP is expected to respond with a SAML assertion *<samlp: Response>* containing a user's attributes [74]. The extension DTN offers is in the attribute response of the IdP. Instead of the IdP responding with a user attributes, it does two things. Firstly it initiates trust negotiations with trusted parties if they have not been already initiated. After a negotiation time-out or on receiving a negotiation response, it prepares a *<samlp:Response>*. Secondly if the negotiation was deduced to be successful from the negotiation response, it responds with a *NAS* attribute value in its *<samlp:Response>*. If the negotiation was not successful or timed out, it responds with the user attributes as it normally would if it was not extended with DTN. When a service provider receives an attribute assertion with *NAS* as an attribute value, it queries its NAS store for negotiated attributes. It is these negotiated attributes that the service provider uses to support authorisation decisions.

4.3 DTN Data Flow

A detailed view of the data flow between DTN components during negotiation is shown in Figure 4.3. The interactions between these components are enumerated as follows:

1. **Resource request:** An entity requests a target resource from a service provider using a request interface e.g. a web browser or portal.
2. **Negotiate credentials:** The IdP of the requester initiates a trust negotiation with other trusted nodes, i.e. next-hop nodes. The IdP negotiates credentials via the negotiation protocol interface.
3. **Negotiate credentials:** If the receiving node is an intermediary node, it does steps 4 to 10a. If the receiving node is the domain of the service provider, it does steps 4 to 10b excluding 10a.
4. **Pass credentials:** The node protocol interface communicates the negotiation request to the node's trust enforcement point (TEP). Remote credentials and the target domain's identification are passed on to the TEP.
5. **Get next-hops:** The TEP queries the discovery service via its management interface for the list of next-hop nodes for the target domain.



1:Resource request; 2:Negotiate credentials; 3:Negotiate credentials; 4:Pass credentials; 5:Get next-hops; 6:Request context; 7:XACML request; 8:Pull policies; 9:XACML response; 10a:Negotiate attributes or 10b:Negotiated attributes

Figure 4.3: The DTN Architecture and Data Flow

6. **Create request context:** The TEP contacts the PIP for the request context. Information necessary for an authorisation request such as foreign credentials, domain of the foreign entity and information about the prospective next-hop nodes are passed on to the PIP. The PIP creates request context based on the information it receives and it has already, and makes it available to the PDP.

7. **XACML request:** The request context is passed on to the negotiation agent for the PDP to decide on.

8. **Pull policies:** The PDP makes authorisation decision based on the request context and corresponding policies. The type of policies include access policies, release policies and trust contract policies.

9. **XACML response:** For each authorisation request, a corresponding XACML response is issued to the TEP.

10. **Negotiate attributes or store negotiated attributes:** If a node is participating in a trust negotiation as an intermediary node and if *permit* was the PDP response, the TEP will negotiate its credentials with authorised next-hop nodes via the protocol interface. If a node is the service provider domain and *permit* was the PDP response,

the TEP stores the negotiated credentials in the NAS. In addition, the TEP will send a negotiation response (negotiation status) to the sender node via the protocol interface.

4.4 Summary

This chapter focused upon describing the architecture governing DTN including the system components for discovery and negotiation protocols. It described the discovery system components, which included the discovery interface (service), the controller, the protocol data processing, the management interface and a routing information handling component. It also described the components that make up the negotiation system, which included the negotiation service, the trust enforcement point, the negotiation agent, the PDP and SAML *plus*. The chapter presented how the SAML *plus* extends the functionality of an identity provider and how it compliments the negotiated attributed store. The negotiated attributed store in itself extends the components of a service provider by providing means of storing negotiated attributes which can subsequently be used as assertions in a SAML response. The chapter concluded with a detailed overview of the DTN architecture showing the flow of data between various components during trust negotiation.

5 Design and Formalisation of DTN

In this chapter, the formal model upon which DTN is based is presented. The chapter starts with the access control model, which serves as the basis for trust negotiation. Second, the DTN model is presented and its key concepts are then described: circle of trust and trust contracts. Third, credential equivalence, the basic concept underlying trust contracts is described. The chapter concludes with a discussion of how DTN limits the disclosure of access control policies.

5.1 Access Control

Access to resources needs to be controlled and managed especially in ensuring that operations carried out on resources are adequately authorised. Decisions have to be made and enforced in order to protect resources from unauthorised disclosure and alterations while confidentiality and privacy needs to be ensured where needed. Access management systems such as attribute-based access control systems (ABAC) [107, 119, 164] use information contained in policies and credentials to manage access. Desirable properties of an ABAC system include:

1. Decentralised attributes where an entity's attribute can be asserted by another entity;
2. Delegation of attributes between authorities in which the authority for an attribute can be delegated to another entity;
3. Attribute intersection in which combinations of attributes are used to infer another entity's attribute(s);
4. Attribute inference where an attribute can be inferred through another attribute;
5. Attribute fields which can be used for parameterised attributes such as defining quantities in a credential.

The key elements of an access control model and their formal representation include:

- S: Subject, defining an entity such as a user, software agent or organisation. There will typically be numerous instances of a subject: $s_0, s_1, ..., s_k \in S$;
- Obj: Object, defining resources or targets more generally. In access control, Objects are typically protected. There will typically be numerous instances of objects: $obj_0, obj_1, ..., obj_k \in Obj$;
- A: Action, defining actions that can be performed on objects. There will typically be numerous instances of actions: $a_0, a_1, ..., a_k \in A$;
- P: Permission, defining what actions are "allowed" to be performed on objects. $P = A \times Obj$ that is $P = < a_i, obj_j >$;
- R: Role is defined as $S.r(h_1, ..., h_n)$ where r is role name, h_i is a parameter that may be applied in case r is a parameterised role, and S is the entity who has the role;
- PS: Permissions to role relation, $PS \subseteq P \times R$.

With respect to these elements, [42] defines mapping functions that can be used to express object access in a role-based context:

- $RP(r : role) \rightarrow 2^{permissions}$, defines a role to permissions mapping, which gives the set of permissions that are assigned to role r. For example if *read* and *write* are permissions that can be assigned to role X, $permissions(X)$ has four possible permissions outcomes: {}, {read}, {write}, {read, write} for role X.
- $PA(p : permission) \rightarrow action$, defines the permission to action mapping, which gives the actions associated with permissions p.
- $PO(p : permission) \rightarrow object$, defines the object to permission mapping, which gives the objects associated with permission p.
- $SU(s : subject) \rightarrow user$, defines the subject to user mapping, which gives the users associated with subject s.
- $AR(s : subject) -> 2^{role}$, defines the active role mapping, which gives the set of roles in which subject s is active. For example if Bob could be granted role x and y, $AR(Bob)$ has four possible outcomes: {}, {x}, {y}, {x, y}.

To determine if a subject can access an object, the following axiom is typically used for access control:

$$access(subject, action, object) \rightarrow boolean$$

true: if subject s can access object obj and invoke action a

false: otherwise

For example, with respect to RBAC [44], object access authorisation can be defined as [42]:

$$\forall\, s : subject;\ obj : object;\ a : action \bullet access(s, a, obj) \Rightarrow$$
$$\exists\, r : role;\ p : permission \mid r \in AR(s) \wedge p \in RP(r) \wedge a \in PA(p) \wedge obj \in PO(p)$$

5.2 Trust Negotiation

Access control policies (a.k.a policies) and credentials can be defined with languages with well formed semantics and expressed as finite sets of statements [120]. Using propositional logic as in [117] a policy P_D for resource D can be defined as follows:

$$P_D \rightarrow F_D(C_1, C_2, ..., C_k)$$

where $C_1, C_2, ..., C_k$ are credentials that must be satisfied by the other party; F_D is an expression that uses these credentials, which may include boolean operators $\vee$ or $\wedge$ and any parenthesis where necessary. Access is granted to a resource D when the other party discloses sufficient C_i that satisfies $F_D(C_1, C_2, ..., C_k)$, i.e if evaluated to true.

Example 5.1 *Bob wants to access cancer patients records D at hospital X as part of a Cancer clinical trial (XCT). Hospital X's policy requires the requestor to be an investigator or clinician on the XCT clinical trial before access can be granted. Thus Bob provides credentials such as* C_1^{Bob} = *"Investigator" or* C_2^{Bob} = *"Clinician" and* C_3^{Bob} = *"XCT", which can be expressed as* $P_D \rightarrow (C_1^{Bob} \vee C_2^{Bob}) \wedge C_3^{Bob}$. *Similarly, Bob's release policies may specify that that the requesting target, which in this case is hospital X prove its identity amongst other properties. So for Bob's credential we have:* $P_{C^{Bob}} \rightarrow F_{C^{Bob}}(C_1, C_2, C_3)$.

In a nutshell, the policy of a resource is satisfied when the other party discloses the correct combination of credentials for that resource $(C_1, C_2, ..., C_k)$. A resource R is said to be unprotected if its access control policy is always satisfied $R \rightarrow true$ or $C \rightarrow true$. A resource is said to have a denial policy if $R \rightarrow false$, that is no credential can satisfy that policy or that resource is not meant for disclosure.

[117, 119] illustrate with examples how trust is established between two peers P_1 and P_2 with each of the peers requesting a series of credentials from one another and how requesting a credential in the series might trigger requests for credentials from the other party. One problem with exchanging credentials this way is that a point of deadlock can be reached where both parties wait on each other to disclose the next credential. This credential negotiation deadlock is explained in [117]. It occurs whenever there is a cyclic credential interdependency: $C_2^X \leftarrow C_2^{Bob}$ and $C_2^{Bob} \leftarrow C_2^X$, i.e. where their credential disclosure policies restricts who is first. [117] also proposed a possible solution to credentials negotiation deadlock. The solution introduces a collaborative peer to the negotiation process called a *locally trusted third party* (LTTP). An LTTP acts as a mediator by disclosing credentials and policy rules to negotiating parties whenever cyclic interdependency occurs to facilitate trust negotiation. A peer P_c is said to be an LTTP for P_a and P_b where P_c has previously successfully exchanged and cached several credentials on more than one occasion at different times with both P_a and P_b. Hence P_a and P_b ask P_c whom they both trust to act as their LTTP. P_c then releases missing credentials to both parties, which breaks the cyclic interdependency.

5.3 Dynamic Trust Negotiation

Dynamic trust negotiation (DTN) also known as dynamic negotiation through delegated trust (DNDT) is the process of negotiating trust between two non-trusting entities through trusted intermediary entities. Any entity can serve as a negotiator for other entities provided it is trusted by the two non-trusting entities or by their intermediaries. Like Automated trust Negotiation (ATN) [110], DTN introduces a mediator called a *trusted intermediary party* (TIP) similar to LTTP [117] in ATN. Unlike ATN, a TIP is just one of the multiple TIPs (many hops) that can exist in a trust negotiation between two peers. These multiple negotiation hops help to protect credentials and access policies.

Consider an example of dynamic trust negotiation between two peers P_1 and P_2, where P_1 is a requestor and P_2 is the domain of the resource R. With the understanding that credentials are also resources, two forms of resources exist in this example: Objects and Credentials. P_1 wants to access an object resource (R_1 or R_2) on P_2. P_1 will have to first negotiate its credential resource[1] for P_2's credential. P_2 has never negotiated with P_1 and it is only open for negotiation with peers it has previously negotiated with such as P_3 or P_4. This is referred to as *circle of trust* in this thesis and is shown in Figure 5.1. Suppose P_2's access policy for

[1] From here on, 'resource' implies 'credential resource'.

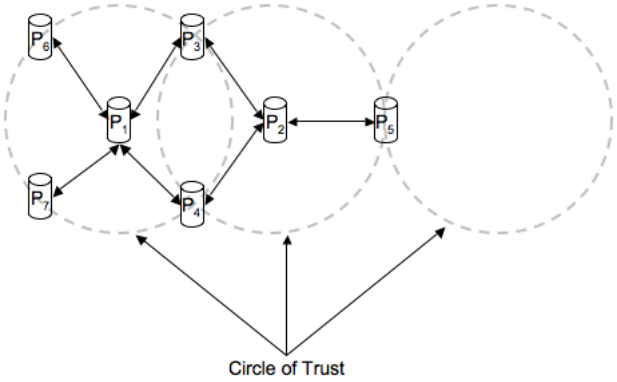


Figure 5.1: Circle of Trust

R is:

$$
\begin{aligned}
&R_1, R_2 \subseteq R \\
&\quad R_1 \leftarrow C_1^{P_3} \wedge C_2^{P_3} \\
&\quad R_2 \leftarrow C_3^{P_3}
\end{aligned}
$$

which means P_2 requires credential C_1 and C_2 from P_3 for resource R_1 while C_3 is required for R_2. Suppose P_1 belongs to the P_3 circle of trust and that P_3 access policy with P_1 is:

$$
\begin{aligned}
C_1^{P_3} &\leftarrow C_1^{P_1} \wedge C_2^{P_1} \\
C_2^{P_3} &\leftarrow C_3^{P_1} \\
C_3^{P_3} &\leftarrow C_3^{P_1} \wedge C_4^{P_1}
\end{aligned}
$$

To access R_1, P_1 would have to negotiate with P_3 by making available:

$$\{C_1^{P_1}, C_2^{P_1}, C_3^{P_1}\}$$

while P_3 will negotiate on behalf of P_1 with:

$$\{C_1^{P_3}, C_2^{P_3}\}$$

However if P_1 only makes available:

$$\{C_3^{P_1}, C_4^{P_1}\}$$

then P_3 can only negotiate on behalf of P_1 with:

$$\{C_3^{P_3}\}$$

which will be for P_2's R_2 and not R_1.

From this it would be seen that P_3 serves as a link peer known as TIP for the trust negotiation between P_2 and P_1.

$$P_2 \leftarrow P_3$$
$$P_2 \leftarrow P_3.P_1$$

In a typical trust negotiation, where a circle of trust exists, it is often possible to find multiple LTIP involved in the trust negotiation. Each of the involved TIP act as a hop or a link to the next TIP and/or finally to the target peer.

5.4 Circle of Trust

DTN introduces the concept of *circle of trust (COT)* [10, 151] for trust negotiation. The COT described here is not related to the Liberty ID-FF [71] circle of trust that implies a federation of service and identity providers. Figure 5.1 shows a COT that describes a circle of *intermediary* peers that are trusted by a peer in which one or more trust-contracts exist between the peers and each *intermediary* peer. A trust contract is an agreement that exists between two entities. This circle of trusted peers enable interactions between seemingly non-trusting domains.

Through overlapping *COTs* a trust-pathway (chain of trust) can be discovered. Consider two peers P_1 and P_5 in Figure 5.1, where P_1 is a requester and P_5 is a resource provider in another domain. P_1 and P_2 has $\{P_3, P_4, P_6, P_7\}$ and $\{P_3, P_4, P_5\}$ in their COT respectively. For P_1 to access P_5 resources, they will need to be trusted by P_2. In addition, P_2 will need to understand and trust credentials from P_1. Since P_1 has trust relationships with $\{P_3, P_4\}$, which are also in a trust relationship with P_2, P_1 will initiate a trust negotiation with P_2 through $\{P_3, P_4\}$. Similarly, P_2 will initiate a trust negotiation with P_5. Thus $\{P_3, P_2\}$, $\{P_4, P_2\}$ are trust-pathways between P_1 and P_5. Hence trust is realised by exploring overlapping COTs between P_1 and P_5.

$$P_1 \leftarrow (P_3 \vee P_4) \leftarrow P_2 \leftarrow P_5$$

More genreally, trust is realised between P_i and P_j when:

$$P_i \leftarrow P_j :$$
$$COT(P_i) \cap COT(P_{i+1}) ... \cap COT(P_j) \neq \{\}$$

COT may improve the likelihood of successful negotiations as TIPs can cache trust chains from previous negotiations, which will reduce the likelihood of future negotiations failing. The cache can also speed up future trust negotiations.

The advantages of having COT are quickly overshadowed as the number of overlapping $COTS$ increases. This is because the more hops you have, the less likely peers will be delegating privileges in open decentralised collaborative environments.

Despite this limitation, COT provides an additional benefit. Overlapping COTs can help to abstract virtual organisations through which trust can be discovered and realised dynamically. In virtual organisations, a relational hierarchy often exists, which can be modelled over the underlying COTs.

5.5 Trust Contract

In DTN, trust is viewed as the possession of *authentic* and *valid* credentials necessary for access control at an end point - typically a target with access control policies defined by the target resource providers. A credential is either *valid* and *authentic* or only *authentic*. An authentic credential implies a verifiable and un-tampered credential, while a valid credential implies a semantically correct credential that is acceptable, useable and tenable to an end point. Trust negotiation aims at delivering valid credentials that are authentic, and able to satisfy an access policy.

However, the presence of multiple domain authorities and policy enforcement points introduce a policy semantics divide between domains. That is knowing for example in some context that $org1.investigator$ is equivalent to $org2.investigator$. To address this divide, trust contracts *(TC)* are introduced in DTN [10]. As defined earlier, a trust contract is an agreeable arrangement made between two mutually suspicious entities to trust each other to some extent, which include agreement on identity and key management, credential mappings and delegation of access attributes such as roles. Trust contracts provide one mechanism to overcome the semantic issue of what a credential from one domain means (or should mean) in another domain. Trust contracts implicitly require that overlapping COTs exist.

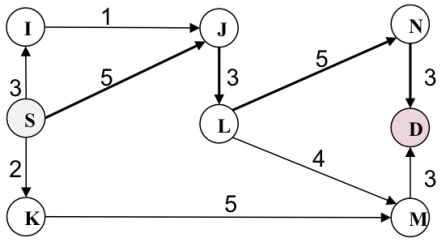


Figure 5.2: A network of collaborating health organisation

Figure 5.2 shows an abstract network of a collaborative environment. The network is a directed and acyclic graph, denoted as G(V,E). The Node V set represents autonomous organisations. A node refers to an end-point in a communication chain and consists of security credentials. The Edge E set represents trust and the direction of trust, which consists of policies and trust agreements (or TC). As described in Section 2.5, trust is asymmetric and this is represented with directed edges. If the trust between two nodes is reciprocal, this will be represented with a bidirectional edge.

Edges have weights, which represents the cardinality of TC sets between two Nodes. The minimum weight on an edge is 1, that is Edge e implies the existence of at least one TC between two nodes. For example, in Figure 5.2 the path that has the highest total weight between sender node S and destination node D is path SJLND since it has in total of 16 trust contracts. Nodes use weights (number of TC) to prioritise negotiations with other nodes as the path with the higher weight is more likely to succeed during trust negotiation. Since nodes cannot determine the total weight on a path, they can prioritise nodes in their COT based on the number of TCs they hold.

A trust contract, $tc \in TC$ is an agreement between two nodes (u, v) that states the mapping/ relationship that exists between two credentials (c^u, c^v). A TC exists between two nodes when more than one credential mapping is agreed between them, that is:

$$TC = (\{c^{u_0}, c^{v_0}\}, \{c^{u_1}, c^{v_1}\}, ..., \{c^{u_k}, c^{v_k}\})$$

Relationships between credentials are based on *credential equivalence rules*, discussed in Section 5.6. A tc stems from these rules, which are modelled by the function tc. More formally, let c^u and c^v be the set of credential in domain u and v respectively.

$$[c^u, c^v]$$

$$tc : c^u \nrightarrow c^v$$
$$\exists\, x : c^u;\, y : c^v \bullet tc(x) = y$$

tc is a partial function that implies a credential in one domain can only be uniquely associated with at most one credential of another domain (co-domain). Although not all credentials in a domain can be associated with a credential of a co-domain, one or more credentials in a domain can be associated with a credential in another domain. The partial function represents the fact that during credential agreement, the receiving domain (co-domain) may be willing to make available one of its credentials in exchange for one or more credentials from the requesting domain. A domain is not obliged to make agreements over all its credentials with a co-domain nor is a co-domain obliged to make agreements with all of its credentials. Thus credential x cannot output different results each time the function is called except if the underlying trust agreement has changed. For example, if credential x is uniquely agreed for y, $tc(x)$ can only have y as its output. Section 6 describes in detail four policies that are used in DTN, including TC-based policies.

5.6 Credential Equivalence

Trust contracts provide one solution to credential equivalence problems that exist between autonomous organisations by using equivalence rules. Credential equivalence rules define the relations that exist between credentials. These relations are used in the folding of one credential to another between different organisations [107]. Credential equivalence rules upon which DTN trust contracts are based include:

1. Transitive membership rule:

$$R \leftarrow R_1$$
$$R_1 \leftarrow R_2$$
$$\Rightarrow R \leftarrow R_2$$

This rule means that R_1 is a member of R and R_2 is a member of R_1, then R_2 is a member of R. As an example,

$$org1.investigator \leftarrow org2.healthpractitioner$$
$$org2.healthpractitioner \leftarrow org3.specialist$$
$$\Rightarrow org1.investigator \leftarrow org3.specialist$$

2. Linking delegation rule:

$$R \leftarrow R_1 \cdot R_2$$

This rule means an entity that has R_2 can act as R if the entity is contained in R_1. This requires at least two dependent roles. As an example,

$$\begin{aligned} Org_1.CancerTrial &\leftarrow Org_2.LeukemiaTrial.Investigator \\ Org_2.LeukemiaTrial &\leftarrow VOTES \\ VOTES.Investigator &\leftarrow Bob \\ \Rightarrow Org_1.CancerTrial &\leftarrow Bob \end{aligned}$$

This implies that for this linking delegation, Bob must satisfy the requirement that he is an Investigator on the VOTES project and also satisfy that VOTES is a member of Org_2 Leukemia Trial. In this case, VOTES is a link since Investigator is a role in VOTES.

3. Intersection rule:

$$R \leftarrow R_1 \cap \cdots \cap R_k$$

This implies an entity that has $R_1, R_2, ...,$ and R_k is delegated R. For example,

$$Org_1.BrainIT \leftarrow Org_2.Neurologist \cap Org_2.Consultant$$

This means an entity who is a Consultant and a Neurologist in Org_2 can participate in the Org_1 BrainIT study. It is an intersection and not a union as the rule suggest a commonality factor exist in each of $R_1, R_2, ...,$ and R_k. This rule can be used in situations where a domain requires authorisation attributes to be proved with more than one credential. For instance Bob is a Neurologist and he is also a Consultant who works with patients that have brain disorders. However, before Bob can be allowed to access BrainIT records, he must satisfy at least two credentials that prove he specialises in brain and nervous system disorders.

5.7 Limiting Disclosure of Access Control Policies

In trust negotiation, peers limit what credentials are disclosed by means of various negotiation strategies. In [112], two strategies were proposed: *eager* and *parsimonious*. An eager

negotiation strategy requires a party to disclose all of its credential to the other non-trusted party right at the start of a negotiation. The benefit of this strategy is that it assures a negotiation will succeed where successful negotiation is possible. Successful negotiation refers to the point when all credential disclosure policies are satisfied. This strategy however discloses more credentials than are potentially necessary thereby reducing party privacy. On the other hand, the parsimonious negotiation strategy enables a party to disclose its credentials only after it has been requested and after the necessary disclosure policy for that credential has been satisfied. This strategy increases party privacy and reduces the risk of unnecessary credential disclosure. However, this strategy can result in credential negotiation deadlock as explained in [117], which occurs when there is cyclic credential interdependency.

A solution to the problem of disclosing sensitive credentials is to limit what is disclosed to a total stranger and to gradually establish trust [118]. In DTN, credentials are only disclosed to intermediary parties, which are trusted with the expectation that privileges would be delegated to it that would not be directly offered to non-trusted parties. As negotiations take place from one intermediary party to another, the likelihood of a remote access control policy being deduced is small. Furthermore, as trust contracts are added or revoked, or as changes in trust-pathways between nodes occur, the chances of deducing remote access policies is further reduced.

Similarly by the nature of trust contracts, credentials should not be unnecessarily disclosed, as both parties are aware of their contract. These contracts limit what credentials can be accepted and which credential can be delegated. Trust can only be negotiated within the constraints of these contracts. However, these constraints only hold during party interactions and do not restrict what inferences parties can make with credentials during negotiations. These inferences enable parties to compute inter-contract relationships, which can subsequently improve the likelihood of successful negotiations.

5.8 Summary

This chapter gave a detailed description of the design and formalisation of DTN. It has presented extensive discussion on access control properties and attributes, access control key elements and model representation, access control policies and credentials. Dynamic trust negotiation (DTN), which is the process of negotiating trust between two non-trusting entities through trusted intermediary parties (TIPs) was presented. Automated trust negotiation (ATN) was also discussed in the chapter to highlight the similarities and differences with

DTN.

The concepts of circle of trust (COT) for trust negotiation was thoroughly explained. The discussion showed that through overlapping COTs a trust pathway can be discovered, which can serve as the basis for trust discovery in DTN. It was highlighted that COT may improve the likelihood of successful trust negotiations as TIPs can cache trust chains from previous negotiations. With respect to virtual organisations (VO), COT can model the relational hierarchies that can often exist, thus enabling the dynamic discovery and realisation of trust in a VO.

In the trust contract section, credentials were discussed and whether they are valid and authentic or only authentic. The effect of multiple domains and various relationships between credentials was presented. Semantic heterogeneity and the impact upon establishment and utilisation of trust contracts in DTN was discussed. Various credential equivalence rules such as transitive membership rules, linking delegation rules, and intersection rules were also described. These equivalence rules serve as the underlying concept for trust contract in DTN.

Unlike the trusted third party in ATN that only helps to facilitate trust negotiation, multiple TIPs in DTN facilitates negotiation and also helps to prevent the disclosure of credentials and access policies. The chapter concluded with the rationale for limiting disclosure of access control policies and discusses two different strategies namely eager and parsimonious.

6 Policy Specification

This chapter specify the syntax and semantics of the policy language used in DTN. The policy language itself is introduced in Section 6.1. The syntax of the policy language rule is specified in Section 6.2 and Section 6.3 specifies the syntax of the associated policy language functions. These functions and rules are used to describe policies whose syntax is given in Section 6.4. Section 6.5 describes credentials and their use for exchanging authorisation information between policy domains. DTN policies are described in detail in Section 6.6 and the chapter concludes with how conflicts are resolved between policies.

6.1 Policy Language

In order to understand policies it is imperative to understand the language in which they are described and specified. To describe the types of policies used for DTN, a rule-based policy language is used. The rule-based framework, as described in [2, 165, 52], is adopted. This framework offers a declarative language that is based on logic to define rules. This makes it possible to implement policies in various languages and programming tools. For instance the policy language described in this chapter is implemented in DTN using the XACML policy specification language [163].

A rule consists of a rule head and rule body. It takes the form:

$$Rh \leftarrow Rb$$

where Rh is the rule head and Rb is the rule body.

In this model, satisfaction of Rb makes Rh true. A rule body typically contains conditions and if these conditions are satisfied, i.e evaluated to true, the rule will be enforced and some action can be triggered. Thus an authorisation policy can be encoded in a rule body and the

rule head can be used for encoding the authorisation. For example, a policy rule that decides on whether subject s can access resource r can be defined as:

$$canAccess(s, r) \leftarrow isOwner(s, r)$$

Given s and r, if the function $isOwner(s, r)$ evaluates to true, i.e. if subject s is the owner of resource r, then access to the resource is granted; otherwise access is denied. If the body of a rule is empty, such as $Rh \leftarrow .$ this implies the rule has no conditions and is always true.

6.2 Policy Language Rules

The policy rule language used in DTN is based on the Cassandra [2] policy rule language. The policy rule language is designed to support and use credentials for trust management. A credential is viewed as a constrained predicate, asserted by an entity called an *issuer* and held by an entity called a *holder*. In a rule, predicates are tagged with both the holder and issuer. Thus $Bob\Diamond OX.p(x)$ is a rule that says Bob holds a predicate $p(x)$ asserted by Oxford, denoted by OX. The holder is postfixed by the symbol "$\Diamond$" and the issuer by ".".

Policy rule		::=	$(Head, Body\ Pred, Constraint)$
Head	head	::=	$E_1 \Diamond E_2.pred$
Body Pred	P	::=	$e_1 \Diamond e_2.pred$
Credential	$cred$	::=	$head \leftarrow c$
Rule	rule	::=	cred
		|	$head \leftarrow P_1, \ldots, P_n, c$
Constraint	c	$\in$	C

The prefix used here for the issuer is similar to the prefix used for roles in [107, 119]. In addition to the issuer prefix, the holder prefix used here is similar to the location prefix used in Cassandra. As in [2], the syntax for policy rules is as follows[1]:

Definition 6.2.1 ***(head, body, location, issuer, credential, policy)***[2] *A policy rule is made up of a prefixed head predicate, a list of prefixed body predicates, and constraints C. The prefixes e_{hol} and e_{iss} of a predicate $e_{hol}\Diamond e_{iss}.p(\vec{e})$ are referred to as the holder and the issuer of the predicate, respectively. The entity that issues a rule is called the issuer of the rule and the entity that holds the rule is called the holder of the rule. A rule of the form*

[1] Definitions for Entities E, Role Names R and Action Names A are presented in [2].
[2] This definition has been modified from [2].

head $\leftarrow$ *c is called a credential or credential rule. A policy rule that is not a credential rule has identical holder and issuer. Only credentials may have an issuer that differs from the holder: these represent credentials that are issued by foreign parties.*

The typing rules for the above policy rule as specified in [2] are as follows:

$$\frac{\Gamma \vdash e : \mathit{entity} \qquad \Gamma \vdash e' : \mathit{entity} \qquad \Gamma \vdash \mathit{pred}}{\Gamma \vdash e \Diamond e'.\mathit{pred}}$$

$$\frac{\Gamma \vdash \mathit{head} \qquad \Gamma \vdash c}{\Gamma \vdash \mathit{head} \leftarrow c}$$

$$\frac{\Gamma \vdash \mathit{head} \qquad \Gamma \vdash P_1 \qquad \cdots \qquad \Gamma \vdash P_n \qquad \Gamma \vdash c}{\Gamma \vdash \mathit{head} \leftarrow P_1, \cdots, P_n, c}$$

The first inference rule represents a rule head, which can be understood as follows: the proposition $e \Diamond e'.\mathit{pred}$ is from the set of propositions for a rule head if e, e' and *pred* are from the same set of propositions where e, e' are entities. In simple terms, these rules express type definitions for the prepositions, e.g. $e \Diamond e'.\mathit{pred}$ receives two input variables of type *entity* and a variable of type *pred*.

The typing rules of predicates[3] (*pred*) as used in DTN are presented in Section 6.3.

A rule constraint that is true is often not shown, so also are the issuer and holder of a rule if they refer to the same entity. Similarly, the holder and/or issuer of a rule is not shown if it is identifiable from the context. For example,

$$D \Diamond D.p \leftarrow D \Diamond D.p_1, D \Diamond E.p_2, \\ E \Diamond F.p_3, \mathit{true}$$

[3]Predicates are often referred to as functions in this chapter except where it is stated as mapping functions.

can be written as,

$$D.p \leftarrow D.p_1, E.p_2, E \Diamond F.p_3$$

or

$$p \leftarrow p_1, E.p_2, E \Diamond F.p_3$$

Since D is the holder and issuer of the rule, it is clear from the context that the rule is for D's domain and the rule can be found in D's policy[4]. Intuitively, this rule can be interpreted as this:

> D may have a service that can be accessed based on an assertion p. p can be deduced from D's policy if: p_1 can be deduced from D's policy; the credential asserting p_2 issued by E is available, and that a request from E for a credential asserting p_3 issued by F exists.

From definition 6.2.1, p and p_1 are assertions in D's policy rules since D is the issuer and holder respectively. p_2 is a credential rule assertion, since the issuer and holder are different, i.e. E and D respectively. p_3 is also a credential rule assertion where F is the issuer and E is the holder.

Authorisation Function	**Description**
$assign(e.p(\vec{e}), r, p)$	implies that an entity with a credential asserting $p(\vec{e})$ issued by e is assigned role r that is granted permission p.
$can\_activate(e, r)$	implies entity e can activate role r, i.e e has been assigned r or can act with the capability of r.
$has\_activated(e, r)$	implies that entity e is currently acting with the capability of role r.
$can\_invoke(e.p(\vec{e}), r)$	implies an entity can invoke a trust contract for role r if the entity has a credential asserting $p(\vec{e})$ issued by e.
$has\_invoked(e, r)$	implies entity e has currently invoked the trust contract for role r.
$can\_release\_cred(e_1, e_2.p(\vec{e}), e_3)$	implies a credential asserting $p(\vec{e})$ issued by entity e_2 can be released to e_1 if e_3 has invoked its trust contract with e_2.

Table 6.1: DTN Authorisation Functions (Predicates)

[4]The policies described in Section 6.6 are in the context of the Glasgow domain.

6.3 Policy Language Functions

Based on definition 6.2.1, policy rules can be defined using predicates. A predicate can be defined as a Boolean function that returns a truth value. Typically, a function has set of arguments and a return value. The return value is true if the conditions for the predicate test is satisfied or false otherwise. Two categories of policy functions are defined in RDM [52]: specification functions and authorisation functions.

Specification functions express basic functions that are defined in the RBAC [42, 44] and often referred to as mapping functions as described in Section 5.1. Authorisation functions are used to express authorisation policies. The authorisation functions defined for DTN are shown in Table 6.1. These functions (predicates) were introduced in Section 3.4 and are used for expressing DTN authorisation policies[5].

These functions return Boolean's, i.e. true or false. The syntax for these functions are as follows:

Predicate	*pred*	::=	$assign(e.pred, r, permission)$
		|	$can\_activate(e, r)$
		|	$has\_activated(e, r)$
		|	$can\_invoke(e.pred, r)$
		|	$has\_invoked(e, r)$
		|	$can\_release\_cred(e_1, e_2.pred, e_3)$
		|	$p(e_1, ..., e_n)$, where n ≥ 0
Permissions	*permission*	::=	$q(a_1, ..., a_n)$, where n ≥ 0
Resource	q	$\in$	*ResourceNames*

[5] A trust contract is often referred to as a contract in this chapter.

The typing rules for the above functions are as follows[a]:

$$\frac{\Gamma \vdash e : \mathit{entity} \qquad \Gamma \vdash r : \mathit{role}(\tau) \qquad \Gamma \vdash \mathit{pred} \qquad \Gamma \vdash \mathit{permission}}{\Gamma \vdash \mathit{assign}(e.\mathit{pred}, r, \mathit{permission})}$$

$$\frac{\Gamma \vdash e : \mathit{entity} \qquad \Gamma \vdash r : \mathit{role}(\tau)}{\Gamma \vdash \mathit{can\_activate}(e, r)}$$

$$\frac{\Gamma \vdash e : \mathit{entity} \qquad \Gamma \vdash r : \mathit{role}(\tau)}{\Gamma \vdash \mathit{has\_activated}(e, r)}$$

$$\frac{\Gamma \vdash e : \mathit{entity} \qquad \Gamma \vdash r : \mathit{role}(\tau) \qquad \Gamma \vdash \mathit{pred}}{\Gamma \vdash \mathit{can\_invoke}(e.\mathit{pred}, r)}$$

$$\frac{\Gamma \vdash e : \mathit{entity} \qquad \Gamma \vdash r : \mathit{role}(\tau)}{\Gamma \vdash \mathit{has\_invoked}(e, r)}$$

$$\frac{\Gamma \vdash e_1 : \mathit{entity} \qquad \Gamma \vdash e_2 : \mathit{entity} \qquad \Gamma \vdash e_3 : \mathit{entity} \qquad \Gamma \vdash \mathit{pred}}{\Gamma \vdash \mathit{can\_release\_cred}(e_1, e_2.\mathit{pred}, e_3)}$$

$$\frac{\begin{array}{l} q \in \mathit{ResourceNames} \\ \Gamma \vdash a_1 : \mathit{action} \qquad \cdots \qquad \Gamma \vdash a_n : \mathit{action} \end{array}}{\Gamma \vdash q(a_1, ..., a_n)}$$

[a]The syntax for roles and actions along with their typing rules are presented in [2].

In simple terms, these rules express type definitions for the formulas. The first rule says the formula, i.e. preposition $\mathit{assign}(e.\mathit{pred}, r, \mathit{permission})$ receives $e, r, \mathit{pred}, \mathit{permissions}$ as input variables, where e is of type *entity*, r is of type $\mathit{role}(\tau)$ and τ is a variable of some type. $\mathit{role}(\tau)$ represents parameterised roles, e.g. Researcher('in computing science dept'), but for non-parameteirsed roles, *role* is simply sufficient. The second rule says the formula $\mathit{can\_activate}(e, r)$ receives two variables of type *entity* and $\mathit{role}(\tau)$. The third, fourth, fifth and sixth rules can be explained in like manner. *ResourceNames* represent a set of user

defined names for a resource e.g. a service, a database table and a file name. The last rule says q an object in *ResourceNames* receives a set of variables of type *action* to represent a permission.

6.4 Authorisation Rules for Policy Enforcement

Various authorisation rules can be used to enforce DTN security policies. These rules are defined as follows:

Rule 1 *: An assign-role rule is of the form:*

$$\begin{aligned} &assign(e.p(\vec{e}), r, p) \leftarrow \\ &has\_invoked(e, r), \\ &p \in permissions(r) \end{aligned}$$

where e, r, p are entity, role and permission respectively.

This rule implies that an entity that holds a credential asserting $p(\vec{e})$ issued by e can be assigned role r granted with permission p. This access control rule says that an entity that has invoked role r is allowed permission p. A permission is a set of actions permitted on an object or resource and $permissions(r : role)$ is a mapping function based on $RP(role)$ mapping function described in Section 5.1. This rule can can also be written as:

$$\begin{aligned} &assign(e_2.can\_activate(e, r_2), r, obj(acts)) \leftarrow \\ &has\_invoked(e_2, r), \\ &acts \in \{act_1, ..., act_n\}, \\ &obj(acts) \in permissions(r) \end{aligned}$$

where e, e_2 are entities, r, r_2 are roles and $act_1, ..., act_n$ are actions associated with object, obj.

The rule implies that entity e can be assigned role r and granted permission $obj(acts)$, if firstly a credential asserting that e has r_2 issued by e_2 is made available, secondly if e_2 has already invoked a contract for role r. In a scenario where e_2 is a local entity and r_2 is in the same domain, the *has_invoked* predicate will always be deduced as true.

Rule 2 *: An entity-role activation rule is of the form:*

$$can\_activate(e, r) \leftarrow has\_activated(e, r_2).$$

where e is an entity, and r, r_2 are roles.
This rule implies that an entity e can act with the capabilities of role r if the entity has first activated role r_2. Often, *can_activate* is used as a predicate in a credential rule asserting a role is assigned to an entity.

Rule 3 *: An entity-role active rule is of the form:*

$$has\_activated(e, r) \leftarrow c$$

where e, r are entity and role respectively.

This rule is true if entity e has role r active. If true, the rule suggests that e is a member of role r and has presented a credential asserting r. The rule can also be expressed as $has\_activated(e, r) \leftarrow isActive(e, r)$, where $isActive(e, r)$ is a mapping function that returns true if in r's domain, e has role r active. Basically, *isActive* is a state checker that can be used to check the state of a role. The *has_activated* predicate is often used with $can\_activate(e, r)$ predicate to decide if an entity has a role active and if not, whether it is allowed to activate the role assuming no other constraints exist.

Rule 4 *: A credential-role invocation rule is of the form:*

$$can\_invoke(e.p(\vec{e}), r) \leftarrow c$$

where e, r, c are an entity, role and constraint respectively.

This rule implies that an entity with a credential asserting $p(\vec{e})$ issued by e can invoke the contract for role r if the constraint c can be deduced to be true. The rule can also be expressed as:

$$can\_invoke(e_1.can\_activate(e_2, r_1), r) \leftarrow$$
$$tc(r_1) = r$$

where e_1, e_2 are entities, r, r_1 are roles and $tc()$ is a trust contract mapping function. For simplicity, $tc(r_1) = r$ represents $tc(e_1.can\_activate(e_2, r_1)) = can\_activate(x, r)$, where x is

an entity variable.

This rule says e_2 can invoke a contract for role r if the credential asserting e_2 has received r_1 issued by e_1. This rule can be deduced to be true if a trust contract exists that maps r_1 to r. $tc(credential)$ is used as the rule constraint.

Rule 5 *: An entity has invoked rule is of the form:*

$$has\_invoked(e, r) \leftarrow c$$

where e, r, c are an entity, role and constraint respectively.

The rule can also be expressed as $has\_invoked(e, r) \leftarrow inSession(e, r)$, where $inSession()$ is a mapping function that returns true if in the current session state, a trust contract for role r has already been invoked by e. This is a rule that can show if a potentially remote entity has been assigned a role based on a trust contract. Like *isActive*, *inSession* is a state checker that can be used to check the invocation state of a contract. $inSession()$ is used as the rule constraint.

Rule 6 *: An entity can receive credential rule is of the form:*

$$can\_release\_cred(e_1, e_2.p(\vec{e_1}), e_3) \leftarrow e_2.p(\vec{e_2}), c$$

where e_1, e_2, e_3 are entities, $p(\vec{e_1}), p(\vec{e_2})$ are rule predicates and c a constraint.

The rule says entity e_1 can delegate a credential that asserts $p(\vec{e_1})$ issued by e_2 if $p(\vec{e_2})$ and c can be deduced. The rule can also be expressed as:

$$\begin{aligned} &can\_release\_cred(e_1, e_2.can\_activate(e_1, r), e_3) \leftarrow \\ &e_2.has\_invoked(e_3, r), \\ &e_1 \in COT(e_2) \end{aligned}$$

This rule says entity e_1 can be delegated a credential issued by entity e_2 which asserts that e_1 can activate role r if e_1 is in e_2's *circle of trust* and if e_3 has invoked the contract for role

r. $COT(e : entity)$ is a mapping function that returns the set of entities in an entity's COT. This rule can also be written as:

$$can\_release\_cred(e_1, e_2.can\_activate(e_1, r), e_3) \leftarrow$$
$$e_2.has\_activated(e_3, r),$$
$$e_1 \in COT(e_2)$$

where e_1, e_2, e_3 are entities and r is a role.

This rule is same as the first rule except that in this case, entities e_2 and e_3 are from the same local domain, and in some cases e_2 and e_3 can refer to the same entity. The example in Section 6.5 makes this distinction clear. However, it should be noted that a foreign entity should not be able to activate a local role, they can only invoke a contract for a local role. Likewise, a local entity should not be able invoke a contract for a local role rather they should activate a local role.

Datalog [166] examples of these rules are listed in Appendix C.

6.5 Credentials and Distributed Policy Rules

Credentials serve as the basis for trust management. They are verifiable statements issued to establish a specific claim. A digital credential is a digitally asserted statement that allows the issuer to be verified. The statement can be used for authorisation if the issuer is trusted to have authority or be in a position to make such an assertion. An entity who signed and issued a credential is called an *issuer*. The entity a credential is assigned to is called the *holder*.

6.5.1 Attribute Certificate

An X.509 Attribute Certificate (AC) [39] is a data construct for encoding security information through which trust can be managed in an open environment. It is similar to an X.509 certificate (PKC) but is signed and issued by an attribute authority (AA). Unlike PKC, ACs do not contain a public key. Instead they contain a series of attributes associated with their holder. Privileges in the form of roles, group membership, security properties or information are represented as attributes. ACs can be likened to capabilities in that they bind an AC

holder with a set of authorisation attributes. The structure of an X.509 AC is given in table 6.2.

Version number	the version number of the certificate e.g. v2
Serial number	the unique identifier for the certificate
Signature algorithm	the algorithm used in signing the certificate
Issuer name	the name of the entity that issued the certificate
AttrCertValidityPeriod	the period in which the certificate is valid
Attributes	sequence of attributes that the certificate represents
Holder	certificate holder
IssuerUniqueID	the identifier for the issuer
Subject unique identifier	the identifier for the subject
Extensions	optional, for additional information.

Table 6.2: X.509 Attribute Certificate v3

AC's are used as digital credentials, which can be securely exchanged to provide authorisation information to access control engines. An AC can either be pushed or pulled depending on the scenario that best suits the inter-domain communications. The push scenario is typically used in inter-domain cases where the entity requesting access has its attributes assigned within its "home" domain. The pull scenario is most useful in inter-domain cases where the entity requesting access has its attributes assigned and stored at a remote domain.

6.5.2 Distributed Policy Rules

Credentials can be used to exchange authorisation information that assert that an entity is assigned a role. A role can be tagged with the role authority similar to a credential issuer. As a credential, it can be assigned to a holder entity regardless of where the issuer is. It can also be stored at a location different from the holder's location. For instance an AC can be issued by an AA [39] and assigned to the entity in the holder field; the AC can be stored by the AA or at a repository/directory that is managed by a delegated domain. In DTN, the issuer is the entity that manages or stores the issued credential. For example a credential can be written as:

$$Bryan \Diamond GLA.can\_activate(Bryan, Investigator) \leftarrow .$$

This states that an entity called Bryan holds a credential issued by the University of Glasgow (GLA) asserting an investigator role. In subsequent representation, for clarity, the holder

domain is used instead of the holder to show the "home" location of the entity. Similarly, to be able to distinguish between local and remote roles, the organisation is postfixed by the symbol ".". For example the *Investigator* role can be written as $GLA.Investigator$.

Using the patient recruitment scenario in Section 3.6, the following example shows how local and foreign (remote) predicates are used in DTN. Alex, a researcher at University of Dundee (DUN), has a credential that can be written as:

$$
\begin{aligned}
&Alex \Diamond DUN.can\_activate(Alex, DUN.Researcher(dept, project, position)) \leftarrow \\
&dept = BiomedicalResearchCentre, \\
&project = CancerTrial, \\
&position = PrincipalInvestigator
\end{aligned}
$$

which represents a credential that is issued by the University of Dundee and held by Alex (expressed as $Alex \Diamond DUN$) that asserts that Alex is a researcher, who is a principal investigator of the cancer clinical trial program at the Biomedical Research Centre. If DUN is willing to release Alex's credentials, the University of Dundee will have policies that include a rule that states, for example, its researcher role can be released to Tayside Children's Hospital, Ninewells (TCHN). This can be written as:

$$
\begin{aligned}
&DUN.can\_release\_cred(TCHN, \\
&can\_activate(TCHN, DUN.Researcher(dept, project, position)), DUN) \leftarrow \\
&has\_activated(Alex, DUN.Researcher(dept, project, position)), \\
&TCHN \in COT(DUN), \\
&dept = BiomedicalResearchCentre, \\
&project = CancerTrial, \\
&position = PrincipalInvestigator
\end{aligned}
$$

Similarly at TCHN, there are policy rules that need to be satisfied if it is to be involved in the negotiation. Two types of policy rules would need to be satisfied for this scenario. The first TCHN rule can be writing as:

$$
\begin{aligned}
&can\_invoke(DUN.can\_activate(DUN, DUN.Researcher), local\_role) \leftarrow \\
&tc(DUN.Researcher) = local\_role, \\
&local\_role = TCHN.Honconsultant
\end{aligned}
$$

This rule implies that an entity from the University of Dundee who has a researcher credential issued by the University of Dundee can invoke the contract which says a DUN researcher role

is recognised as a TCHN honorary consultant role. The second TCHN rule identified can be written as:

$$
\begin{aligned}
&can\_release\_cred(GHG, can\_activate(GHG, local\_role), DUN) \leftarrow \\
&has\_invoked(DUN, local\_role), \\
&local\_role = TCHN.Honconsultant, \\
&GHG \in COT(TCHN)
\end{aligned}
$$

This rule says that TCHN honorary consultant role can be released to Gartnavel hospital Glasgow (GHG) if DUN has invoked the contract for TCHN honorary consultant role. In both rules, Alex (from above) is not referred to, but his organisation is. This is because in DTN, negotiation is viewed to be between organisations and not individuals. Secondly this enables rules to be more scalable since the entity (Alex in this case) is linked to the negotiation session (Section 7.3). In a situation where TCHN wants to restrict role access to a named entity, a constraint to that effect will be made, e.g. $sessionEntity = CN/\ Alex$. However, this requires TCHN at the point of formulating this policy to know the names of remote entities.

To support this DTN negotiation at GHG the following policy rules must exist. The first rule can be written as:

$$
\begin{aligned}
&can\_invoke(TCHN.can\_activate(DUN, TCHN.Honconsultant), local\_role) \leftarrow \\
&tc(TCHN.Honconsultant) = local\_role, \\
&local\_role = GHG.Honconsultant
\end{aligned}
$$

This rule implies that an entity from TCHN who has an honorary consultant credential issued by TCHN can invoke the contract TCHN has for GHG's honorary consultant role. GHG can deduce from the TCHN credential who the entity is. It is possible for GHG to use these deductions for finer grained access control.

The second GHG rule for the patient recruitment scenario can be written as:

$$
\begin{aligned}
&assign(TCHN.can\_activate(DUN, TCHN.Honconsultant), \\
&local\_role, Patientmaster(acts)) \leftarrow \\
&has\_invoked(TCHN, local\_role), \\
&acts = select, \\
&Patientmaster(acts) \in permissions(local\_role), \\
&local\_role = GHG.Honconsultant
\end{aligned}
$$

This represents what permissions TCHN's honorary consultant role is allowed within its domain. The rule implies that an entity that holds TCHN's honorary consultant credential is equivalent to GHG honorary consultant role and is granted a *select on patientmaster* permission.

Alex is able to get a query response from Gartnavel Hospital Glasgow despite the fact that his domain is not trusted, but because his researcher credential at Dundee University were able to be negotiated and able to satisfy each negotiating party's policies. However, if Alex (with the same researcher role) requested an update query, e.g. to change the PatientMaster table records, his request would be denied regardless of complete negotiations since this is not one of the permissions of a GHG consultant.

6.6 DTN Policies

Four types of security policies are used in DTN [167]: local policies; trust-contract policies; acceptance policies, and release policies. These triple-based policies consist of $\langle subjects, objects, actions \rangle$ and optionally $\langle obligations \rangle$. Each type of policy relates to different stages of a negotiation and for different resources. Two types of resources suffice in this section. These are credentials as resources and data objects, e.g. files and databases. In DTN, these policies are expressed as XACML [163, 84] policies. Each type of policy consists of multiple rules which can be combined to form a higher level PolicySet, i.e. a collection of policies.

The policies described throughout this section are samples of DTN policies which are defined and stored by a representative Glasgow (GLA) organisation.

6.6.1 Local Policies

These are policies put in place by service providers to make access decisions for services or resources they provide. The type of resources mostly protected by these policies are general resources like data files or database objects such as tables, data views or stored procedures. These policies are stored in a policy repository and are available to the policy decision point (PDP) for authorisation decisions on data resources. In a decentralised environment, service providers can often delegate authority to various AAs for privilege management allowing the creation and management of security policies which are decentralised and support fine-grained access control.

A tabular representation of some local policies are shown in Table 6.3. This shows various subjects (entities), resources that can be accessed and actions that can be performed. The subject field contains user names or roles or both. Thus a request may contain a user name and role as subject attributes. Obligations are conditions on policies, which are to be enforced by policy enforcement points (PEP). For example, the obligation Anonymise(NHSno, CHI) in Table 6.3 obliges the PEP to ensure that NHS number and CHI data are anonymised before they are made available.

Subject	**Resource**	**Action**	**Obligation**
Femi:GLA.GP	PatientMaster	Select, Insert, Update	
John:GLA.Investigator	PatientMaster	Select	Anonymise(NHSno, CHI)
GLA.Nurse	PatientMaster	Select, Insert	
GLA.Clinician	PatientMaster	Select	
GLA.Specialist	PatientMaster	Select, Insert	
GLA.GP/Investigator	PatientDrug	Select	Anonymise(NHSno, CHI)
GLA.VOTES	PatientDrug	Select, Insert	Anonymise(NHSno, CHI)
GLA.VOTES/GP	PatientDrug	Update	

Table 6.3: Tabular View of Local Policies
GP/Investigator is a role label. It indicates Investigator is a **subrole** of GP.

Local policies indicate what roles are available and the associated permissions for each role in an organisation. Each row in table 6.3 is enforced as a policy rule. Each of these rules can be described with *Rule 1*, described in Section 6.2:

$$assign(can\_activate(Femi, GLA.GP), GLA.GP, PatientMaster(acts)) \leftarrow$$
$$PatientMaster(acts) \in permissons(GLA.GP)$$

$$assign(can\_activate(x, GLA.GP), GLA.GP, PatientMaster(acts)) \leftarrow$$
$$PatientMaster(acts) \in permissons(GLA.GP)$$

In some instances, these rules can include obligations for roles, which must be enforced. Rules may be implemented in XACML policies, which are stored as PolicySets. In XACML, the results of each policy can be combined using policy-combining algorithms in order to reach a final authorisation decision. Use of a rule-combining algorithm allows the effects of all rules in a policy to be used for an authorisation decision for that policy. Based on the final authorisation decision reached, where applicable, obligations with a matching "fulfilOn" effect are also included in the PDP response. An extract of a policy that allows a GP to perform a select, insert or update on a PatientMaster table is shown in Figure 6.1.

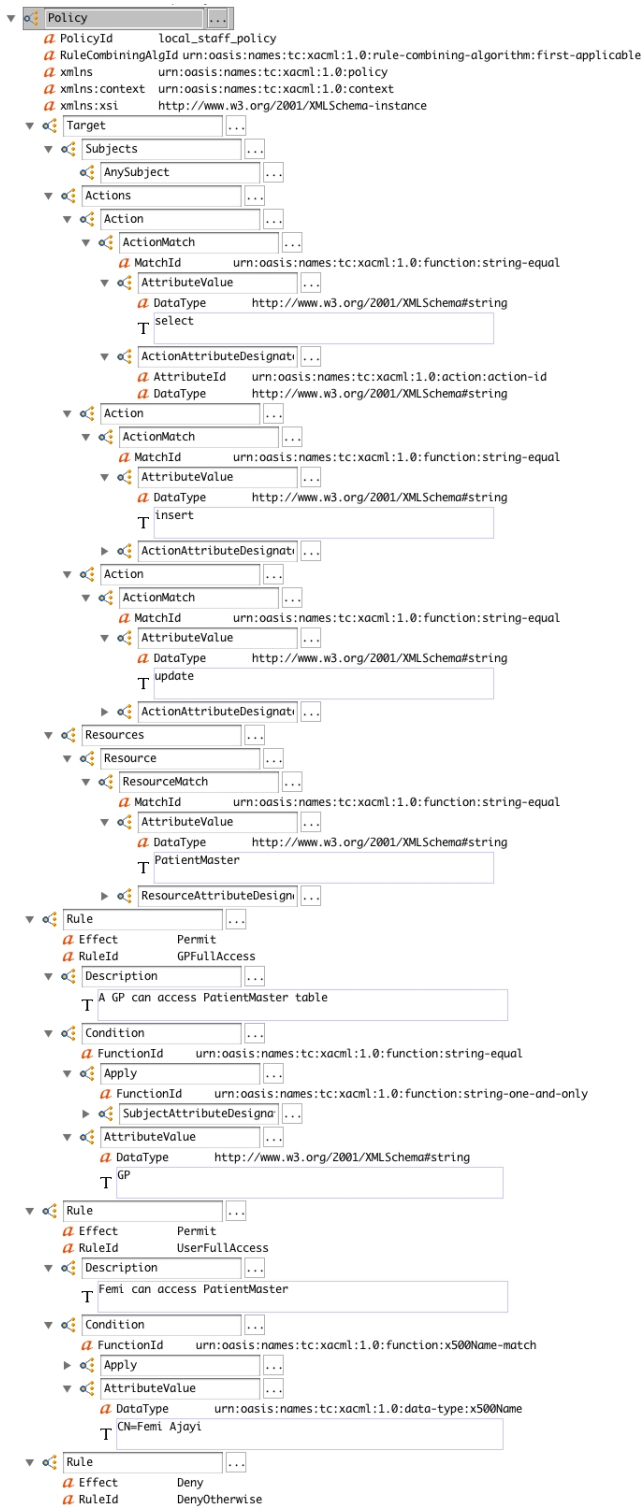


Figure 6.1: An extract of a local policy

In XACML rule combining algorithms [163] can be defined for policies so that multiple rules can be combined in one policy. Rules contained in a policy are evaluated and the results combined and evaluated in order to reach a policy decision. A rule contains a condition, which is a boolean function. If a rule condition is evaluated to be true the value of the "Rule Effect", which can be either a Permit or Deny, is returned for that rule. If a rule cannot be evaluated or if a rule is evaluated to be false, the result of the rule is Indeterminate or NotApplicable respectively. For example, the first two rules shown in Figure 6.1 have "Permit" as the value of their "Rule Effect". The third rule has a "Deny" value for its "Rule Effect" and since the rule has no condition it is evaluated to be true.

The policy shown in Figure 6.1 defines access control for the PatientMaster object (resource). The rules defined in this policy contain the logic of who can access the PatientMaster resource. Each subject in Table 6.3 who is listed to have access to the PatientMaster resource could be defined as a rule in the policy. The rule combining algorithm used in the above policy was the "first-applicable" algorithm. This means that the value of the first rule evaluated to be true is returned as the final result of the policy. For example, the above policy will return a permit result if the requester has a GP role or a CN for Femi Ajayi, otherwise it returns a deny result.

A sample of an XACML policy that allows an Investigator to *select only* records from PatientMaster table with obligations that log access and anonymise certain fields is shown in **Extract 1 of Appendix A**.

6.6.2 Trust-contract Policies

Trust-contract policies are similar to the local policies described in Section 6.6.1. However they are designed to validate and enforce trust contracts (or agreements) with respect to existing local policies. Trust-contract policies allow both trusted and third-party trusted entities to be included as subjects in a policy. They are used when a remote entity (trusted or a third-party trusted entity) requests a service or resource. Trust-contract policies provide restrictive access to resources and services in that they limit privileges of roles. For example, the first row in table 6.4 implies that *any one from Oxford (OX)* with a GP role issued by Oxford will have up to *full privileges* of a GP, i.e. they are able to perform select, insert or update of the PatientMaster resource. The second row in the same table implies that *any one* that presents a GP role issued by Oxford will have *partial privileges* of a GP, i.e. they are able to perform select or insert on the PatientMaster resource. A local policy showing *full privileges* of the GP role is illustrated by the first row of table 6.3.

Subject	Resource	Action	Obligation
OX:OX.GP, GLA.GP	PatientMaster	Select, Insert, Update	Anonymise(NHSno, CHI), logging
Any:OX.GP, GLA.GP	PatientMaster	Select, Insert	Anonymise(NHSno, CHI), logging
OX:OX.(Nurse ∩ Clinician), *GLA.GP*	PatientMaster	Select	Anonymise(NHSno, CHI), logging
IMP:IMP.GP, GLA.GP	PatientMaster	Select, Insert	Anonymise(NHSno, CHI), logging
Any:IMP.GP, GLA.GP	PatientMaster	Select	Anonymise(NHSno, CHI), logging
OX:OX.Investigator, GLA.GP	PatientMaster	Select	Anonymise(NHSno, CHI), logging
Any:OX.Investigator, GLA.GP	PatientMaster	Select	Anonymise(NHSno, CHI), logging
OX:OX.VOTES, GLA.VOTES	PatientDrug	Select, Insert	Anonymise(NHSno, CHI), logging
OX:OX.GP/Investigator, GLA.GP/Investigator	PatientDrug	Select	Anonymise(NHSno, CHI), logging
OX:D_{links}, GLA.VOTES/GP	PatientDrug	Update	Anonymise(NHSno, CHI), logging

Table 6.4: Tabular View of Trust-Contract policies

OX.GP is combination of two labels that implies OX issued a GP role i.e. OX's GP. OX.(Nurse ∩ Clinician) implies an entity must have both Nurse and Clinician roles credentials issued by OX. While D_{links} is an abbreviation for dynamic links discussed in Section 6.6.3. The intersection (∩) signifies commonality, that is to be able to perform *Select* on a PatientMaster resource. The remote entity is expected to provide more than one credential for roles that can perform *Select* on the PatientMaster resource. In this case, the remote entity is expected to provide credentials for Nurse and Clinician roles since *Select* with respect to PatientMaster is common to both Nurse and Clinician roles as defined in its local policies (see Table 6.3).

Trust-contract policies include obligations for negotiation requests received from remote entities. A tabular representation of some trust-contract policies is shown in Table 6.4. Each row represents a trust contract, which is enforced as one or more policy rules. *Rule 1* described in Section 6.2 can be used to enforce trust-contract rules:

$$assign(OX.can\_activate(OX, OX.GP), GLA.GP, PatientMaster(acts)) \leftarrow$$
$$has\_invoked(OX, GLA.GP),$$
$$acts = \{select, insert, update\},$$
$$PatientMaster(acts) \in permissions(GLA.GP)$$

The $permissions(GLA.GP)$ returns all permissions granted to $GLA.GP$, that can be deduced from permissions allowed to $GLA.GP$ in GLA local policies. It can be used to enforce permission constraints for trust contracts and also to ensure that permissions granted are restricted to allowed local policy permissions.

The subject field in Table 6.4 shows attributes such as a third-party-node-identity, remote roles or trusted entity roles, and local roles (based on existing trust contracts). Policies could be written to accommodate *Any* third-party-node-identity and/or trusted entity roles for policy scalability reasons. Policy obligations ensure that authorisation decisions are logged and managed as needed, e.g. for future auditing.

Third-party-node-identity enables a PDP to identify and make decisions on when a trusted third party is making requests for itself or for a third party. This is necessary for scenarios where a service provider restricts a trusted party (trust contract) from negotiating on-behalf of other parties or to restrict third-party access. A sample of a trust-contract policy that allows a remote Investigator to perform *select but not insert or update* on a PatientMaster table that is shown in **Extract 2 & 3 of Appendix A**.

6.6.3 Attribute Access Policies

Attribute Access policies capture the rules that govern access to local attributes. It describes who can invoke and assume local attributes and from where. The subject field of these policies include attributes for third-party-node-identities and roles from trusted parties agreed in a trust contract. However, since third-party-node-identities are unknown in advance, their subject attributes are generic. Local attributes (credentials) are resources to be protected which can only be invoked through satisfying trust contracts that have been agreed between trusted parties.

Table 6.5 shows an exemple of a tabular representation of information described by attribute access policies. Attribute access policy rule types can be enforced based on *Rule 4* (see

Subject	Resource	Action	Obligation
OX: OX.GP	GLA.GP	Invoke contract	logging
OX: OX.VOTES	GLA.VOTES	Invoke contract	logging
OX: D_{links}	GLA.VOTES/GP	Invoke contract	logging
Any: OX.GP	GLA.GP	Invoke contract	logging
Any: IMP.GP	GLA.GP	Invoke contract	logging

Table 6.5: Tabular View of Acceptance policies

Section 6.2). For example a *GLA* rule can be given as:

$$can\_invoke(OX.can\_activate(Any, OX.GP), local\_role) \leftarrow$$
$$tc(OX.GP) = local\_role,$$
$$local\_role = GLA.GP$$

A foreign entity, i.e. *OX* in this case, can invoke a contract for a local role if it satisfies the rule condition. *Any* signifies an unknown entity to *GLA* but someone who has a credential issued by *OX* asserting that it has an *OX.GP* role. Attribute access policy rules ensure that a foreign entity (domain) cannot participate in a negotiation if the foreign entity is not in a position to invoke any of the trust contracts.

It should be noted from Table 6.5 that a subject attribute called D_{links} represents the interpretation of linking delegation rule (discussed in Section 5.6). This is dynamically deduced based on satisfying trust contracts. For example, in Table 6.5, when a trusted party, e.g. *OX* satisfies trust contracts for *GP* and *VOTES*, shown as the first two rows in Table 6.5, it is implied that *GP* is a role in the *VOTES* trial and that *VOTES* is the linking attribute for the *GP* role. For example, an entity with credentials issued by *OX* that satisfies the first two rows in Table 6.5 is deduced to also satisfy the third row. This means that the entity can perform *select, insert, update* on the *PatientMaster* and perform *select* on *PatientDrug*, and can also by deduction perform *update* on *PatientDrug*.

6.6.4 Release Policies

Release policies determine whether negotiated local attributes can be released to a trusted party for the purpose of trust negotiation either for itself or for other parties. Release policies control what can be released and who it can be released to. Local attributes (credentials) are regarded as sensitive resources, and are protected by policies. Since release policies are mainly geared towards negotiations with trusted parties, and the subject attributes of these

policies can only contain trusted-node identities. Table 6.6 shows a tabular view of some release policies.

Subject 1	Subject 2	Resource	Action	Obligation
Any	NOTT	GLA.GP	Negotiate	logging
NOTT	IMP	GLA.VOTES/GP	Negotiate	logging
IMP	OX	GLA.Nurse	Negotiate	logging

Table 6.6: Tabular View of Release policies

This type of policy can be enforced based on *Rule 6* (Section 6.2):

$$
\begin{aligned}
&\mathit{can\_release\_cred}(OX, \mathit{can\_activate}(OX, \mathit{Nurse}), \mathit{IMP}) \leftarrow \\
&\mathit{has\_invoked}(\mathit{IMP}, \mathit{GLA.nurse}), \\
&OX \in \mathit{COT}(\mathit{GLA})
\end{aligned}
$$

The *GLA* rule says *GLA.nurse* credential can be released to *OX* domain during a negotiation session if Imperial college (*IMP*) has invoked the trust contract, $\mathit{has\_invoked}(\mathit{IMP}, \mathit{GLA.nurse})$.

6.7 Resolving Conflicts between Policies

It is not uncommon for conflicts to arise between policies. In DTN, two categories of conflict are identified. The first category is internal policy conflicts, which arise through conflicts between local and trust-contract policies. The second is multiple paths policy conflict, which exist when more than one trust negotiation from the same source node reaches a target node[6].

Internal conflicts can occur when trust-contract policies are created with no awareness of local policies. By definition, trust-contract policies are based on existing local policies, where the privileges granted to a role through a trust-contract are subsets of privileges granted to the same role in local policies. For instance, a *GP/Investigator* role described by a local policy in Table 6.3 can perform *select* on the *PatientDrug* resource. The same role described by a trust-contract policy in Table 6.4 has the same privileges. However, if a trust-contract policy exists that states that a *GP/Investigator* role can also perform *select, insert* on the *PatientDrug* resource, and a conflict will arise between the trust contract and the local policy. The conflict is triggered only when a remote user with *GP/Investigator* role wants to perform *insert* on the *PatientDrug* resource.

[6]Source and target nodes are defined in 7.1.

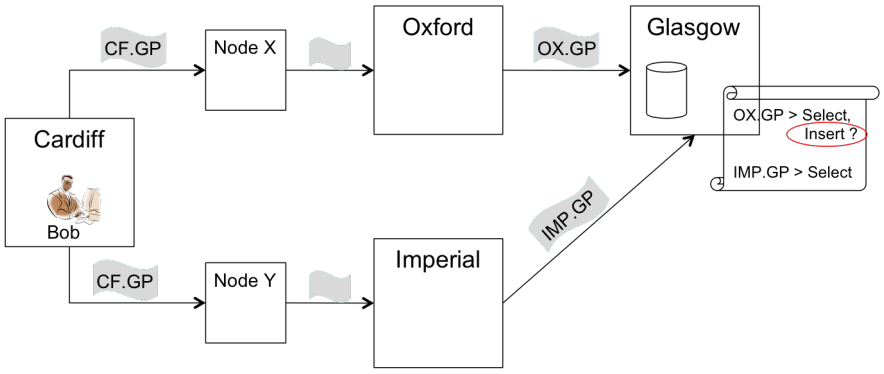


Figure 6.2: Multiple-path conflict

Multiple-path policy conflicts can occur when more than one trust-contract satisfy the same request. This type of conflict happens when a remote credential is negotiated through multiple end-points.

Example 6.1 *Bob who is a GP in the Cardiff node wants to insert data into Glasgow's PatientMaster resource. Since the Cardiff node is not directly trusted by Glasgow, they come under Any node in Glasgow's policies. The Cardiff node negotiates Bob's credential through intermediary nodes and finally through both Oxford and Imperial nodes. These negotiations enable Bob to present the IMP.GP and OX.GP (from the Imperial and Oxford node respectively) credentials to Glasgow (shown in Figure 6.2). Assuming Table 6.4 represents Glasgow's trust-contract policies, Bob from Any node, presenting OX.GP can perform select, insert on PatientMaster. Similarly, Bob presenting IMP.GP can perform select on PatientMaster. However, when Bob presents both credentials, which is not unusual, a conflict will arise between the two trust-contract policies (see rows 4 and 5 of table 6.4). The OX.GP implies he can perform select, insert on PatientMaster resource while GLA.GP implies he can only perform select. The question then is, should Bob be allowed to perform 'Select' or both 'Select and Insert'.*

In DTN, multiple-path conflicts are resolved by implementing role exclusion mechanisms as discussed in [48]. Defining non-hierarchical roles as mutually exclusive and applying exclusion rules such as run-time/complete exclusion, safety can be ensured. Local roles defined in trust-contract policies are mutually exclusive in order to ensure constraints that trust-contract policies offer are enforced. For instance, *OX.GP* and *IMP.GP* from *Any* node are mutually exclusive. XACML policy-combining algorithms [163] are implemented in DTN to resolve multiple path policy conflicts by combining conflicting trust-contract policies. Policy

combining algorithms such as the "deny overrides" algorithm returns a final *Deny* result if any of the policies that are being combined return a *Deny* result. In Example 6.1, if Glasgow decides to use the "deny overrides" algorithm as the combining algorithm, Bob would be denied from inserting data into *Patientmaster*, since *GLA.GP* would return *Deny* for the *insert* action on *PatientMaster*, regardless of the result of *OX.GP* that evaluates to *Permit*. **Extract 3 of Appendix 1** shows a PolicySet sample of how two mutually exclusive trust-contract policies are resolved.

On the other hand, DTN resolves internal policy conflicts by basing its final access decision on the evaluation of local policies. This is called local policy priority. One reason for this priority is that remote entities should not have more privileges than local entities. The second reason is that DTN is designed to complement existing access control infrastructures and policies[7]. For instance, using the internal conflict example, the policy decision engine will return a *deny* result for the *insert* action on *PatientDrug* resource.

There are also scenarios where both internal policy conflicts and multiple-path conflicts exist during a request (or user) session. The order of resolution is to first resolve all multiple-path conflicts and then resolve any internal policy conflicts that are still outstanding.

6.8 Summary

The syntax and semantics of the policy language used in DTN were specified in this chapter. The policy language, a rule-based policy language was discussed. The rule-based language uses propositional logic for rules definition. It was shown how a rule consists of a rule head and rule body. A rule body contains conditions and if these conditions are satisfied, the rule will be enforced. With this, a rule body can be used to encode parts of an authorisation policy and the rule head for encoding what should be enforced by the policy.

The syntax of the rule-based policy language was defined and specified. Using the specification, six policy language functions or predicates were defined for DTN. These functions express the various authorisation rules that can be used to enforce DTN policies. The chapter described how policies defined for DTN include: local, trust contact, attribute access and release policies. Through these policies it was shown that the potential risks associated with the notion of transitive trust can be reduced if properly designed and implemented. It was also described how DTN policies could be used to ensure that trust contracts, which themselves are sensitive, could be protected with policies and thus used to control the invocation

[7] These policies are referred to as local policies in DTN.

of trust contracts. By using such policies, it was shown how trust contract privileges could be constrained based on who is invoking the contract and for whom it is being invoked.

The chapter concluded with a discussion on how conflicts between policies could be resolved. Two types of conflict were identified. An internal policy conflict, which arises when the evaluation of trust-contract policies disagree with the evaluation of local policies. This conflict is resolved by given priority to the evaluation of local policies. It was shown that this is valid since trust-contract policies are supposed to be derived from local policies. The second conflict identified was based on multiple paths policy conflict. This arises when more than one trust negotiation from the same source node reaches a target node. It was shown how this could be resolved by ensuring mutual exclusivity between trust contracts and by the use of policy-combining algorithms.

7 DTN Protocols

This chapter describes details of two protocols used in the DTN framework: the Discovery and Negotiation protocols. These protocols describe how trust pathways are discovered in a virtual network and how credentials (trust) are negotiated across the network.

7.1 Node Classification

In order to understand the protocols, nodes are classified differently based on the role(s) they play in the network. The various nodes classification are shown in Figure 7.1 and described below:

- **Source node:** A node is a *Source* node if the node was the one that initiated a message request. A message could be a request or a response but a *Source* node is the originator of a route request or a negotiation request.
- **Target node:** A node is a *Target* node if the node is the resource provider or the end point for a message request. It is also referred to as a sink node.
- **Intermediary node:** An *Intermediary* node is a third party node that acts as a gateway that relays a message between a *Source* node and a *Target* node or another intermediary node.

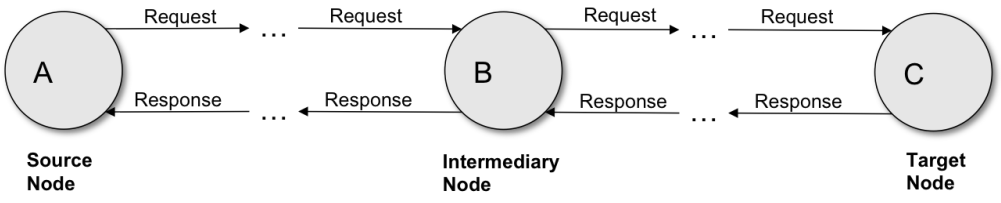


Figure 7.1: Classification of nodes

In addition to this classification, a Sender node is a node from whom a message (request or response) has been received. A Sender node is not necessarily a *Source* node, but a *Source* node is also a Sender node when it sends a request. An Intermediary node is viewed as a Sender node since it relays requests or responses to other nodes. Similarly, a *Target* node is viewed as a Sender node when it sends a response. Fundamentally, each node is a sender node whenever it sends or relays a message.

7.2 Discovery

Discovery is the process of selecting paths in a network of connected nodes. Nodes are connected based on trust, i.e. nodes that trust each other are said to be connected. Through discovery, nodes can select trusted paths to a non-trusted node. These paths are called trust pathways or form what is known as circle of trust. A cicle of trust is an ordered list of nodes that possess credentials that may be used for trust negotiation. These nodes constitute a path (trust-path) between two points in a network. A path between two points can be discovered by traversing nodes that are connected through overlapping *COTs*.

Trust pathways enable nodes to negotiate attributes from point to point. This discovery process of DTN is similar to network routing. In DTN, the objectives of the discovery process include:

1. to identify paths that may exist between nodes through which trust negotiation can occur;
2. to limit route requests to nodes that exist in their COT;
3. to be dynamic and respond to changes as nodes join or leave the network, i.e. as trust between nodes changes;
4. to be responsive to changes in network conditions and be able to scale up in meeting network demands.
5. to be secure, offering point-to-point and end-to-end security in the network.

Discovery enables each node to construct next-hop trust (routing) table(s) automatically, based on the information carried by the discovery protocol. The discovery protocol describes how nodes (or domains) are discovered based on available trust contracts in a network of nodes. The protocol uses messages that are packaged as request and response elements, and

rules that a network must follow when producing and consuming these elements. In the implementation of DTN, these messages are bound to SOAP [66] and subsequently propagated to trusted nodes as described in Section 7.2.1.

7.2.1 Discovery Messages

Underlying DTN is a generic message structure that abstracts the basic elements of a request and response. The elements of a message include:

- **Source:** The Source of a discovery message is represented by its X.500 Distinguished Name (DN). All nodes (domains) must have a non-empty DN and it is up to other domains to verify the source DN to the Public Key Certificate (PKC) of the source. The DN uniquely identifies a node (domain) that initiated the path discovery. It is based on the subject name of a digital certificate. An example of a source DN is:
  /C=UK/O=eScience/OU=Glasgow/L=Compserv
  /CN=**host/dhaulagiri.nesc.gla.ac.uk**
  /emailAddress=ajayio@dcs.gla.ac.uk
  where host/dhaulagiri.nesc.gla.ac.uk is a placeholder for the service.
- **Target:** A distinguished name (DN) that uniquely identifies the domain (service Provider) that is to be discovered. It is expressed as the subject name from its corresponding PKC. This binding can be used to establish an authenticated security context in which messages can be protected.
- **SeqNum:** A sequence number that is unique and maintained by the *Source* node for each message.
- **MAC:** Message Authentication Code generated using a keyed hash algorithm [168]. The *MAC* which is OPTIONAL and if used covers the whole message.
- **Type:** Identifies the message *Type*, which can be a request or a response.

The request and response messages can be likened to input and output messages as shown in Figure 7.1.

Route Request: A route request is a message (query) sent by a node (Sender node) to its trusted nodes for the discovery of trust paths. These trusted nodes are nodes in the Sender's circle of trust *(COT)* as shown in Figure 7.2. A request message identifies the *Source* node that initiated the route request, the *Target* node (SP) that is to be negotiated with, the

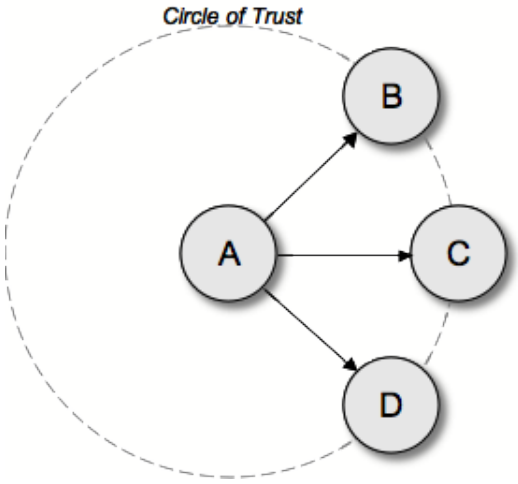


Figure 7.2: A Sender's COT

sequence number that uniquely identifies the message and a message authentication code that can be verified between a given node-pair.

The sequence number is different for each route request initiated by a *Source* node but does not change as a route request is being propagated through intermediary nodes. Different sequence numbers help a *Source* node to distinguished between route queries. With every message sent, a time-stamp is associated with the message. This helps intermediary nodes to distinguish between previously and newly sent messages. Sequence numbers also help to identify loops and prevent cycles.

The *MAC* covers the whole message and is generated through a keyed hash algorithm that uses a shared key or a signed hash value using the sender's private key. The shared key must be agreed between two nodes as part of their trust agreement. The shared or private key used is meant to deter external modification. However, if the message communication is over an authenticated and secure channel then a *MAC* is optional. Having a *MAC* does not prevent internal attacks, where a sequence number could be modified to potentially destroy routing information of trust relationships.

Route Response: For every query a response is expected. A route response is a message sent as a reply to a route query. Even though a route request may be propagated to other intermediary nodes, only one reply is sent back as shown in Figure 7.3. The route response is a message *Type*, which identifies the *Source* node that initiated the route request, the *Target*

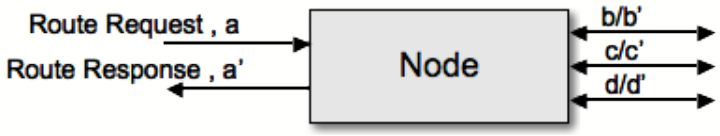


Figure 7.3: Discovery messages: Single route response for multiple route requests

node (SP) that is to be negotiated with, the sequence number that uniquely identifies the message and message authentication code that can be verified between a given node-pair.

The sequence number must be the same sequence number as in the original request otherwise the response may be associated with another request. If a response cannot be associated with a request, the response is invalid. As with received route requests, a time-stamp is associated with the message. This helps intermediary nodes to distinguish an already received response from another. Similarly, sequence numbers also help to identify loops and prevent cycles.

Like route requests, a *MAC* is associated with every route response. It is optional if the response is communicated over a secure channel.

7.2.2 Key Management

Key management is an important part of the DTN protocol. In situations where a *MAC* is used, a shared key must be agreed between two parties. This is usually agreed when trust-contracts are agreed. Periodically, a shared key is renewed and both parties make use of this new key. In other environments where parties have public and private key-pairs, and trust the same CAs or Root CAs, then a hash value based on an agreed hash algorithm can be digitally signed using private keys.

7.2.3 Discovery Process

In DTN a modified Ad-hoc On Demand Vector (AODV) [169, 170] routing protocol is used to build chains of trusts or discover trust-pathways which allow credentials to be negotiated. Since DTN is not about shortest paths to a destination, the algorithm used in the protocol is modified to support discovery of multiple paths to a destination. Similarly, as notification messages hold sensitive information, notifications are restricted to trusted peers and messages are encrypted with shared keys or key pairs. Once each node collates routing information,

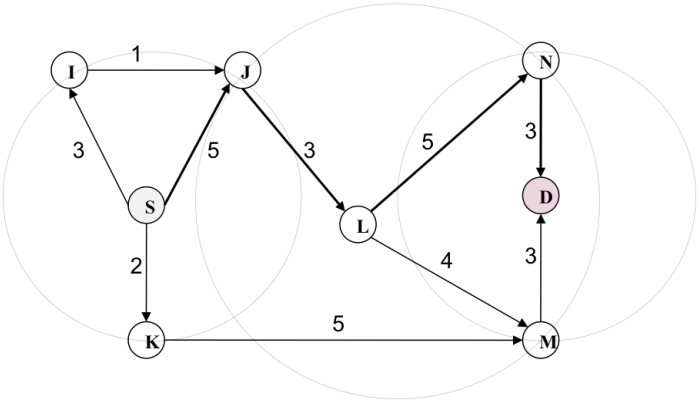


Figure 7.4: Discovery process, using Circles of Trust

nodes judiciously select appropriate nodes with respect to destination nodes. The discovery algorithm is listed in Algorithm 7.1.

Each node u keeps a list of nodes $v_0, v_1, ..., v_k \in V$ in its *circle of trust (COT)* along with their respective weights, encryption keys and rule-set also known as constraints. When an entity at a node S in Figure 7.4, makes a request for remote resources[1], say a list of cancer patients for a given trial; a provider service may suggest that the relevant data sets exist at nodes I, J and D. Node S will then check for nodes I, J, D in its COT and sends route requests to nodes that are in its COT, i.e. nodes I and J. It uses the discovery process to discover nodes that are not in its COT, i.e. node D.

Since node D is not in S's COT, a route request is initiated. A route request is initiated to discover trust-pathways to other nodes if such information does not already exist in the node's trust-pathways table. The trust-pathways table contains lists of COT nodes that act as 'next hops' or routes to *non-COT* nodes. The list is prioritised based on weights of those routes as described in Section 5.5.

Route Request: Consider Figure 7.4 where S is the *Source* node and D is the *Destination* or *Target* node S sends a route request (RREQ) to nodes that exist in its COT. Node S contains nodes I, J and K in its COT and they will all receive a RREQ. A typical RREQ has the following structure: a source distinguished name; a destination distinguished name; a sequence number and a message authentication code (MAC) which is computed using a shared key, i.e. a RREQ is given as:

[1] It is assumed that a discovery service for service providers exists that returns a list of service providers that provide various resources

Algorithm 7.1 DTN Discovery Protocol – when a message (RREQ or RREP) is received

```
validate MAC
if msg is RREQ then
    if RREQ had not been previously received from sender then
        save a copy of the RREQ and timestamp
        if this node is the target then
            send RREP to RREQ's sender
        else
            send RREQ to nodes in COT
        end if
    end if
else {msg is RREP}
    if RREP had not been previously received from sender then
        save a copy of the RREP and timestamp
        update routing table
        if this node is not the source node then
            find RREP's request senders i.e. match (source, target, seqnum)
            send RREP to nodes that sent request
        end if
    end if
end if
```

$$RREQ : \{DN_s, DN_d, Seqnum\} + K_{sd}(MAC)$$

DN_s and DN_d are distinguished names for *Source* and *Destination* nodes respectively. *Seqnum* represents a sequence number. $K_{sd}(MAC)$ is a message authentication code computed using K_{sd}, which is a key shared between the sender and receiver nodes.

Each node implements a RREQ table that stores route requests it receives from *Source* nodes and intermediary nodes. When a non-destination node receives a RREQ, it checks its COT to see if the sender is a trusted node. It reads the MAC using their shared key and forwards the message to other trusted nodes, re-computing the MAC using keys it shares with those nodes. It then stores that node as the 'next hop' to the *Source* node in its trust-pathways table. Similarly, if it receives the same RREQ from multiple nodes, it stores those nodes as the 'next hops' to the *Source* node in its trust-pathways table.

Route Response: The destination node DN_d, on receiving a RREQ, creates a route response (RREP) which is sent to nodes it receives a RREQ from. A RREP is given as:

$$RREP : \{DN_d, DN_s, Seqnum\} + K_{ds}(MAC)$$

Where DN_d and DN_s are distinguished names for the destination and *Source* nodes respectively. The sequence number *Seqnum* is maintained by the destination node for each reply made to a *Source* node. $K_{ds}(MAC)$ is a message authentication code computed using K_{ds}, which is a key shared between two nodes, i.e. the sender node and receiver node.

When a non-*Source* node receives a RREP, it checks its COT for the sender, checks for the corresponding RREQ in its RREQ table and if valid, verifies the MAC using its key. For every valid MAC, a node updates its trust-pathways table registering the RREP sender node as the 'next-hop' to the destination node. The non-*Source* node also re-broadcasts the RREP to other nodes it receives a RREQ from but re-computes the MACs using keys it shares with those nodes.

Route Update: A node may revoke its trust contracts with other nodes and thus render some routes in the trust chains invalid. Similarly, new nodes may be added at any time and new trust relations added. Thus when a node detects broken routes it sends error messages to other nodes that are in its COT. An error message contains a MAC, which is computed using shared keys. When a node receives route errors, it authenticates the sender and verifies the MAC using their shared key. If the MAC is valid, it updates its trust-pathways table, and if other routes to a destination do not exist, it re-sends error messages to other nodes in its COT.

7.2.4 Routing Algorithm

The algorithm applied to support DTN was based on a number of factors: route discovery, confidentiality, maintainability and cost (i.e. number of trust contracts). Various routing algorithms were compared and some of these are discussed below.

Distance Vector algorithm (DV): This algorithm described in [171] assigns a cost to each of the links between nodes in the network. Cost in DV is the distance or the number of hops between nodes. Cost can be evaluated based on message delay, reliability and hop count. Nodes send information from one point to the other via the path that results in the lowest total cost, i.e. the sum of the costs of links between the nodes. In DTN, costs are viewed as weights, i.e. the number of TCs that exist between two nodes. The algorithm works when each node sends to each neighbouring node its own current idea of the total cost to get to all the destinations it knows. Neighbouring nodes compare this information to what they already have in their routing table, and, based on the lowest cost, they update their own

routing tables. After a while, all nodes in the network will discover the best route and hence next-hop nodes for all destinations.

When a node becomes unavailable, the network corrects itself when adjacent nodes drop the unavailable node from their routing tables and pass on their new routing table to their neighbouring nodes. This process is repeated from each node to their adjacent nodes. Eventually all the nodes in the network have updated their routing tables based on the new information. However, the DV algorithm has several drawbacks:

- It performs poorly as the number of hops increases due to the time it takes converge, i.e. time to propagate changes across the entire network;
- It does not scale well due to the fact that information shared between nodes are often inaccurate, incomplete or obsolete. It causes the count-to-infinity problem discussed in [172][2];
- It has less computational complexity and message overhead compared to other algorithms, since it requires more resources to evaluate a route;
- It raises privacy issues in trust-based networks as nodes might not want to disclose the nodes they currently trust to other non-trusted nodes.

Link-State algorithm (LS): [173] describes a LS algorithm that enables each node to build a map of the entire network in the form of a graph. It works when each node floods the entire network with information about nodes it can connect to, and each node uses this information to create a map of the network. With this map, each node calculates the least-cost path from itself to other nodes. This enables each node to construct a tree with itself as the root and a given path through the tree is the least cost path to any other nodes. Routing tables are based on this tree. LS algorithms also have several drawbacks

- They create heavy traffic because of flooding;
- They need a considerable amount of computational resources (e.g. storage and processing unit) to calculate routing tables since it has to build a map of the entire network;
- They assume all nodes can be reached, which is not always the case in trust-based networks – where reachability is for a selected few;

[2]This is not an issue for DTN since its aim is to discover multiple paths and not the lowest cost for reaching a node. It is noted that it might be important to prioritise the path to a node based on the total cost (number of TCs) of reaching a node

- They raise privacy issues similar to the DV algorithm as nodes might not want to disclose the trust relationships they have with other nodes.

Ad-hoc On-Demand Vector algorithm (AODV): Unlike the distance vector algorithm, AODV [170] is not particularly interested in the shortest route to a destination node but rather in all the possible routes to a destination node from a *Source* node. As the name suggests, the algorithm is widely used in ad-hoc networks such as mobile ad-hoc networks (MANET). The algorithm is suitable for a dynamic self-starting network with little or no dependence on adverts from nodes in the network. AODV works by broadcasting route discovery packets to neighbouring nodes when necessary; it detects nodes in its neighbourhood (i.e. reachable nodes), and also shares information about changes in local connectivity to other neighbourhood nodes that might need the information. To maintain the most recent routing information, nodes use a monotonically increasing sequence counter to identify obsolete cached routes. With respect to DTN, AODV has the following benefits:

- It scores better on privacy as the algorithm only discloses routing information to its trusted partners;
- The cost of maintaining routing information is acceptable since the algorithm offers resilience owing to its discovery of multiple paths;
- It is well suited to discovering possible routes since its objective is to discover multiple paths rather than the best path.

Distributed Hash Tables (DHT): DHT are not considered as the objective is not to find a *Target* node or resource, but rather in finding all nodes that are in a relationship with a given *Target* node. Similarly, DHT raises privacy issues since it requires nodes to disclose the nodes they currently trust to other non-trusted nodes.

7.3 Negotiation

The negotiation protocol describes how negotiation elements are embedded as packages within negotiation request and response messages. It also describes the processing rules that a negotiator must follow when producing or consuming these messages. The protocol is based on a request-response protocol.

Typically the negotiation process involves an entity through its local-domain, e.g. a *Source* node, starts a session with a service provider, e.g. a *Target* node requests a service or resource. A user handle (token) and a resource handle (token) are exchanged. On receipt of a resource token, the local-domain initiates the trust negotiation process with nodes in its *routing table* that act as a next-hop to the service provider. If no routes exist, the local-domain initiates a discovery process with all nodes in its *COT*. After the discovery process, negotiation requests are sent to next-hop nodes and a response is expected until the negotiation window closes. Every node that receives a negotiation request is expected to send back a response.

7.3.1 Negotiation Messages

A negotiation message is a generic message that abstracts the basic elements of a negotiation request and response. The elements of a negotiation message follow the same structure as discovery messages, i.e. *Source*, *Target*, *SeqNum*, *MAC* and *Type* but also includes elements:

- **RCred:** Credentials presented as attribute certificates or security tokens. These are the credentials that a peer wants to negotiate.
- **Token:** A resource token is created by the service provider, to identify a request and for linking the request to a user token. It is provided by the service provider during a resource-request handshake and only remains valid throughout the resource-request session.

Negotiation Request: A negotiation request is sent as a message from a user's local-domain on behalf of a user. The request includes entity credentials (depending on the negotiation strategy [120]); resource tokens provided by a service provider that are used during the session to undertake the handshake; a sequence number that uniquely identifies the message, and a message authentication code for ensuring message integrity.

From a *Source* node perspective, credentials included in a message are the credentials of a user that request services from a service provider. Credentials are made available for negotiation if the local-domain's release policy does not prevent their disclosure. The trusted domain or *Intermediary* domain (next-hop node) on validating the message, will subject these credentials to its acceptance policy. The sequence number is different for each negotiation request sent to trusted domains. It is noted that the resource token remains the same for each negotiation request since it binds negotiation requests to a resource-request session. Different sequence numbers help a *Source* node to distinguish negotiations between different

negotiation paths. A time-stamp is associated with every request sent. This is provided with the sequence number that can help prevent or detect some external attacks.

An *Intermediary* node upon receiving a negotiation request, validates the message; subjects the request credential to its acceptance policies and determines potential delegated local credentials based on *trust-contracts* that it holds with the requester and service provider and/or other intermediaries. These delegated local credentials are similarly subjected to release policies. If release policies are satisfied, a negotiation request with these delegated local credentials is created and sent to other trusted nodes (next-hop nodes). The resource token and sequence number remain unchanged during the received negotiation request. Any modification to the resource token and sequence number breaks the chain-of-trust for that particular negotiation. As before, the MAC and time-stamp are associated with every request sent.

A *Target* node or service provider receives a negotiation request; validates the message; subjects the request credential to its acceptance policies, and determines delegated local credentials based on *trust-contracts* that it holds with the requester. The service provider populates its negotiated attribute store (NAS), which is described later in Section 4.2.5 with the delegated local credentials and resource token. A time-stamp is also saved with each entry in the store. The resource token links the negotiated attributes to a resource-request session.

Negotiation Response: A negotiation response is a message sent as a reply to a negotiation request. It extends the negotiation request message with an additional message element:

- **Status:** A binary value, 1 or 0. It contains a 1 if at least one local credential was successfully delegated by the *Target* node (SP). Otherwise, it contains a 0 indicating that negotiation was not successful either at the *Target* node or at any *Intermediary* node.

A *Target* node populates its NAS when a negotiation is successful and it sends a response with a '1' status value. If delegated local credentials were not available for a negotiation request, a negotiation response with a '0' status is returned. The credential in the negotiation request is also included in the response. The sequence number and resource token also remain the same as the negotiation request. This enables the requester to differentiate between requests it has sent.

When an *Intermediary* node receives a negotiation request, it sends back a response with a '0' status if the credential in the negotiation request fails either its acceptance policies or release

policy, or both. Credentials in the negotiation request are also included in the response. The sequence number and resource token are unchanged from the received negotiation request. This enables the requester to differentiate between the requests it has sent.

In addition, when a negotiation response is received and validated, negotiation responses are sent to all nodes that it has received *corresponding* negotiation requests from. The status value in each of these responses is the same as the received negotiation response. Credentials from corresponding requests are sent in these responses. The sequence number and resource token are the same as that of the received negotiation response. This enables corresponding request senders to differentiate between requests they have sent.

7.3.2 Key Management

As described in Section 7.2.2, a shared key has to be agreed between parties. Though a shared key could be used for both discovery and negotiation, using different keys is preferred since the compromise of one key would not affect the other. This does not extend to situations where hash values are signed using private keys, however.

7.3.3 Negotiation Process

Once trust paths or *next-hop* information exists in the routing tables, domains can prioritise *next-hop* values based on the *number of trust-contracts* they have with these *next-hop* nodes. The more *trust-contracts* the better their chance of successful negotiations, since increase in trust contracts increases the likelihood of mapping between credentials. On the other hand, more *trust contracts* do not guarantee that negotiations will be successful.

The algorithm used by the negotiation protocol is listed in Algorithm 7.2. It shows what happens when a node receives a negotiation message, which is either a request or a response. The message states of the negotiation protocol are shown in table 7.1.

When a *Source* node receives a negotiation response (NREP), it validates the message and checks the response against any negotiation request (NREQ) it may have sent. The *Source* node then waits for more negotiation responses or a negotiation time-out before proceeding to the Service Provider with the user's token (or handle) and resource token. The return of both tokens (with an attribute assertion in the case of Shibboleth) suggest to the SP that trust negotiations are complete for the requested session and that it can proceed to make

Message state	Source	Target	SeqNum	MAC	Token
NREQ $_{\text{sent}}$: default state	1	1	1	1	1
$\text{NREP}_{\text{sent}}$: fail state	1	1	1	1	1
$\text{NREP}_{\text{recv}}$: default state	1	1	1	1	1
$\text{NREQ}_{\text{recv}}$: fail state	1	1	1	1	1
$\text{NREQ}_{\text{recv}}$: pass state	1	1	1	1	1
$\text{NREP}_{\text{sent}}$: pass state	1	1	1	1	1
$\text{NREP}_{\text{sent}}$: forward state	1	1	1	1	1
	Rcred	**Status**	**Lcred**	**Sender**	**Next-hop**
NREQ $_{\text{sent}}$: default state	-1	-1	1	1_{LO}	1
$\text{NREP}_{\text{sent}}$: fail state	1_S	0	-1	1	-1
$\text{NREP}_{\text{recv}}$: default state	1	1=	-1	1	-1
$\text{NREQ}_{\text{recv}}$: fail state	1	-1	-1	1	-1
$\text{NREQ}_{\text{recv}}$: pass state	1	-1	1	1	-1
$\text{NREP}_{\text{sent}}$: pass state	1_S	1	-1	1	-1
$\text{NREP}_{\text{sent}}$: forward state	1_{OS}	1=	-1	1	-1

Legend:

1	Yes or Succeed
0	No or Fail
-1	Not Applicable
1=	No change in reply
1_S	Sender's credential
1_{OS}	Original sender's credential
1_{LO}	This node i.e. Self

Table 7.1: Data States for Negotiation Messages

resource policy decisions based on credentials stored in its NAS. The Shibboleth-oriented flow diagram of a negotiation process is shown in Figure 7.5.

The negotiation time-out is based on a negotiation window. All nodes involved in trust negotiation are expected to individually set the value of their negotiation window based on a metric of their choosing. In the current implementation, the value is set to 90 seconds based on the performance tests described in Section 9.8. This value is based on the sum of the maximum run time values of both discovery and negotiation. The consideration for the maximum run time value of the discovery process is due to the fact that node discovery may be required prior to a trust negotiation. This value can subsequently be changed, e.g. based on experience. In most cases, a negotiation response is received before a negotiation time-out occurs. The negotiation window helps to prevent replay attacks (discussed in 7.4.1) and for requesting session closing.

7.4 Attacks to Protocols

Different attacks exist that can alter data or gain authentication or authorisation privileges by inserting false information. The attacks described in this section are general to the AODV routing protocols and are either externally or internally instigated. External attacks come from entities, e.g. nodes, that do not belong to the network. Internal attacks come from nodes that are compromised in the network.

7.4.1 External Attacks

The ways in which an external entity could attack a routing protocol are numerous and include:

- **Replay attack:** An attacker could collect routing information such as route request/ response and later propagate stale messages, i.e. messages with sequence numbers that have already been received or have timed out. If these stale messages are not detected and dropped, incorrect routing information could be made. This attack could be detected by monitoring sequence numbers and checking them against a list of sequence numbers that have been received or known to have expired.
- **Denial of Service (DoS) attack:** A malicious node may attempt to bring down a network by saturating the network with false routing messages, which could eat up network resources, e.g. CPU cycles and memory. This would prevent legitimate routing messages from being delivered and thus make the gathering of routing information difficult. Similarly an attacker can broadcast false route error messages stating that some nodes are no longer available thus preventing access to those nodes. This attacker could be prevented or detected by applying well known DoS solution techniques such as filtering or intrusion detection techniques [174, 175, 23]. For instance, a node that is known to be compromised could be backlisted and all messages from the node could be dropped, until the node's new public-key and credentials have been re-evaluated.
- **Modification:** An attacker can modify message elements like sequence numbers to cause traffic redirection in a network. If integrity measures are taken such as the use of a *MAC* or signed hash value, then these modified messages could be detected and dropped. However, this does not prevent a DoS attack.
- **Spoofing:** A malicious node can impersonate other nodes through an identification

spoofing attack and sending bogus routing messages. The bogus messages, if not detected, can cause inconsistent routing tables or false routing entries. This attack can be prevented or detected when nodes manage and protect their secret keys as described in Section 7.2.2.

7.4.2 Internal Attacks

Internal attacks are more difficult to detect and are more potent in causing service disruptions or inconsistent routing information. An internal attack can take over compromised nodes and even access private keys or shared keys with other nodes. This attack could generate valid *MAC*s for modified or false routing messages.

An internal attack can be limited where a *Source* node digitally signs part of a message: {*Source*, *Target*, *SeqNum*, *Token*} thus preventing internal modifications. However, this would require a key management model for all nodes in the VO (or network), using a hierarchical approach such as a centralised CA or a *web of trust* approach such as OpenPGP [176].

7.5 Summary

In this chapter the details of the discovery and negotiation protocols used in the DTN framework were discussed. The chapter described how trust pathways could be discovered in a virtual network and how credentials (trust) could subsequently be negotiated across the network.

It described how the discovery protocol allows domains that form trust-pathways to be discovered through circles of trust and trust contracts. An outline of how the protocol uses request and response messages, and rules that a network must follow when producing and consuming the messages was given. The chapter also described how the discovery protocol made use of the AODV routing protocol. Since DTN is not about shortest paths to a destination, it was explained how the AODV protocol was modified to support discovery of multiple paths to a given destination. The chapter described how the messages exchanged using the protocol were restricted to trusted nodes and how messages could be encrypted with shared keys or key pairs.

The chapter also described how the negotiation protocol supports negotiation elements that

could be exchanged using negotiation request and response messages. It discussed the protocol and processing rules that a negotiator must follow when producing or consuming negotiation messages. It also presented the algorithm used by the negotiation protocol along with the protocol's message states.

The chapter concluded with a discussion of how the protocols can be protected from various security attacks. The security attacks discussed included replay, denial of service, modification and spoofing attacks. It described how DTN detects and prevents replay attacks by employing monitoring of message sequence numbers and checking them against a list of known numbers. The chapter discussed how denial of service attacks could be prevented or detected by using filtering and intrusion detection techniques. The chapter also discussed how modification and spoofing attacks could be prevented and detected by combining key management and cryptographic techniques.

Algorithm 7.2 DTN Negotiation Protocol – when a message (NREQ or NREP) is received

```
1: validate Mat
2: if msg is NREQ then
3:    if NREQ had not been previously received then
4:       if this node is the target then
5:          if acceptance policy returns true then
6:             save NREQ_recv
7:             send NREP to sender
8:             save NREP_sent
9:          else {failed acceptance policy}
10:            save NREQ_recv
11:            send NREP to sender
12:            save NREP_sent
13:         end if
14:      else if this node is an intermediary then
15:         if acceptance policy returns true then
16:            save NREQ_recv
17:            if release policy returns true then
18:               send NREQ to next_hop nodes OR forward corresponding replies that was received
                  from next_hop nodes
19:               save each of NREQ_sent
20:            else {failed released policy}
21:               send NREP to sender
22:               save NREP_sent
23:            end if
24:         else {failed acceptance policy}
25:            save NREQ_recv
26:            send NREP to sender
27:            save NREP_sent
28:         end if
29:      end if
30:   end if
31: else {msg is NREP}
32:   if NREP has not been previous received then
33:      if this node is the source then
34:         save NREP_recv
35:         wait for other replies
36:         on time out or after all replies, resume SAML (get resource from SP)
37:      end if
38:      if this node is not target then
39:         save NREP_recv
40:         send NREP to nodes that sent request, match (source, target, seqnum, token, request_cred)
41:         save all NREP_sent
42:      end if
43:   end if
44: end if
```

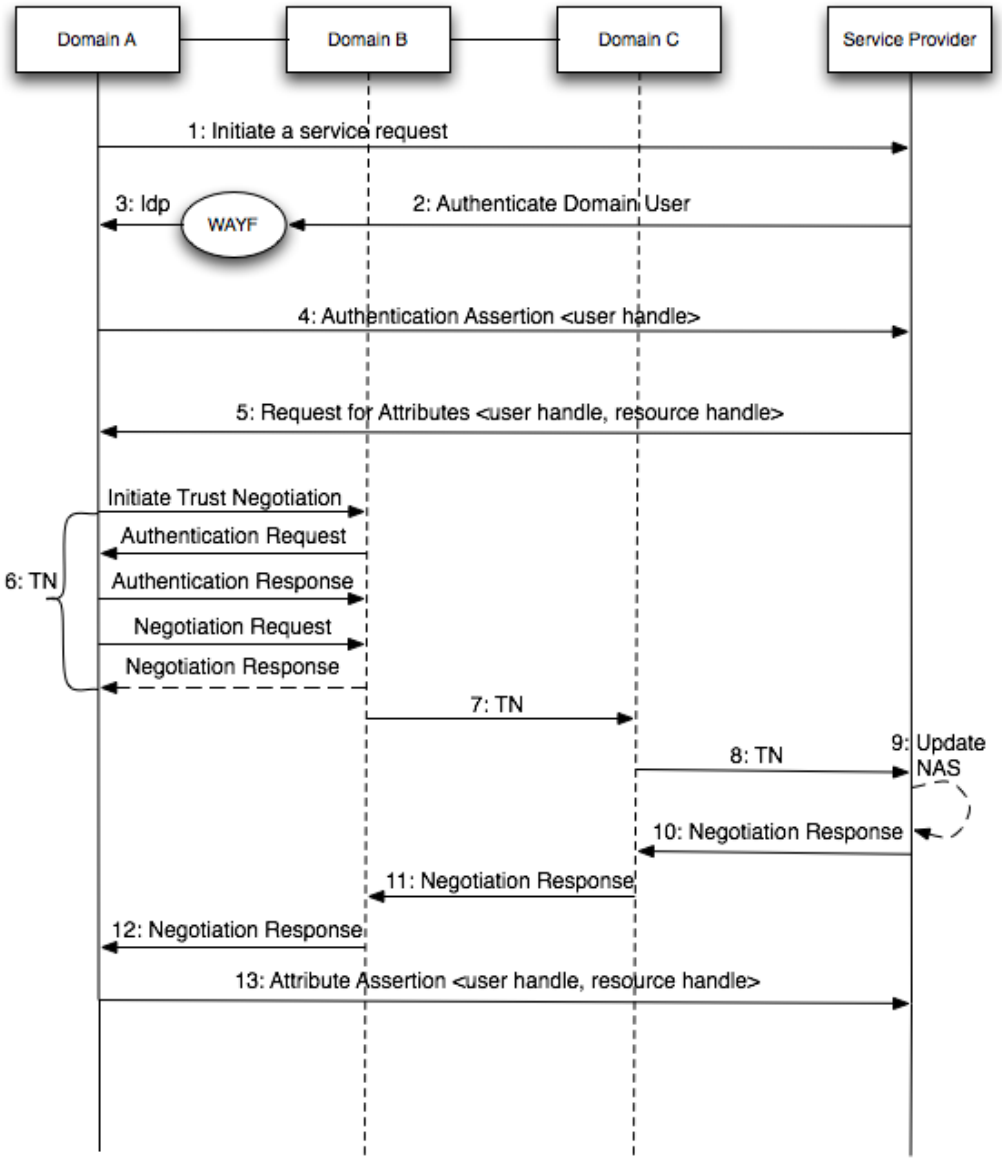


Figure 7.5: Trust Negotiation Flow for VOs

8 Case Study: Clinical Trials

This chapter presents clinical trials and scenarios that serve as the basis for testing the DTN framework described in this thesis. The chapter starts with an overview of the e-Health domain and then focuses on an area of the domain – clinical trials.

8.1 Overview

The e-Health domain is perhaps one of the most interesting domain where data ownership and access control management is very acute and of concern to the health community. At present there is an ongoing drive for health researchers to collaborate and access data across institutional boundaries for improved drug administration [177], disease diagnosis, epidemiological studies [178] and healthcare delivery. e-Health researchers need to share and securely access data resources that are geographically distributed, owned and controlled by various organisations.

Two key areas of e-Health research include epidemiological studies and clinical trials. An epidemiological study is a descriptive, analytic and/or experimental study typically identifying risk factors for diseases and for determining optimal treatment approaches in clinical medicine. Clinical trials are research studies in which new methods for diagnosing, treating, detecting and preventing diseases in people are investigated. Both studies require the collection of data typically on many hundreds or thousands of people over a long period of time (see Appendix B.2), and from different sources and using numerous devices[1]. Usually these studies require collaborative effort from individuals, GPs, clinicians, health boards and hospitals, among others, and they require a supporting e-Infrastructures.

Some of the issues that affect clinical trials include access to and/or performing analysis on patient records hosted on different sites and identifying eligible participants in ethically driven frameworks, e.g. addressing patient consent. Clinical trials also include the collection

[1] As an example, the UK BioBank is a study which is currently recruiting 500,000 people in the UK, who will be followed up over 25 years [147].

and aggregation of data generated during studies in many different forms. Similarly, data transmission is an issue for data collected during a study is often at mobile health centres and is substantial in total size. Other issues include portability for remote data entry systems that are needed for large-scale studies, data processing, and study management. Last, but not the least, are issues of data protection and access control, privacy, confidentiality, data integrity, data ownership management, provenance, aggregation and inference, interoperability and heterogeneity.

8.2 Clinical Trials

Clinical trials are studies carried out in various magnitudes of scale, i.e. study length, size and time as summarised in Appendix B.2, to test and validate new treatments and health devices [179, 180]. Generally, clinical trials can be summarised into these areas: patient recruitment, data management, study administration and co-ordination. Typically, before a trial can start, willing participants have to be identified, their eligibility for the study evaluated and their consent obtained. In most trials, targeted populations are commonly represented so as to cover all possibilities for the medication. Clinical trials are managed and executed in phases [179] e.g. for a drug to be approved by the Food and Drug Administration (FDA) [181], it must undergo clinical trials. Clinical trials phases are summarised in Appendix B.2.

Two clinical trial examples are presented in this section. These scenarios are taken from completed studies and are presented with little medical jargon for clarity. These scenarios highlight some typical processes, issues and requirements of clinical trials. The second scenario is a collaborative study, which shows the impact of collaboration on study management. In addition, questions raised by these scenarios are presented. The clinical trial, controls and terms used in this section are defined in Appendix B.

8.2.1 The West of Scotland Coronary Prevention Study

The West of Scotland Coronary Prevention Study (WOSCOPS) was a clinical trial that investigated pravastatin in a primary prevention context [182, 183]. The randomised, double-blind, placebo-controlled trial tested the hypothesis that the use of 40mg pravastatin each night, over an average five year period would reduce coronary morbidity and mortality in 45-64 year old men who had raised plasma cholesterol levels. The study's principal endpoints were coronary heart disease deaths, in addition to non-fatal myocardial infarction; coronary

heart disease death, and non-fatal myocardial infarction.

WOSCOPS Screening and Recruitment: About 160,000 men were invited to attend clinics to assess their coronary risk factors. Approximately 81,000 men in the West of Scotland came for the first visit and those with total cholesterol level of at least 6.5mmol/L were given lipid-lowering dietary advice and invited to a second screening 4 weeks later. 20,912 men came for the second visit and those with cholesterol level of at least 4.0mmol/L were advised to further stay on lipid-lowering diet and invited for a third visit. 13,654 men attended, and were screened and invited to a fourth visit. On the fourth visit, men with conditions of $\geq$ 4.5mmol/L and $\leq$ 6.0mmol/L on the last two visits were randomised [184]. About 6,595 men with raised plasma cholesterol levels who gave written informed consent were finally recruited to the trial.

WOSCOPS Functional Units: Random blocks of randomised participants were allocated to various health centres. Participants were followed up in each health centre. A trial centre team included a physician, a nurse and an administrator responsible for an average of 400 participants [183]. Participants were seen at an average interval of three months and continued to undergo dietary advice. The participants received a full medical examination each year and an electrocardiogram (ECG) was obtained yearly or as required clinically.

Blood and ECG data collected at the health centre were digitally coded and sent to a central *Analysis Centre* where they were analysed using modified ECG software. Results and data generated by this analyses was sent to a central *Data Centre*. Similarly, data collected at the health centres were archived at the central Data Centre. Case report forms were collected and validated at the data centre. When required, the data centre provided blinded and non-blinded reports to an *Executive Committee* and *Data & Safety Monitoring Committee* respectively. A report is blinded when the identity and treatments of individual participant are unknown.

Study co-ordination and management of drug dispensing was the responsibility of an Administrative Centre. The administrative centre was also responsible for the servicing of committees including ethics, adverse-events and end-points committees. The administrative centre was where all randomisation was approved and where all pre-randomisation data was gathered before being sent to the Data Centre. Figure 8.1 shows the study's organisation structure and the flow of data.

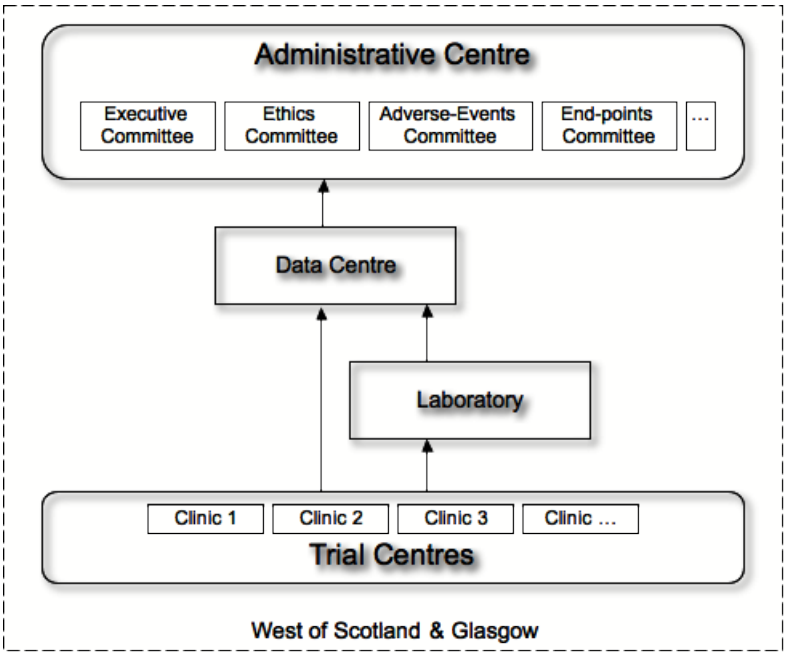


Figure 8.1: WOSCOPS Organisational Structure and Data flow

WOSCOPS Roles: Apart from the roles committees play, other role were identified. In each health centre, the trial physician conducted the randomisation and yearly visits. The trial nurse carried out other visits and documented the visits, which the trial physician checked and approved. Participants with cases that the nurse identified were referred to the trial physician for treatment. The physician documented these cases. Signs and symptom data, which was documented at each visit, was evaluated by the trial physician and scrutinised by the appropriate committees. A clinical research associate from the Administrative Centre collected detailed hospital records in cases where participants were hospitalised during the course of the trial. These cases were documented and scrutinised by the Adverse Events Committee, and the participant potentially withdrawn from trial medication. In some cases these events were also reported to regulatory bodies.

8.2.2 The Prospective Study of Pravastatin in the Elderly at Risk

The Prospective Study of Pravastatin in the Elderly at Risk (PROSPER) was a study that tested the hypothesis that the use of 40mg/day of pravastatin would reduce the risk of cardiovascular and cerebrovascular episodes in elderly people who have vascular disease or were at

high risk of developing vascular disease [185, 186]. The study was double-blinded, randomised and placebo-controlled to test this hypothesis and was tested in collaborating centres in three countries: Cork (Ireland), Glasgow (Scotland) and Leiden (The Netherlands) over a period of about three and half years. The studies primary endpoints were: "definite plus suspect coronary heart disease death; definite plus suspect non-fatal myocardial infarction; and fatal plus non-fatal stroke" [186].

PROSPER Screening and Recruitment: Between the three cities, 23,770 men and women were invited to the first screening. At this visit, written informed consent (Appendix B) was received from all those that came. Also a brief medical history was taken, the subject's vital signs were recorded and dietary advice given. Subjects that satisfied the eligibility criteria for the next visit were invited to the second screening. At the second visit, a more detailed medical history was taken, blood samples collected and medication checks carried out. Based on the results of blood tests, subjects were invited to the first enrolment visit. Second blood samples were collected for further testing at this stage and stored in the PROSPER *bio-bank*. A 12-lead Electrocardiogram (ECG) was recorded and a Mini-Mental State Examination (MMSE) along with other psychometric tests conducted. At the final enrolment visit, more physical data was collected and more medical examinations carried out with final checks performed by the study investigator. Subjects that satisfied all the recruitment criteria had their consent endorsed by their general practitioner and were randomised for the study.

About 5,800 elderly men (2800) and women (3000), aged between 70 – 82 years, with plasma cholesterol levels between 4.0 – 9.0 mmol/L were recruited to the study. It was noted that about half of the study population had evidence of vascular disease and that the other half were at high risk of vascular disease. The trial participants were identified after their third visit - during a 10 week screening and enrolment period. The participants were randomised and during the double-blinded phase, they visited the trial centres every three months.

PROSPER Functional Units: Figure 8.2 shows the distribution of the functional units among collaborating partners. Most of the centralised units were located in Glasgow under different organisation control: the Data Centre of the Robertson Centre for Biostatistics; the Central lipid laboratory of the Department of Patholgical Biochemsitry at Glasgow Royal Infirmary and the ECG laboratory of the Department of Cardiology at Glasgow Royal Infirmary. The Bio-Bank unit was located and managed at the Leiden University Medical Centre.

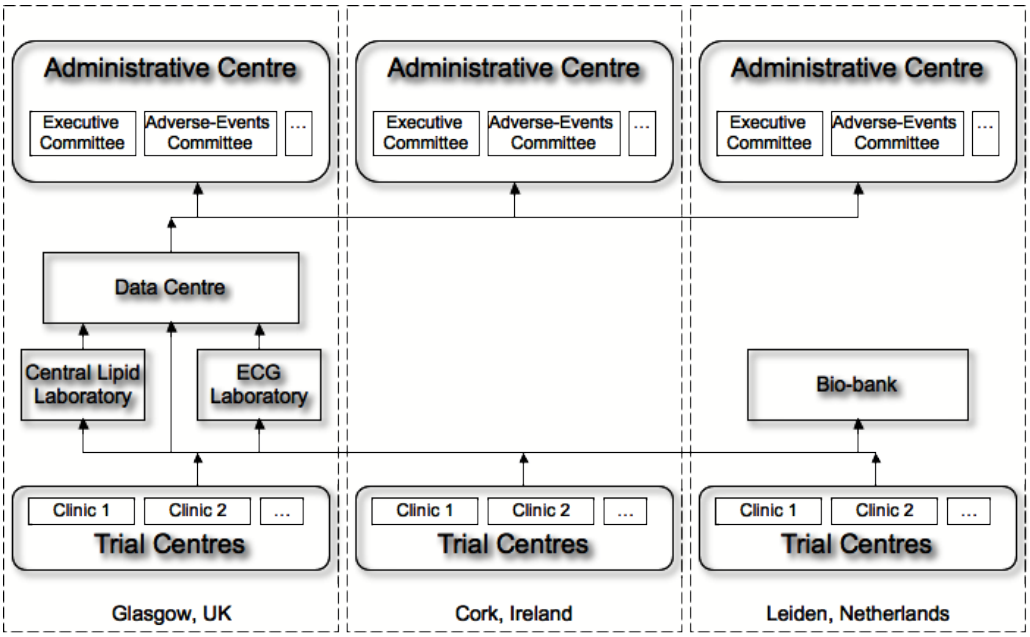


Figure 8.2: PROSPER Organisational Structure and Data flow

The Data Centre provided the central randomisation system; statistical analyses for the study; case report forms, data entry and validation, and appropriate reports to committees. The central lipid laboratory provided all lipid analyses and the ECG laboratory provided all electrocardiograph analyses for the study. All medical samples such as blood, plasma, serum and cells, were stored at the central Bio-bank. The three administrative centres, one for each country, managed the study centres and provided clinical support to the study nurses and sub-investigators, e.g. general practitioners. They also managed the flow of data and trial medication, as well servicing study committees in each country.

PROSPER Roles: The key roles identified in the PROSPER study include: the study investigator, study nurse, study sub-investigator or general practitioner (GP), clinician, statistician and study committees. Study committees include executive, data and safety monitoring, endpoint and publications committees. Study investigators have access to participants medical data and history, whilst general practitioners only have access to their patients (participant) data. Study nurses are responsible for data collection on a per centre basis, whilst clinicians have access to data collected for laboratory analyses. Statisticians have access to randomisation data for statistical analyses. Executive committee members have access to

blinded reports, whilst data and safety committee have access to unblinded reports.

8.2.3 Trials Review

In order to automate the clinical trial process electronically, it is important to understand how the process works. The trials described above helped to raise some important questions that are common across local and multi-centre research trials. The first question is how to invite and recruit participants for a trial. As can be seen in the above examples, this is dependent on the endpoints (defined in Appendix B) of a given trial. However, from a technical perspective, this is difficult to achieve as medical records required for this purpose are locked away in individual silos, which are managed and protected by different health organisations with their own security policies. In multi-countries trials, data privacy laws and governmental policies of participating countries become a point of contention, most especially as they impact keeping patient data private and confidential between countries. Another question related to this is how to obtain and use patient consent as it affects querying patient records for the selection and invitation stages.

The second question is how to co-ordinate the screening process across health centres. Ubiquitous systems that support multi-centre screening and management would have be in place for this purpose. However, a key challenge to this is how to authorise and enforce such a process in an "open" and decentralised environment[2]. Related to this is the question of who does what or can do what. For example, what roles are involved in a trial [177], how role membership is determined and how these roles are mapped to local privileges. For example, over 33,000 people work in the NHS Glasgow alone, and over 187 different IT-related roles exist with various kinds of access to data, i.e. different clinical roles with different privileges to IT systems and data. In a multi-centre trial, the question is how roles can be identified and mapped to roles across different centres considering the heterogeneity of local roles and multiplicity of actions that can be performed on data, which offers huge challenges – not least due to the legacy systems in place.

The third question is how data is collected and stored in data centres. The question of who has access to this data is very important. As can be seen in above examples, data centres can be centralised or independently managed at various distributed centres. This requires access on an ongoing basis, which raises management and data access challenges for data stored

[2] Open in terms of ubiquitous systems but not in terms of organisations. The nature of organisations in the health domain can be said to be closed as access to their systems, resources and data are tightly controlled.

across boundaries and behind firewalls. For example, decisions need to be made about how to allow access to these data sets to the right people for the legitimate purpose. Similarly, end users should be unaware of the fine-grained security solutions that are restricting and controlling their access and use of the data at different data centres. These questions present far reaching issues in collaborative multi-centre trials, i.e. on how trust can be realised in such security-oriented collaborative environments.

8.3 Frameworks for Clinical Trials

As each individual trial faces the same kinds of challenges for recruitment, data management and study co-ordination, a framework supporting a multitude of trials would be extremely beneficial. To establish a framework for clinical trials requires addressing heterogeneity, the distribution of systems and data sets, and differences in general practices, e.g. how data is backed up (or not) at given sites. It could be argued that the immediate challenge in the establishment of an electronic clinical trial is how to recruit people. Key sources of data in Scotland include national census data sets such as the General Register Office for Scotland [187] which includes information such as the registration of births, marriages, deaths as well as being the main sources of family history records. The access to such information, whilst useful, does not include direct health related information which is likely to impact upon the suitability of patients to a trial. Primary and secondary healthcare data sets are other immediate choices. However, access to and use of, these data sets will require ethical approval. However, in running a clinical trial, it is often the case that statistical information is enough, e.g. the number of participants that matches a criteria, the percentage of participants that developed an adverse-events, the length of time for an adverse-event. Thus, rather than disclosing information on specific patients, statistical information is sufficient.

A key challenge from an IT perspective, as discussed in this thesis, is security. The risk of data disclosure is ever present and cannot be over emphasised. Ensuring that data guardians and health professionals with strategic roles for the management of the data protection or confidentiality associated with patient data sets are involved in the decisions that influence the development of such infrastructures is crucial to their success; from their development, their acceptance, and perhaps more importantly their ethical use [126].

8.4 Infrastructures and Data Sets Across Scotland

A framework that supports a multitudes of trials was explored in the MRC funded Virtual Organisation for Trials of Epidemiological Studies (VOTES) project [126, 127]. The VOTES project was a collaborative effort between e-Science, clinical and ethical research centres across the UK, including the universities of Oxford, Glasgow, Imperial, Nottingham and Leicester. The primary focus of VOTES was to build a clinical framework that supports a multitude of clinical virtual organisations (VOs). As noted, VOs are a common concept in the Grid community and provide a conceptual framework through which the rules associated with the participants, their roles and the resources to be shared can be agreed and subsequently enforced across the Grid. VOs in the clinical trials domain are characterised by a much greater emphasis on security, data access and ownership. These are called Clinical Virtual Organisations (CVOs) since they have requirements not typical to other HPC-oriented VOs common to the wider Grid community. Rather than developing bespoke CVOs for each individual clinical trial, VOTES intention was to develop a framework supporting a multitude of CVOs.

Each of these CVOs was derived from the framework and adapted depending on the needs of the trial or study being conducted. Common phases of many clinical trials and epidemiological studies, and the primary focus for core components that exist in the VOTES Grid framework were:

- Patient recruitment enabling semi-automated recruitment methods for investigators conducting large-scale clinical studies in a variety of settings;
- Data collection incorporating data entry including intermittent connectivity to other resources, such as trial-specific databases, code lists for adverse events and non-study drugs[3], randomisation programs and support for internationalisation in case report forms;
- Study administration, which include detailed logging of essential documents, enabling rapid dissemination of study documentation and by co-ordinating the transport of study treatment and collection of study samples.

In order to develop a Grid framework for clinical trials, the potential sources of data and services that allow access to data are first identified and analysed. Within the Scottish element of VOTES, the NHS in Scotland are the primary source of data sets and software that

[3] These are drug prescriptions or treatments that participants may be undergoing besides treatments being received on the study.

are needed for clinical trials and epidemiological studies. The following are some identified data sets and software in Scotland[4]:

- **General Practice Administration System for Scotland (GPASS)** [155] is the core IT application used by over 85% of clinicians and general practitioners involved in primary care across Scotland;
- **Scottish Morbidity Records (SMR)** [188] includes records relating to all patients discharged from non-psychiatric and non-obstetric wards in Scottish hospitals (including datasets on death, cancer and hospital admissions);
- **Scottish Care Information Store (SCI Store)** [154] provides a batch storage system which allows hospitals to add a variety of information to be shared across the community, e.g. pathology, radiology and biochemistry lab results are just some of the data that are supported. Regular updates to SCI Store are provided by the commercial supplier using a Web Services interface. Currently there are 19 different SCI Store installations across Scotland (with 3 across the Strathclyde region alone). Each of these SCI Store versions has their own data models (and schemas) based upon the regional hospital systems they are supporting. The schemas and software are undergoing continued development;
- **NHS data dictionary** [189] provides a one-stop shop for health and social care data definitions and standards. It contains a summary of concepts for SMR datasets including online manuals for the datasets; information on the clinical datasets in use in healthcare and social care datasets along with the data standards upon which they are based.

8.5 Data Standards and Distributed Security Challenges

As CVOs commonly span heterogeneous domains, a requirement for the construction of distributed queries and aggregation, or joining, of distributed data is the development and use of a standard method of classification or common vocabulary. This includes the naming of the data sets themselves, the people involved and their roles (privileges) with regard to access and use of these data sets. Preferably these data sets and roles should be standardised so that comparisons can be made and queries joined together, for example, across a range of clinical data sets.

[4]This does not imply that these data are readily available directly, but that are potentially available to interface with.

There are quite a few developments in standards for the description of data sets used in the clinical trials domain. However, this can be an involved process depending on the standards groups developing and acting on strategies put together through major initiatives such as Health-Level 7 (HL7) [135], SNOMED-CT [190] and OpenEHR [191]. There are a wide range of legacy data sets and naming conventions which impact upon standardisation processes and their acceptance. The International Statistical Classification of Disease and Related Health Problems version 10 (ICD-10) [192] is used for the recording of diseases, health related problems and is supported by the World Health Organisation. For example, within the NHS Scotland, ICD-10 is used along with ICD-9 and read codes in the SMR data sets.

Linking standardised data descriptions between domains so that entities and relationships within one organisational hierarchy can be mapped or understood within the context of another domain is fundamental to any framework. Once it has been established how meaningful comparisons can be made between the differing domains, the framework can be applied to a generic clinical trial that could run queries across heterogeneous domains, bringing back results, richer in scope and information than if single local sites had been independently queried.

Since information stored in clinical trials are highly sensitive, data obtained or collected must be kept in the strictest confidence and the integrity of the data must be maintained. The exact data should only be revealed to few roles in a trial. This was one of the most fundamental challenges of VOTES to realise the opportunities and benefits that can be brought to clinical trials by Grid technology, but also to maintain the high security standards that must be strictly adhered to.

Security policies will naturally differ between local sites, which leads to several challenges when defining and implementing policies that take into account both local and remote security concerns. These include:

- Applying a generic policy that takes into account each local policy or links local policies together using a standard interface;
- Dynamically enforcing these policies so that, for example, restrictions applied by a site not providing pertinent information for a particular query will not impact on the other sites that are involved;
- Building a trust chain that allows local sites to authenticate with a VO and therefore, by proxy, be authenticated to access limited resources at other sites without compromising protected resources at those sites;

- Prevention of inference (statistical disclosure) that may arise when data from multiple sources are joined together;
- Maintaining data ownership and enforcing ownership policies regardless of where the data might be moved, stored or used.

8.6 Scenarios

To consider the challenges of clinical trials and provide the context in which DTN is explored, several key scenarios are outlined and the requirements these place on secure collaborations. The first scenario presents a representative sequence of interactions demonstrating how recruitment of patients can be *ethically* achieved. The second and third are overview scenarios for data collection and study management respectively.

Scenario 1 - Patient recruitment: This scenario presents a representative sequence of interactions demonstrating how recruitment of patients can be *ethically* achieved.

1. A trials co-ordinator logs into a web portal that provides a visual interface to various CVOs associated with a variety of clinical trials[5] and/or tentative trials. After authenticating, a personalised environment is created based upon the specific role (in this case, that of the trials coordinator) in the CVO and the location from where they are accessing the portal. He/She is only shown the Grid services pertinent to the appropriate trials applicable to him/her, and hence the data sets associated with those services.
2. The trial coordinator wishes to recruit patients for a leukaemia cancer trial. Patient details are available in hospital and GPs local (and secure) databases. Emails are sent to the GP practices or hospitals with information describing the particular trial to be conducted, the general criteria applicable to matching patients and other information, e.g. financial information about participating in the trial. The email contains a link to a Grid service (Leukaemia trial 2006). The GPs and Consultants themselves are described in policies associated with the tentative set up of a CVO, for patient identification and recruitment.

[5] Of course there are scenarios which predate this one, e.g. how the CVO is established in the first instance and the policies by which the VO will be organised, managed and enforced.

3. It is assumed that the GP/Consultant is interested in entering into the trial, i.e. have matching patients and they follow the attached link. The GP/Consultant may securely access the Grid service either using a username and password combination or using a digital certificate, e.g. X509 certificate. In this scenario, it is assumed that X509 certificates are being used.

4. After extracting more information about the trial from the portal, the GP/Consultant decides to download a signed XML pro-form pre-designed for the trial. This is a partially completed document describing the main information relevant to the trial as documented in the trial protocol, where the empty fields need to be filled through a query to the GP practice or hospital databases.

5. The signature of the signed pro-forma document is checked to ensure its authenticity and to ensure that it has not been corrupted. If these are both true, the document is used as the basis for an XML query against the GP practice or hospital databases (GPASS supports such an interface). This query might, in turn, result in further information being extracted from other resources.

6. At this point, letters describing the trial to selected patients can be automatically produced. These are used to obtain patient consent before continuing further with the recruitment.

7. The selected patients may then consent to participating into the trial. Note that their letters of consent may be sent directly to the trial coordinator instead of the GP/ Consultant as described here.

8. The forms are automatically completed based on the results of the queries to the GP practice or hospital database. The forms are digitally signed and returned to the Grid service for that particular trial (Leukaemia trial 2006).

9. The returned and signed XML document is authenticated. Verification that the sender (the GP/Consultant) is authorised to upload the document are made, e.g. through checking that they were one of the GPs/Consultants contacted initially. The document is validated to ensure its correctness, e.g. by ensuring it satisfies the associated schema and the relevant data fields are meaningfully completed (and match the desired constraints associated with participation in the trial). At this point, the responding GP/Consultant is formally added to the CVO. Further follow up information may subsequently be sought, e.g. monitoring information related to the selected patients.

10. The completed XML document and the associated meta-data describing the history of

how this information was established, by whom, when, and for what trial are uploaded and securely added to the CVO repository for this particular trial.

It is important to note in this scenario that patient consent is given (step 6) before patient data is returned to the clinical trials team. Another important aspect here is that the GP can decide whether this might be in the patient's interest. The patient may ultimately say no and hence is always involved in the consent process.

Scenario 2 - Data collection: A trial investigator submits a query for data that is generated or stored at different study centres. The types of data to collect will include lab results, patient medical history, trial data from trial databases, code lists for adverse events and follow-up data. Data is expected to be pulled and aggregated from geographically distributed locations, which include hospitals, primary care information systems, mobile centres, PDAs and laptops. Security concerns include statistical disclosure, confidentiality, privacy and data integrity.

Scenario 3 - Study Management: A steering or data monitoring committee member investigating the adherence/compliance of a study to an agreed protocol, requests access to collected data and trial investigators audit trails. He/She is expected to generate reports based on his/her observations and analysis. He/She is expected to execute statistical programs on data collected from different sources. His/Her requirements include understanding the semantics and structure of collected data. Security requirements here include confidentiality, privacy and data integrity.

8.7 Summary

This chapter reported on the various clinical trials investigations that served as the basis for testing the DTN framework and its implementation. The chapter described how two case studies were carried out, of various magnitudes and scale, to test and validate the security-oriented requirements of collaborative centres in e-Health environments. The trial reviews revealed the need for trust realisation in security-oriented collaborative environments. These reviews formed the basis for the DTN framework, developed and applied for clinical trials.

The chapter described how the VOTES project provided a basis for exploring this research work. Background information on VOTES was given including how it was a collaborative effort between e-Science, clinical and ethical research centres across the UK including the

universities of Oxford, Glasgow, Imperial, Nottingham and Leicester. The chapter described how the aim of VOTES was to build a clinical framework that supported a multitude of clinical virtual organisations. Among the findings from this work it was identified that a key need exists for data standardisation and distributed security. The chapter concluded with different scenarios tackling the issue of patient recruitment, data collection and study management in the e-Health clinical trials domain.

9 DTN Implementation in VOTES

This chapter describes the DTN implementation in systems that have been developed to support clinical trials. Two systems developed for VOTES are described and the implementation of DTN in these systems is discussed. The chapter concludes with the performance and evaluation of the DTN implementation.

9.1 Virtual Organisations for Trials of Epidemiological Studies

Successful e-Health research depends on access to, and use of, a wide range of clinical, biomedical, social, geospatial, environmental and other data sets. In large scale, multi-centre clinical studies crossing geographical and organisational divides, the need to access, link and aggregate data securely is core. Whilst the Grid community have come up with a wide variety of technologies that support authentication and authorisation, experiences in the Virtual Organisations for Trials and Epidemiological Studies (VOTES) project have shown that irrespective of the technological advances and capabilities offered by the Grid community, data providers themselves are typically unwilling to provide direct access to their data sets, i.e. through the penetration of the NHS firewalls, for example, from Higher Education / Further Education (HE/FE). This is in addition to the European Union directive that says health providers, like the NHS, should only interact with parties they have explicit contracts with [157], e.g. informed consent [193, 194] given by a patient for his/her data to be accessible by another party is a form of a contract. To this end, prototype systems were developed as part of the VOTES project. These are described in the following sections.

9.2 The VOTES Distributed Data Framework

The VOTES Distributed Data Framework (VOTES-DDF) was the first system prototype and was based on the development of a framework for clinical trials infrastructure exploiting Grid technologies. The architecture shown in Figure 9.1 was used for exploring scenarios of federating data across clinical domains. The framework relies on various Grid technologies to access data sets that are provided by data providers such as SCI store and GPASS. The architecture relies on a multi-database to decompose queries and aggregate results, which are made available using a Grid data service.

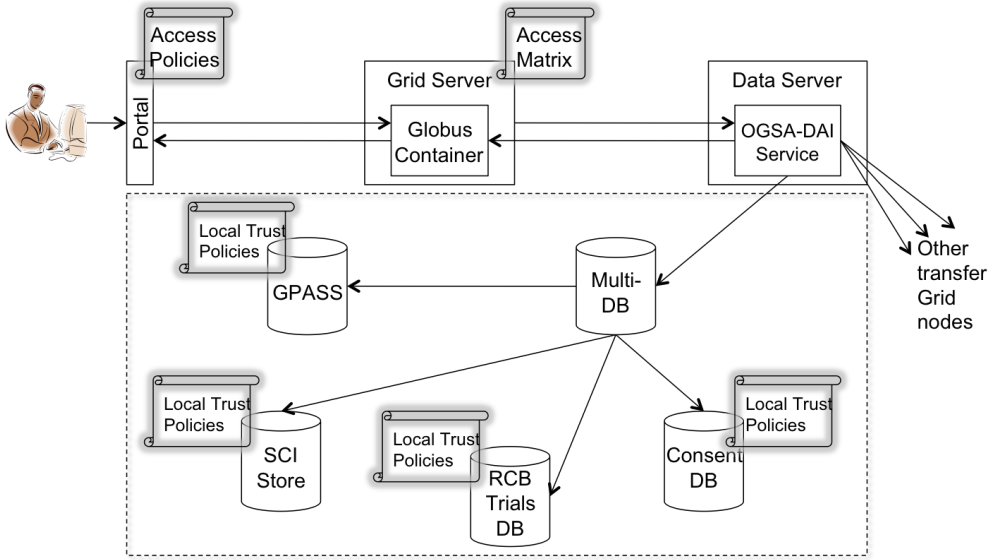


Figure 9.1: The VOTES Architecture [195]

The framework was modelled after the federated database system presented in [4]. Within the framework, distributed, heterogeneous and autonomous databases could be accessed and managed. The implementation of VOTES-DDF provided access to multiple SCI Store repositories, a GPASS repository, a consent database, and a clinical trial repository containing representative clinical trial data from the Robertson Centre for Biostatistics at the University of Glasgow.

Using Grid services often requires appropriate adaptors to facilitate access to, and use of, specific data resources. The OGSA-DAI [196], a Grid data service, provides a middleware framework for managing, accessing, and integrating relational data in XML and data held in a

	R_1	R_2	R_3	R_4
	h_1	h_2	h_3	h_4
U_1	0	0	1	0
U_2	0	0	0	1
U_3	1	1	1	1
U_4	0	1	0	0

$$U_1(R_1 \triangle h_3) = 1$$

$$U_2(R_1 \triangle h_2) = 0$$

$$U_3(R_3 \triangle h_1) = 1$$

$$U_4(R_1 \triangle R_3 \triangle h_2) = 1$$

where $\triangle$ is a combination function
0, 1 are deny and grant respectively
R_x, h_x are resources and subjects respectively

Figure 9.2: An Access Matrix model

variety of different databases such as SQL Server, Oracle, MySQL and Xindice [197]. OGSA-DAI fits into the data federation model because it provides a collection of data components for querying, transforming, and for delivering data in various forms.

A key aspect of the infrastructure was exploring how patient consent could be handled. The system supported a variety of models which allowed patients to consent to their data being used for a specific clinical trial, e.g. for a particular disease area or to be used generally. In addition, the system also allowed for patient to opt out, i.e. their data sets may not be used for any purpose. Numerous variations on this were also explored, e.g. the patient's data may only be used provided they are contacted in advance. To support this, a consent database was established and was used when joining the federated queries undertaken to decide whether the data should be anonymised and displayed, or not displayed at all.

The authorisation mechanism implemented an access matrix model [1] that specified bit-wise privileges of users and their associations to data objects in the CVO. Data objects are defined as fields, tables, views, databases and sites, for the purposes of fine-grained authorisation. As shown in Figure 9.2, the access matrix was designed to enforce discretionary and role based access control policies.

As shown in Figure 9.1 The federated data system was composed of four autonomous test sites, each providing clinical data sources. The data sources exposed by these sites were configured as data resources on an OGSA-DAI data service. Each data resource was seen as a

node in the data federation. The OGSA-DAI data service implemented a head node model to drive the data federation. The head node was selected based on rules or request requirements and was responsible for decomposing queries, distributing sub-queries and aggregating query results.

In the implementation, data federation security was achieved at both local and remote levels. The local level security, managed by each test site, filters and validates requests based on local policies at DBMS levels. The remote level security was achieved by the exchange of access tokens between the designated Source of Authority (SOA) at each site. These access tokens were used to establish remote database connections between the sites in the federation. In principle local sites authorise their users based on delegated remote policies. This is similar to the CAS model [64].

9.3 VOTES-DDF$_{\mathrm{DTN}}$

Authorisation based on local and remote security levels is a great challenge. The ideal situation is when ultimate access control rests with the local resource providers - and the VO policy simply acknowledges their autonomy. In order for this to be achieved, these roles (or credentials) must be negotiated and exchanged between resource providers (nodes) in a flexible and secure way. DTN facilitates this by introducing a negotiation layer, where the local trust policies are managed by resource managers (RM) which grant or deny access to their resources based on negotiated attributes. Needless to say in, the clinical domain, data providers are unwilling to negotiate with parties that are not trusted directly.

The negotiation and discovery layers were implemented as GT4 services. In order to integrate it with VOTES, the Identity Provider (IdP) connector initiated credential negotiation via the negotiation service. The discovery service was used to realise trust-pathways, which must be invoked whenever a new path needed to be discovered or when existing paths needed to be revalidated.

When a user tries to access a remote data resource protected by a Service Provider (SP), they are redirected to their home Identity Provider (IdP) through the WAYF service [198], shown in Figure 9.3. The user is authenticated at the IdP, e.g. using an LDAP repository. The IdP sends the SP a SAML [199] response that contains an authentication assertion. This assertion is forwarded to the SP's assertion consumer service, which validates the assertion. This authentication assertion includes a temporary pseudonym for the user (the handle) that the SP can use to reference the user. After validating the authentication assertion, the SP

creates an attribute-token, which is sent to the user's IdP's attribute authority along with the user's handle in a SAML attribute query message. The attribute-token and the user's handle can then be linked together by the SP to provide the SAML-DTN support.

On receiving a SAML attribute query message, the IdP initiates the home negotiation service using the SP's resource token to negotiate for the user attributes. The home negotiation service negotiates the user attributes with nodes (organisations/sites) that are next-hop nodes to the target node (SP). Each negotiation hop includes the passing of resource tokens from node to node, which links negotiated attributes to a resource request. Resource tokens and negotiated attributes are stored by the target node in its Negotiated Attributes Store (NAS).

When an IdP receives negotiation responses from its next-hop nodes, it returns a SAML attribute assertion message to the SP. If negotiations were successful, the assertion notifies the SP to collect negotiated attributes from its NAS. The SP uses a user's handle to retrieve a user's attribute-token, which is used to query a NAS for negotiated attributes. The negotiated attributes are used to make authorisation decisions as to what the user can access. If the negotiation fails or times out, the assertion will contain the user attributes as it normally would, i.e. if it was not extended with DTN. In this case the user attributes may be invalid at the SP and may not be applicable for authorisation decisions.

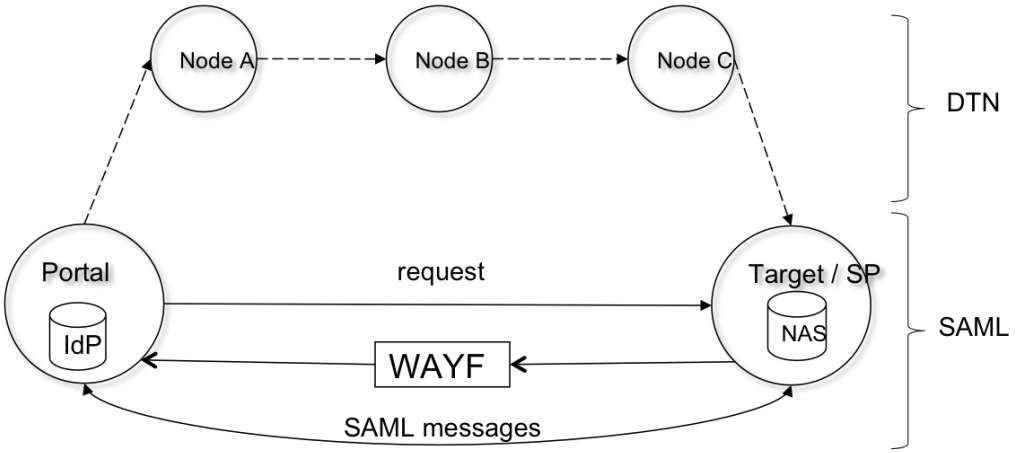


Figure 9.3: SAML-DTN model view

9.4 The Virtual Anonymisation Grid for Unified Access to Remote Clinical Data (VANGAURD)

Data providers and the key stake-holders in the health domain are acutely aware of confidentiality and ethics concerns on data access and use. They will only release their data provided it can be ensured that it is not possible to link it with other data sets that can result in potential violations of patient confidentiality, for example, through statistical disclosure. The Virtual Anonymisation Grid for Unified Access to Remote Clinical Data (VANGAURD) [200, 195] is a system designed in VOTES for secure anonymised data access and linkage that meets the needs of data providers in clinical trials. Key features of VANGAURD include:

- support for pull models of interaction with data providers such as the NHS, who do not necessarily have to open up their firewalls and thereby become susceptible to risks of attack;
- support of secure, anonymous data aggregation thus ensuring data integrity;
- support for ways in which data release to users undertaking research allows them to obtain and use data in a secure, disclosure free environment where third parties cannot access/use any released data.

VANGAURD explored a new paradigm in the way health institutions share data and collaborate. VANGAURD deviates from existing approaches where clinical data systems are queried by Grid based systems directly. Rather, a pull and push model approach where clinical data systems pull query requests and push query results to Grid based services is offered. This approach was created from the increasing wariness or scepticism of data providers who will not allow direct access through their firewall to their data.

Focussing on the need for information governance, VANGAURD was designed to ensure that data release is tightly controlled with strict compliance to confidentiality and integrity. To achieve this, strong encryption was used whenever data was being exchanged between systems or temporarily stored in memory. Access to datasets is controlled locally by the provider in the VANGAURD system. That is, different parts of a resultset are controlled and secured by the different providers, and yet it is possible for a trusted intermediary to aggregate and link the different parts of the resultset without the possibility for further linkage.

In VANGAURD, queries are defined based on understanding of the data models of different systems, i.e. based on the knowledge of the schema that have been made available by data providers. If a site has joined a CVO participating in a given study, it may subsequently pull

queries that are targeted to itself into its clinical system. Based on its local security policies, these queries are validated and authorised, and if valid, are executed. Once executed, the protected resultsets are pushed out to the requester.

In short, the clinical systems are completely protected from inbound Internet connections. Rather they are based upon an outbound Internet connection model. However, the question of security must still be explicitly satisfied, i.e. what queries are being defined by whom and what artefacts are co-ordinating the access to, and use of, clinical data resources to users with particular privileges. The VANGAURD system architecture is shown in Figure 9.4 and shows the following principal components: Viewer, Guardian, Agent and Banker.

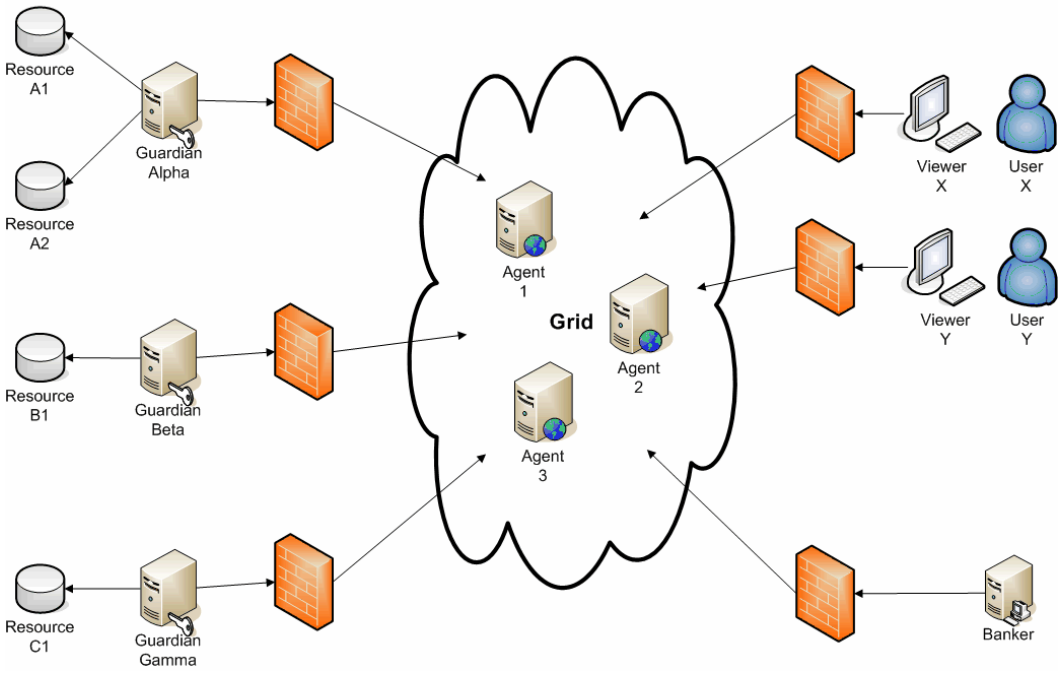


Figure 9.4: VANGAURD Component Model [195]

Viewer: The Viewer is a component run by the end-users of VANGAURD. It provide a means by which users (researchers / investigator for example) can authenticate, submit queries and collect query results. Viewers only interact with Agents on behalf of their users. Only a requester can view a result, as results are encrypted with a users individual public-key. This implies that an Agent (or any other third party) cannot read a user's results.

The data available to a user is dependent upon the privileges that the user possesses, e.g. this model presents authorisation challenges as user privileges are artefacts of the user local authority. To address the challenges, digitally signed security attributes incorporating role-based attribute access control models are used. These security attributes are specific to a given VO, study or trial where they have been agreed upon by all data providers. The security attributes are then used to enforce local access decisions, i.e. data requests combined with security attributes are used to determine the authorisation decision on access to the local data.

Agent: Services as Agents (SaA), are the central piece of the VANGAURD model. They act as intermediaries between other components. Agents securely collect and aggregate data from multiple Guardians. Every data request passes through Agents to Guardians. As a request could require results from multiple Guardians, Agents are responsible for decomposing requests and compartmentalising requests for each Guardian to collect, authorise and execute. Guardians deliver results to Agents and Viewers collect results meant for their users from Agents.

In addition to the responsibility of co-ordination and secure communications between components, an Agent is responsible for the generation of hashing keys for use by Guardian systems. By generating and co-ordinating hashing keys, Agents are able to securely link data across sites without direct data disclosure being made. For example, data x from Guardian A can be linked to data y from Guardian B that are both issued a hashing key k from Agent C, based on the following expression:

$$hash(k, index(x)) \iff hash(k, index(y))$$

where,

$$index(x) = index(y)$$

Agents can be knowledge or domain or clinical trial specific. In a collaborative environment it may well be the case that more than one Agent exists for different purposes. For example, in a collaborative environment, like clinical trials, an Agent might exist for diabetes studies whilst another for cardiovascular studies and another for cancer studies. In this environment, a data provider may register with a diabetes Agent to participate in a diabetes trial and a different Agent in cardiovascular trial. Having multiple Agents is key to VANGAURD as Agents must be able to know which Guardian has what data available. In essence, an Agent acts like a registry service that SP may use to advertise what services they provide. This

registry is, however, only populated with trusted services and should only be accessible to trusted clients. However, this trust may well have to be negotiated as is described in Section 9.5.

Guardian: Guardians represent the interface/ protection of data providers. They only interact with Agents to collect user queries, authorise queries, execute queries, provide hashed fields for joining and encrypt results using user public-keys. One or more Guardian systems can be involved in responding to a request made by a user. A site can implement more than one Guardian system for different categories of data or for different participation in a given VO or study. Guardians publish their data schema along with access policies to Agents, so that Agents are aware of what they provide and, in case of agreed attributes, user roles they recognise.

Guardians pull queries *periodically* from Agents and push the results to the respective Agents. In situations where a Guardian's data is to be joined with data from other providers, the Guardian uses an Agent hashing-key hashes the index that uniquely identifies a row in the resultset that is pushed to the Agent. As the Community Health Index (CHI) and NHS number are the most widely used indexes in the UK, hashing based on a key-based hashing algorithm [168] make it possible to link data between multiple Guardians.

Prior to receiving any queries, a Guardian provides and *periodically* updates an Agent with detailed information on the data model it has available, along with access information for the data model, i.e. what part of the data model is accessible with what security attributes. A Guardian is able to bind access information to a data model in situations where access attributes have already been agreed. The access information is to enable a Viewer, instructed by an Agent, to display different data resources available to a particular user.

Banker: The Banker is responsible for managing the monetary value placed on resources across the whole VANGUARD system. It provides the accounting functionality for the services rendered by Guardians. Users buy credits with which they pay for every data request they make in VANGUARD. The purpose of the Banker is that the services rendered may be quantified, controlled and monitored. With the Banker every requests made can be logged and can later be audited. To achieve its goal, the Banker periodically interacts with Agents to update Agents with credits that a user has left. It also queries Agents for users requests so as to be able to quantify what has a user has spent.

The notion of a Banker in VANGUARD is relatively new. It echoes the notion of Accounting

in Authentication, Authorisation and Accounting (AAA) for Grid systems [201]. It is still not yet clear how scalable the Banker will be in a large collaborative environment. This is an area that needs further research.

9.5 VANGUARD$_{DTN}$

The sensitive nature of clinical data makes security a high priority and any method of federating this data must adhere rigourously to the local security policies that protect this data. Whilst VANGUARD addressed the concern of data providers for secure anonymised data access and the pull model requirement, it presents decentralised authorisation challenges as data providers are autonomous and they control access to their resources.

Decentralised authorisation decisions can be made based on security attributes that have been agreed for a VO or study. However, experience has shown that there is a slow uptake for reaching agreement on security attributes between multiple parties as each party wants attributes that are closely related or equal to their local security attributes. To address this decentralised authorisation challenge, DTN is proposed as it not only provides a means to negotiate security attributes for data access, it also adheres to EU directives that require explicit contracts for data access between parties.

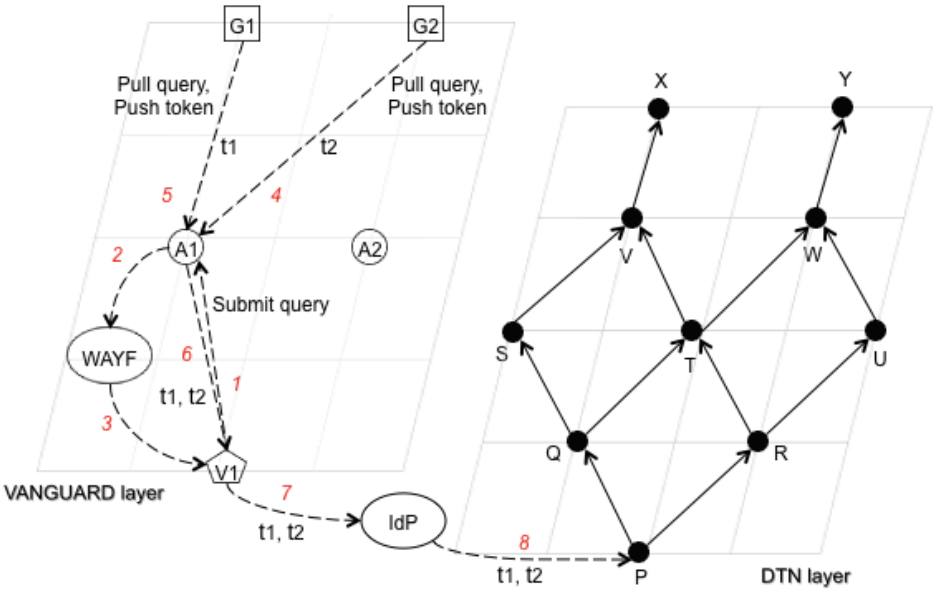


Figure 9.5: VANGAURD-DTN Layered Architecture

In Figure 9.5, DTN is shown acting as an underlying framework for the VANGAURD system. This is used within a Shibboleth-based access management environment. To understand the role of DTN in VANGAURD a typical scenario is provided. Bob from the Robertson Centre for Biostatistics (RCB), Glasgow, is a bio-statistician who wants to find out how many patients aged between $40-50$ are currently being been treated for Leukaemia across Scotland. He logs in via a RCB Viewer ($V1$) to a Leukaemia (cancer) Agent ($A1$). The Agent responds with a Where Are You From (WAYF) and Bob gets redirected to RCB identity provider (IdP) for authentication. The IdP sends a SAML authentication assertion response back to the Agent. Bob then submits his query to the Agent and the Agent decomposes the query and identifies that Guardians $G1$ and $G2$ that are needed for that particular query. Periodically, $G1$ and $G2$ pull queries meant for them from $A1$ and both respond with a request (authorisation) token, $t1$ from $G1$ and $t2$ from $G2$. The Agent then makes a SAML authorisation request to Bob's ($V1$) IdP passing along request tokens $t1$ and $t2$ for Guardians $G1$ and $G2$ respectively. In this scenario in Figure 9.5 each of the sites have components, i.e. a negotiation service for trust negotiation. $G1$'s negotiation service is X; $G2$'s is Y and $V1$'s is P. In the DTN network shown in Figure 9.5, other sites negotiation services exist also, i.e. Q, R, S, T, U, V and W. The $V1$ IdP invokes its negotiation service P, which initiates a trust negotiation with Bob's credentials along with the request tokens $t1$ and $t2$ for target nodes X and Y respectively. On completion of trust negotiations between P, Q, R, S, T, U, V and W, $G1$ and $G2$ make authorisation decisions with Bob's negotiated attributes in their NAS. Based on the authorisation decision $G1$ and $G2$ push the query result to $A1$ which joins the data on the hashed indexes and removes this joining information to prevent further data linkage. Bob via $V1$ is able to pull the linked, joined and anonymised query result from $A1$.

9.6 Experiment and Evaluation

Performance study of computer networks has gained prominence because of the need to understand network protocols and the need to create more efficient network protocols. These studies focus on protocol behaviours under numerous defined conditions owing to the many attributes of protocols and networks, and the studies are usually carried out through simulations and/or on real networks. However, performance characteristics such as network bandwidth, delays, queue size, traffic sources and network loads are difficult to control on real networks, hence has led to the development of many network simulators, like NS-2 [202],

REAL [203] and Dummynet [204]. Owing to the complexity of these simulators and the specialised requirement of DTN like policies, credentials and access management, a specialised P2P simulator was developed. The simulator goal was to analyse the DTN model and its associated properties in a peer-to-peer (P2P) environment.

The DTN simulator modelled like a lightweight version of NS-2 was implemented using Java. The simulator included the following components: a scheduler, message objects, nodes, policy evaluator, and protocols. The scheduler is responsible for threads and scheduler queues. The scheduler keeps a queue of threads to be run and executes threads based on a schedule. A message is either a request or a response. The scheduler using a random distribution generates node messages. A Message or process can spun other sub messages. Nodes represent network edges, and protocols were based on the DTN protocols and algorithm described in Chapter 7. The capability of the simulator depends on the amount of resources made available by the host system. The topology of the simulated environment, credentials, trust contracts and COTs were generated using a random distribution for every experiment.

Several experiments were conducted using the simulated peer-to-peer (P2P) environment. The simulator ran on a dual core 2.4GHz Xeon processor machine with 2Gbytes of memory running Scientific Linux OS. A P2P network of 10 to 1,000 nodes with varying degree of overlapping COTs were simulated. In all conducted experiments similar negotiation effects were noticed. In this section results from experiments with 2 to 14 overlapping COTs P2P simulated network are presented, since there were no significant results for higher overlapping COTs.

Peers in the simulator are autonomous, each with unique node properties and capabilities such as services and resources. Each peer in the simulation had a randomly chosen number of credentials with a maximum of 20 in their local security infrastructure, e.g. an LDAP server. The objective behind this maximum value was to provide a highly concentrated pool of credentials from which trust contracts may be randomly generated.

Each peer (node) has randomly chosen nodes in its COT, without any priorities or hierarchies existing between the nodes. Randomly generated trust contracts tc: $1 \leq tc \leq thresh$ were established between each peer and peers that exist in their COT, where $thresh$ was a threshold percentage of credentials in each peer LDAP server. Every peer had a deny rule for any remote credential from non-COT peers and also for any non-tc remote credential from COT peers. The simulator is started when the number of nodes and $thresh$ size are initialised.

For each trust negotiation run, the simulator randomly chooses two peers P_i and P_j, $i \neq j$. P_i initiates a request with its local credential, $C_i^{P_i}$ for P_j's credentials. The request was

made to all peers in P_i's COT. The result of each trust negotiation was recorded at the P_j. 10,000 negotiations runs were divided into 50 rounds for each simulation and results were collated for each simulation. Each data point shown in the Figures 9.6 and 9.7 represents the average of 20 simulations with different random seeds.

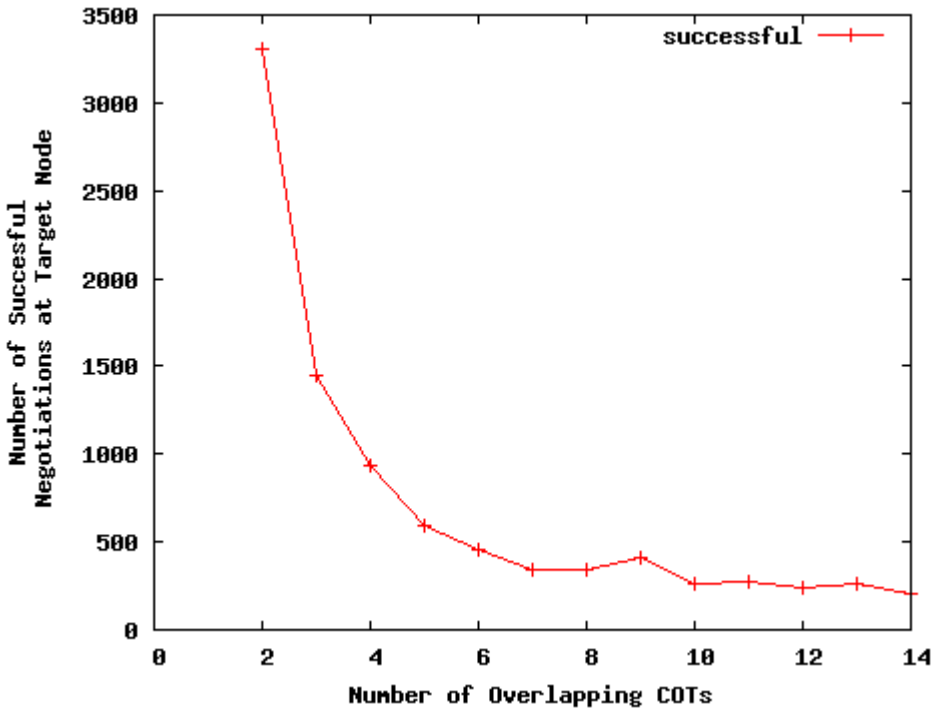


Figure 9.6: Number of overlapping COTs versus number of negotiations at the target node

Figure 9.6 shows the result when the number of COT involved in the trust negotiation increases. The number of successful negotiations at the target fell exponentially to a very low value. Similarly, failed negotiations at the target shows that the number of negotiations reaching the target was rapidly affected by the increase in COT. A 10-30% successful negotiation rate was recorded when the number of overlapping COT was not more than five.

The effect of N-tier delegation hops in the system was also compared. Figure 9.7 shows the effect of number-of-hops in DTN. The results shows that regardless of the number of negotiations, the number of negotiations occurring at a target node do not improve if the number-of-hops (intermediary nodes) involved in negotiations is more than five hops. It shows that successful negotiation is dependent on fewer numbers of hops. These results are in agreement with the expected effect of COT on the system.

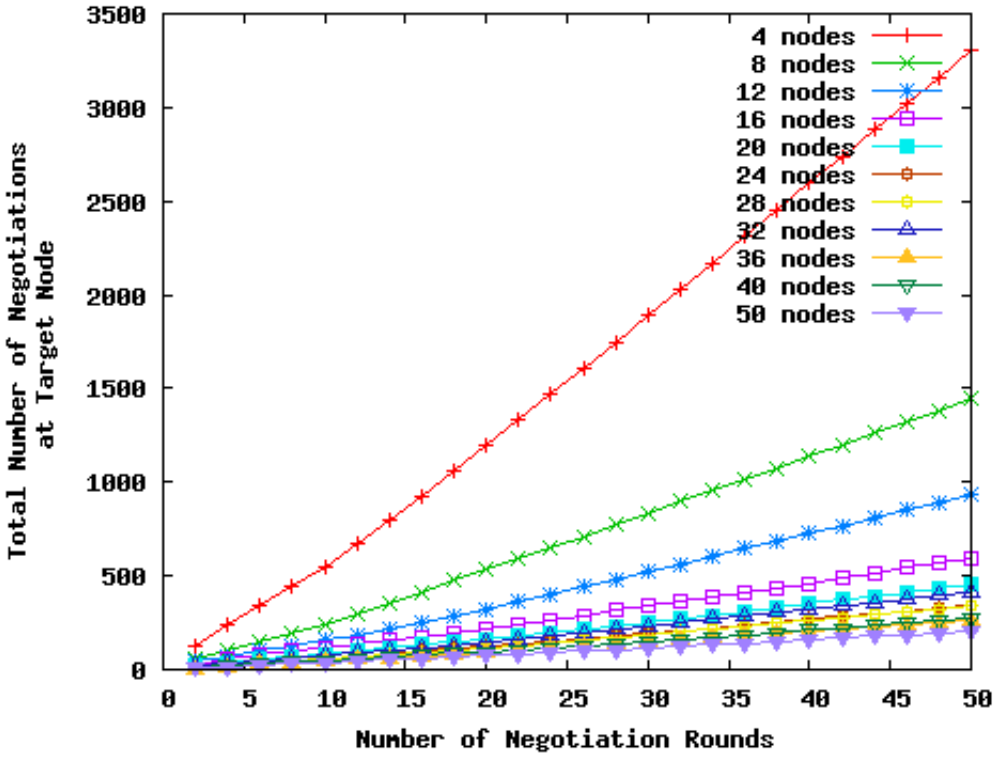


Figure 9.7: Number of negotiation rounds versus number of negotiations at the target node

9.7 DTN Similarities with BGP

The Border Gateway Protocol (BGP) is an inter-autonomous system routing protocol [205]. BGP is used to exchange routing or reachability information with other BGP systems. Internet service providers (ISP) use this protocol to determine a route to a destination. Since multiple paths to a destination can exist, BGP uses various attributes (or properties) to determine the best route. BGP attributes include: weights, local preference, origin and next-hop [205].

As with BGP, in a *circle of trust*, the number-of-hops and size of *trust contracts* are attributes that affect the way DTN performs. The combined effect of these attributes exhibits characteristics that are similar to BGP. DTN similarities with BGP include the following:

- **Weights:** e.g. number of *trust contracts (TC)*, trust levels, and level of risks. In this case the route with the highest weights will be preferred. Since negotiation success is

based on the satisfiability of one or more TC at each hop, more TCs at each node improve the negotiation success rate.

- **Local preferences:** if there are multiple paths to a destination, the local preference attribute is used to select the next node for a particular destination. Local preferences are based on the level of trust, which are often based on past negotiations with a particular node. For example, if a previous negotiation was successful through a particular node, that node would be preferred in subsequent negotiations.
- **Next-hop attribute:** as nodes create trust-pathways [10] to a target node, intermediary nodes that exist in the local COT are identified as next-hops. These intermediary nodes are then considered when a target node is to be reached.

9.8 Performance and Evaluation

The DTN implementation has been simulated and tested in a networked environment. Results from the DTN simulation [11] was discussed in Section 9.6. The main points to observe from the simulation is that DTN performance drops exponentially as the number of overlapping COTs increases. In the simulation, no limit was placed on the number of nodes that could exist in a COT as it would not reflect reality. Placing a limit on the number of nodes in a COT would be similar to limiting the formation of trust relationships, hence unrealistic.

The effect of overlapping $COTs$ is related to the effect of the number-of-hops. Based on the simulation, DTN was tested across four $COTs$ in a networked environment. No limit was placed on the number of nodes in each of the $COTs$, but they were all ultimately constrained by the number of nodes (8 nodes) in the test environment.

The performance of the implementation was tested using several scenarios. Each of the scenarios was tested over several runs on similar network (trust) topologies, each with a total of 8 nodes. Each scenario explores the effect of the different amount of trust relationships, i.e. number of nodes in a COT and the number of TCs between nodes. Each run in a scenario involved varying the node that acts as the sender node and node that acts as the target node. Each run also involved varying the attributes a sender node makes available for trust negotiation.

Each node was hosted on a 2.2Ghz Celeron with 512Mbytes RAM running Linux. All nodes had GT4 installed, which hosted both the discovery and negotiation services. Discovery scenarios involved choosing a node to act as a source node to request the discovery of a

target node by sending route messages to nodes that exists in its *COT*. On average the discovery scenario executed in 43 seconds, ranging from 10 to 65 seconds, for all next-hop nodes to a target node to be discovered by a source node. Negotiation scenarios involved a source node negotiating security attributes with nodes that serve as intermediaries for a target resource. On average the negotiation scenario ran for about 15 seconds, ranging from 5 to 25 seconds, for all negotiation responses to be received the source node. In network terms, the run time or duration was similar to Transmission Control Protocol (TCP) round-trip time (RTT) [206].

The reasons for the spread of duration values between discovery and negotiation processes varies. The discovery process involves a partial multicast of messages on the network compared to the negotiation process that is tightly controlled. It is a partial multicast because not every node on the network will receive a discovery message. However, some nodes will receive a message more than once as route requests for a target node are likely to be sent more than once through multiple nodes. Another associated reason for the variance in discovery duration is due to the fact that the process is a service that runs at the application layer [207]. As such, it is susceptible to application errors and network load factors as nodes are connected over the internet, and other processes running in the application layer.

The reasons for variation in negotiation duration include the network size, i.e. the number of nodes involved in trust negotiations between a sender and target node; the number of negotiation hops and the number of trust contracts that exist between nodes. The combined factor of the number-of-hops and number of trust contracts has the most effect on negotiation duration. Like the discovery service, the negotiation service is somewhat affected by the fact that it runs in the application layer.

A property of the discovery and negotiation protocol is its time-out value. The value determines what is called the negotiation window. The initial time-out value used in DTN is based on the performance tests and is computed to be the sum of the maximum values of discovery and negotiation run times. Periodically nodes may re-adjust their time-out or negotiation window value based on the maximum duration values of discovery and negotiation run times over a period of time. More research needs to be done in this area to determine how best to calculate the value of the discovery time and hence the time-out or negotiation window.

9.9 Summary

This chapter discussed the implementation and testing of DTN in a clinical trial setting. The two systems developed for VOTES were described, and the implementation of DTN in these systems was discussed.

The first system referred to as the VOTES distributed data framework, provided access to multiple data repository including clinical software systems such as SCI-Store. Using Grid middleware services such as OGSA-DAI, it was shown how repositories could be made available in a common web services-oriented format. The chapter described a key component of the system used for patient consent, i.e. how data could be accessed and used. It discussed how the access matrix model discussed in Section 2.2.3 was implemented as an authorisation mechanism and showed how access policies from various sites could be combined to form a single access policy. Another key component of the system described here was a federated data component. This component was designed using a multi-database approach, which supported the decomposition of queries and aggregation of query results. The chapter also argued that the access matrix model offered only a single policy approach and hence was not scalable. It discussed how the ideal approach would be when local resource providers could control and enforce access to their resources in a dynamic manner. It was shown how DTN facilitates this by introducing a negotiation layer, where the local trust policies could be managed by resource managers that grant or deny access to resources based on negotiated attributes.

The chapter also presented the VOTES VANGUARD system. It described how this system provides a secure anonymised data access and linkage model that meets the needs of data providers. In particular it was shown how VANGUARD offers a pull and push model approach, where clinical data providers pull query requests and push query results to Grid based services. It was described why this was necessary, as data providers are wary of allowing direct access to their resources, i.e. through their firewalls. The chapter described how VANGUARD presents an authorisation challenge, since data providers are not willing to yield access control to a centralised authority. It was also outlined how DTN offers a solution in that it provides an underlying trust layer that makes decentralised access control possible.

Finally the chapter presented and discussed the experimental results showing the feasibility, performance, and application of DTN. The chapter concluded with a performance evaluation of the DTN implementation itself.

10 Conclusions, Discussion and Potential Areas of Further Work

A couple of things have been assumed in this work. First, it assumes that a means of authentication exists across e-Health domains either by federated or centralised authentication. Key to these models, and the Grid in particular, is the notion of single sign-on. Secondly, the thesis assumes that a limited trust relationship exists between all the nodes for federated authentication for single sign-on to exist. This implies that nodes are able to identify and communicate with one another from a service (application) layer perspective. The focus of this work is mainly in the area of security attributes and credentials, which acts as a basis for trust realisation and that are useful for authorisation decisions.

This chapter provides an overview of the work described in this dissertation. It presents the conclusions and discussion. Finally it describes potential areas of future work.

10.1 Summary

This chapter has drawn conclusions and identified the main results of the work as a whole. These primarily stem from the design and development of DTN and how it addresses cross-boundary decentralised authorisation issues. This includes how the DTN approach supports trust discovery and trust realisation in e-Health environments. This thesis contribution can be summarised as follows:

- Inter-domain authorisation – DTN offers a novel approach to address inter-domain authorisation challenges. By negotiating trust on the lines of linked trust contracts, authorisation across domains is made possible.
- Access to resources across organisational boundaries – DTN offers the possibility where a non-trusted remote entity can request resources and present credentials that are

acceptable, useable and tenable for local authorisation decisions, to remote and initially non-trusting resources.

- DTN proposes an alternative to the single global attribute ontology approach. Instead of having one large security attribute ontology, many peer-to-peer security attributes can be linked together by means of trust contracts and circles-of-trust, which are able to offer the same benefit as large ontologies, but without the overhead of supporting and maintaining such large ontologies.
- Multiple negotiation hops in trust negotiations – DTN introduces trusted intermediary parties (TIP), which are similar to locally trusted third parties (LTTP) [117] in automated trust negotiation (ATN). Unlike ATN however, more than one TIP can exist in a trust negotiation between two peers. With TIP, multiple negotiation paths can be explored. Apart from providing richer negotiation opportunities they can also increase the chances of a successful trust negotiation.
- The discovery and establishment of trust pathways – Different routing algorithms were investigated in order to address trust pathway discovery and realisation requirements. The AODV algorithm was chosen since it provided a basis for the discovery protocol used in DTN. This protocol makes it possible to discover and establish trust in decentralised security-oriented environments such as e-Health.

10.2 Conclusions and Discussion

This thesis concludes that the designed and developed dynamic trust negotiation (DTN) can address the problem of the discovery and realisation of trust between heterogeneous and autonomous entities, thereby making decentralised authorisation possible in a collaborative environment. DTN differs from ATN in that it introduces trust contracts and trusted intermediary parties. DTN provides multiple negotiation and delegation hops, which protect credentials and access policies in collaborative environments. To support these, DTN explores pair-wise trust relationships that exist between collaborators and ways of discovering trust pathways from these trust relationships.

ARTEMIS reviewed in Section 2.7.3 showed that by mapping a global security attribute ontology to local security attributes ontologies, decentralised access control can be achieved. This thesis argues that instead of having one large security attributes ontology, many small security attributes ontologies can be linked together to support decentralised access control.

It was noted that large security (attributes) ontologies are difficult to create and problematic to change, but the DTN approach presents an alternative to the creation, update and management of large ontologies.

Today negotiations are largely global and static in nature and they reduce the independence and flexibility that each partner has because they try to reach a balance between autonomy and heterogeneity. However, DTN ensures that these negotiations do not have to be global and do not have to be static. This is achieved by:

- collaborating members forming a pair-wise trust relationship with neighbouring partners through exchanging security attributes;
- allowing collaborating members to release security attributes to other neighbouring nodes in their COT;
- collaborating members sharing common and overlapping sets of vocabularies that enable resource requests to be understood and used to trigger trust negotiations;
- collaborating members implementing their own attribute-based or role-based access control model.

The DTN model has been evaluated and analysed in Section 9.8 showing the pros and cons of *circle of trust (COT)* as well as *trust contracts*. The model shows how trust can be discovered and realised between non-trusting entities; how credential semantics between domains can be mapped; and how disclosure of access control policies can be limited. In conclusion, this thesis asserts that through DTN one can discover and realise trust in open and decentralised e-Health environments. The contributions of this thesis include the employment of routing protocols to trust discovery, the application of trust contracts to trust negotiation, and the use of negotiation hops to trust realisation.

10.2.1 Collaboration Issues in e-Health

Many of the challenges facing the e-Health domain are security related, e.g. data integrity and confidentiality. Many of these receive considerable attention since the fallout from security events are often very damaging and severe, not only to patients but also to healthcare providers and the government. In this regard, controlling or restricting access to sensitive data has been a primary concern for healthcare providers. This is not to say that other e-Health security issues such as patient identification, user authentication, data linkage and inference are not being addressed but that access control has been of higher priority. Often

this drive to control access to clinical data has resulted in a form of paranoia, not necessarily in a negative sense but in an overly protective sense. This protectiveness is one reason why data sharing between healthcare providers and researchers has been difficult. DTN has been shown to integrate well with existing security infrastructures in supporting existing security infrastructures since DTN makes it possible to receive and use remote security attributes for local access control decisions, thereby providing support for healthcare providers and their collaborators [208].

The VOTES project illustrates some of the strict security requirements imposed by health providers like the National Health Service (NHS) towards Health research collaborations. Healthcare providers and key stakeholders are concerned about confidentiality and ethical issues since they directly affect the sharing of data across organisational boundaries. In VOTES, cross-boundary concerns span from secure data access to data aggregation and statistical disclosure. Some data providers are not willing to release data as it is feared that data sets, once released, cannot be controlled or monitored. Similarly they are wary that released data sets can possibly be linked with other data sets, which can result in the violation of patient confidentially. The VANGAURD system [195] presents an approach that directly addresses some of these issues. In Section 9.5, DTN was shown to support this approach and provide a trust layer that makes decentralised access control possible as opposed to the centralised VO-agreed security attributes approach.

10.2.2 Access Control in Decentralised Systems

Security in decentralised collaborative environments presents many challenges where entities from different autonomous security domains want to access and share resources. This is largely due to cross-boundary issues where security credentials and policies are heterogeneous, and where yielding control to a centralised authority is not an option. Numerous cross-boundary approaches exist today and trust negotiation remains a promising solution that is rapidly evolving.

One approach is the use of a single access control policy that governs authorisation across domains. This is achieved when all collaborating domains pre-negotiate and agree on privileges amongst other things related to access to shared resources [64]. The implication of this approach includes having detailed knowledge and agreements on global policies potentially comprising numerous local policies; global policy maintenance; initial integration effort and the static relationship between security attributes and credentials. A variation of this approach is delegating authority to remote entities to assign privileges to their (remote) users

[126].

Another approach to the single policy paradigm is based on a shared ontology. In this approach collaborating parties agree on a security ontology that describe roles or privileges that could be used for collaboration. Each collaborating party maps their local security ontology to the shared ontology in order to access remote resources. Remote parties in turn map the shared security ontology onto their local security ontology in order to grant or deny access to their resources [5, 6]. Associated implications of this approach are the high maintenance cost and initial integration effort associated with the global ontology.

Dynamic trust negotiation (DTN) presented in this thesis is an optimal approach that maps remote security credentials into local security credentials through trust contracts, thereby bridging the gap currently making decentralised security policies for multi-domain collaboration so difficult. This approach requires a minimal knowledge of global policies. DTN also has a low maintenance cost over global policies since it allows feasible relationships between security attributes/credentials to be dynamically established and subsequently enforced.

10.2.3 Trust Issues – Discovery and Realisation

Trust is the underlying phenomenon of any security system. Most security systems are designed using security policies, which define and describe what, how and where it is trusted. Trust is built on the concept of limiting expected behaviour [16]. It is associated with an assurance measurement. The level of confidence in limiting behaviour within a security policy determines the level of assurance.

From an authorisation point of view, trust is often currently viewed from a digital credential perspective, that is what should be the expected behaviour of an entity in possession of the digital credentials and security attributes asserted by a remote entity. This implies security attributes are related to trust and more importantly that they are defined with respect to behaviour in a context, e.g. a domain. To be able to tackle trust from an authorisation point of view in a decentralised and open environment, one has to be able to understand, relate and map security attributes across domains.

DTN addresses this as the issue of trust realisation from a security attribute perspective. Through trust contracts, mutual agreements on security attributes among potentially suspicious entities utilising circles of trust and pair-wise relationships, allow foreign security attributes to be mapped to local security attributes. So instead of mapping a local attribute ontology to a global attribute ontology, local attributes can be related and combined together,

and be used in a more flexible and scalable manner.

Another aspect to the trust issue in decentralised and open environments is how to discover trust relationships. Some work has been done in this area [102, 104, 107] and [107] in particular describes a discovery algorithm that can be used to relate credentials that are needed for authorisation decisions in decentralised environments. However, the algorithm requires that related credentials are reachable or accessible. DTN on the other hand works in cases where related credentials are not reachable but where credential relationship's exist. This is particularly necessary in e-Health environments where accessibility is restricted to a trusted few. The DTN discovery algorithm is based on a vector routing algorithm and works within a contained and security driven environment, where nodes only reach out to neighbouring nodes, e.g. nodes in their circle of trust.

10.2.4 Credential Equivalence

One of the key challenges in this work is with regard to credential equivalence, i.e. how to relate or map between security attributes that are asserted with credentials, especially where a security attribute may originate, be asserted and issued by foreign and autonomous entities. This is essential to address the cross-boundary challenges of collaboration. In this thesis, three equivalence rules were presented: a transitive rule; a linking delegation rule, and an intersection rule. The transitive rule says that if domain A trusts domain B and B trust domain C then A trust C. Without such transitivity, it would be difficult to link trust contracts across domains.

The linking of trust contracts is the basis of DTN's negotiation and delegation hops. The linking delegation rule when applied to trust contracts and security credentials, i.e. asserted security attributes, makes it possible to deduce relationship that may exist between credentials. This rule says that credential B is equivalent to credential A if B can be deduced to relate to credential C.

The intersection rule says that credential equivalence may be deduced based on commonality between a collection of credentials. The rule highlights the relationship that may exist between a credential and a collection of credentials. It is based on these rules that DTN trust contracts are modelled. These rules are derived from work done by [107, 108, 119] for trust negotiation.

10.2.5 Security policies

Four types of security policies were presented for DTN to support trust negotiation across boundaries. These policies show that potential risks associated with the notion of transitive trust can be reduced if properly designed and implemented. These policies ensure that trust contracts, which themselves are sensitive and can be regarded as privileged information, could be protected with policies and thus used to control the invocation of trust contracts. Similarly, by using these policies, privileges associated with trust contracts can be constrained based on who is invoking the contract and for whom it is being invoked. With these policies in place, various aspects of trust negotiation can subsequently be controlled and managed. This includes controlling the release of credentials to remote domains in scenarios where the domain is participating in a trust negotiation as an intermediary domain.

10.2.6 Trust Negotiation and Policy Disclosure

Automated trust negotiation (ATN) is one of the promising approaches in the area of trust realisation, which enables entities including strangers to access resources across autonomous boundaries through the iterative exchange of credentials. Various negotiation strategies have been proposed for ATN to protect credential disclosure during trust negotiations. However, in some domains such as e-Health, not all entities are willing to negotiate credentials or disclose access control policies directly to strangers irrespective of trust negotiation strategies. Instead they prefer to negotiate and disclose sensitive information only to those entities existing within a trust community (or in dynamic trust negotiations (DTN) through a *circle of trust*). DTN proposes a model that not only protects sensitive information from disclosure but also reduces semantic issues that exist with credentials in decentralised systems. In the model, the process whereby trusted intermediary parties act as links between communities and provide multiple negotiation and delegation hops, as well as protects the disclosure of access control policies including for example a service provider's access control policies.

However, as the network reaches a stable state, i.e. as DTN factors settle over a period of time, the property of the system where access policies are protected can be inadvertently reduced. As factors such as trust contracts and COT memberships remain constant over time, foreign entities will be able to infer a service provider's access control policies. For example, in the e-Health environment, there is possibility that existing trust-contracts over a period of time can be inferred since TCs may not change or be revoked. This is especially so in the Health domain where organisations maintain their existing trust relationships and

nodes are not joining, leaving or changing their COT.

10.3 Potential Areas of Future Work

There are several areas where the work described in this thesis could potentially be further investigated or improved.

Firstly, the discovery protocol described in this thesis was implemented as a service in the application layer [207]. This allowed it to integrate with existing systems and subsequently share data between them. However, it could well be the case that the discovery protocol described in section Section 7.2 would be able to function in the network layer as well, as is common with routing protocols. This would require investigations in the network layer space, but hypothetically in addition to feasibility acceptability challenges, it could present other challenges such as trust information which would subsequently need to be shareable with services that exist in the application layer.

Secondly, other ways of improving performance of the framework described in this thesis would need to be studied. Two ways of possibly improving the performance of DTN includes peer clustering and exploiting RBAC hierarchies. Peer clustering is similar to route reflector techniques of Border Gateway Protocol (BGP) systems. This method would entail investigating ways of grouping nodes together and having super nodes, which potentially will reduce the number of hops needed for trust discovery and realisation. Another way of possibly improving DTN negotiation performance could be through RBAC role hierarchies [42]. The effect of role hierarchies was not investigated in this thesis but potentially, if it can be applied in the context of trust contracts, it could significantly increase trust negotiation success rates.

Thirdly, more work needs to be done in the area of credential equivalence. Much work is currently being done in the area of matching and mapping domain ontologies [125, 6]. However, in the area of security attributes and contexts, applying ontological mapping techniques would be challenging since ontology mapping and matching techniques are still largely in their infant stages. In addition, the health domain is itself complex with a vast and wide spectrum of roles and access attributes applicable to myriad resources. It will offer significant challenges and case studies for ontology mapping. Work is on-going in this area in the health domain with significant progress in the areas of clinical ontologies like SNOMED-CT [190] and ICD-10 [192].

Fourthly, further work is needed in the area of trust realisation so as to reduce to a barest minimum the security challenges of decentralised collaborations. A more robust algorithm could be developed or investigated to achieve this since the approach described in this thesis is not the only method and arguably is not the best method in all situations. In order to achieve this, trust would need to be defined and modelled differently. It may also have to be combined with advances made in the area of ontologies for credentials used in access control.

A hybrid combination of DTN and ontologies exploiting push and pull of security attributes, or combining centralised and federated DTN approaches are other areas of applications research. Finally the work described here has focused largely on the e-Health domain. The application of DTN could equally be applied to other domains which would likely raise their own requirements that would need to be addressed.

APPENDIX

A XACML Policies for DTN

This appendix provides an outline and example of some of the XACML policies that have been implemented to explore DTN.

A.1 Extract 1

An extract of a policy that allows an Investigator to select records from PatientMaster table only with obligations that log access and anonymise certain fields is shown in Figures A.1 – A.2:

A sample request for the policy shown in Figures A.1 – A.2 is shown in Figure A.3. The request when evaluated will yield a deny response as Richard is not allowed to perform insert or update on PatientMaster.

A.2 Extract 2

An extract of a trust-contract policy that allows a remote Investigator to perform select but not insert or update a PatientMaster table is shown in Figures A.4 – A.5:

A sample request for the policy shown in Figures A.4 – A.5 is shown in Figure A.6. The request when evaluated will yield a permit response.

A.3 Extract 3

The XACML extract shown in Figures A.7 – A.10, show how two mutually exclusive trust-contract policies can be resolved.

A sample request for the policy shown in Figures A.7 – A.10 is shown in Figure A.11. The request when evaluated will yield a deny response as both roles are mutually exclusive.

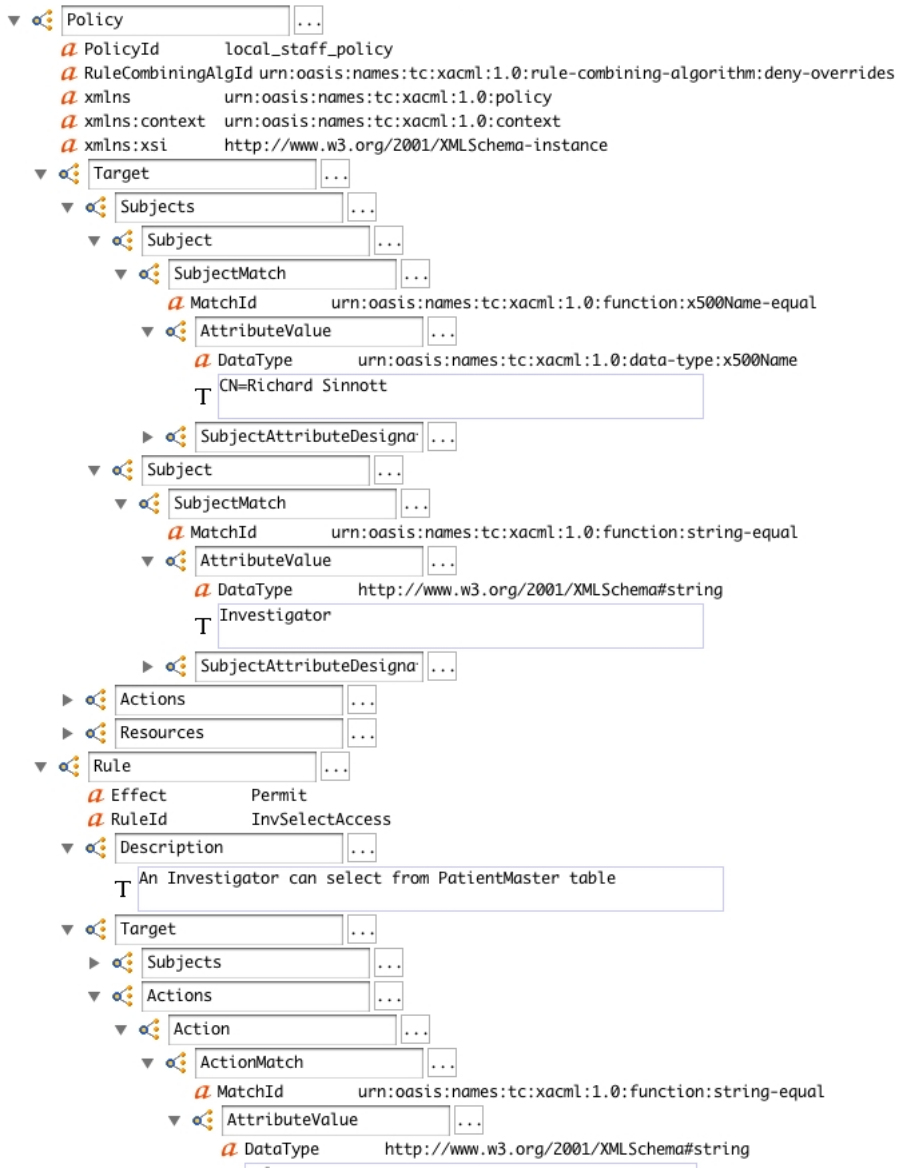


Figure A.1: An extract of a local policy with obligation – Part 1

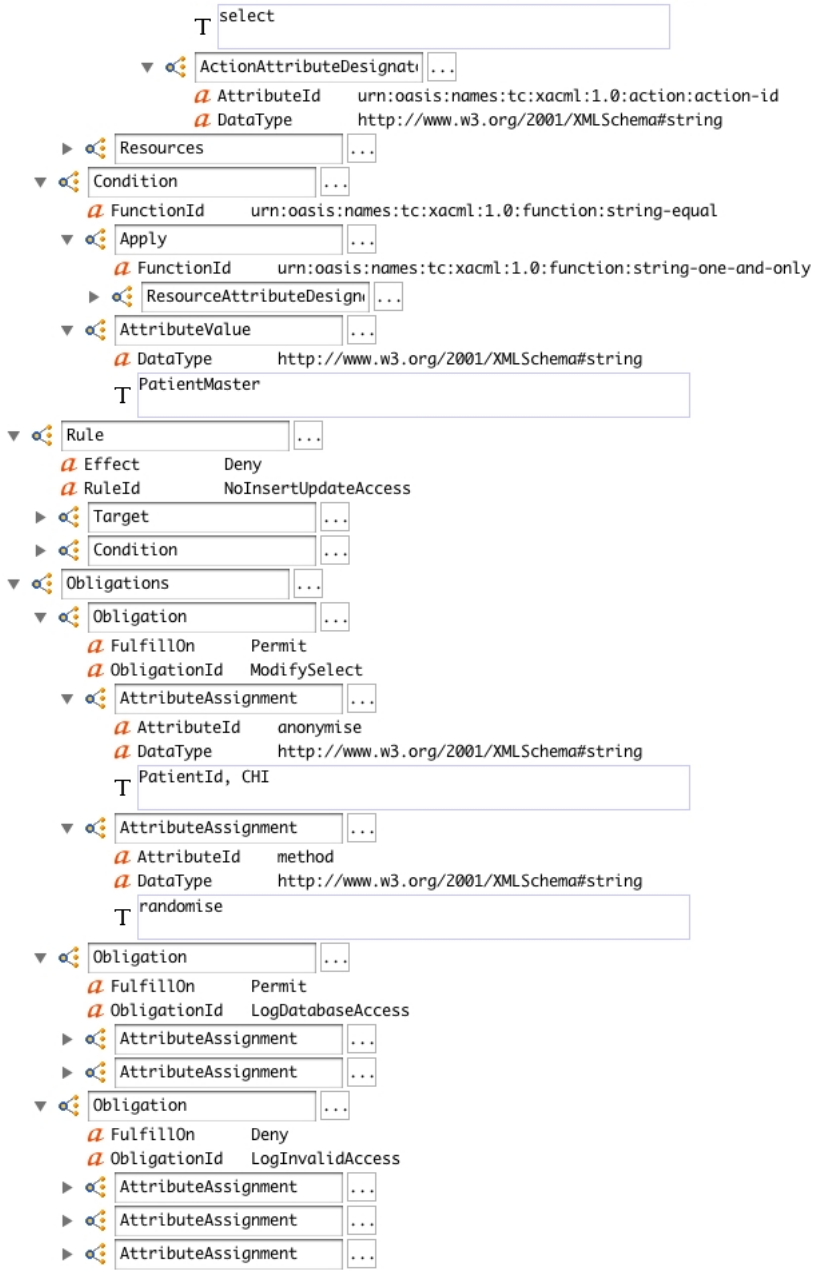


Figure A.2: An extract of a local policy with obligation – Part 2

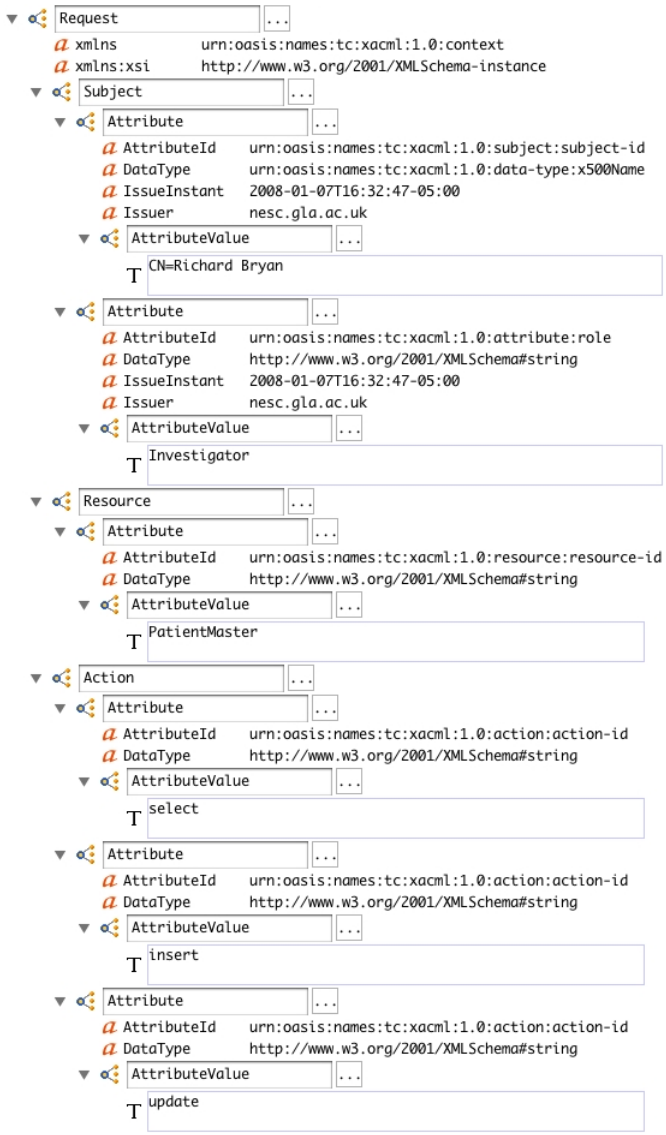


Figure A.3: A request from a local entity

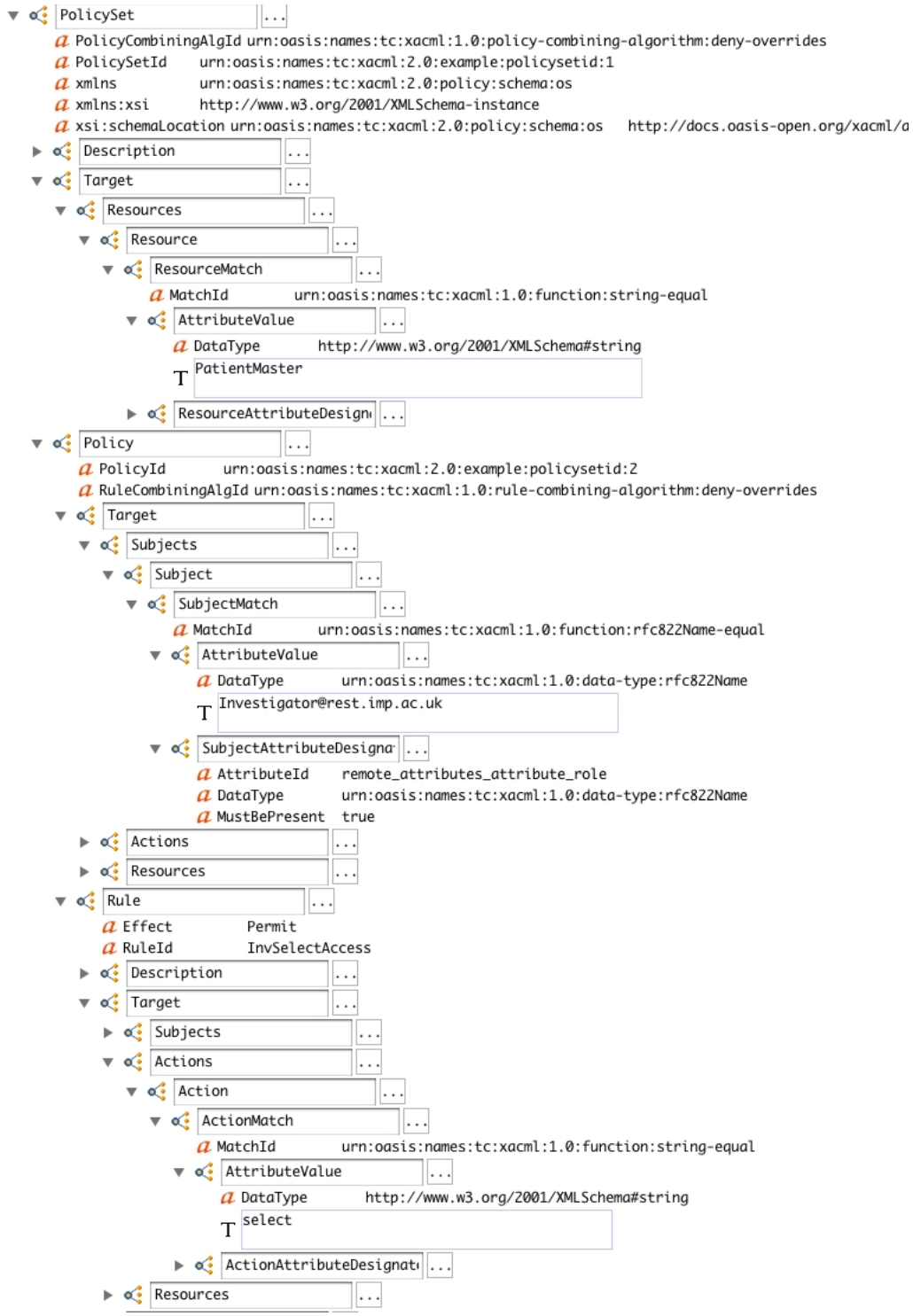


Figure A.4: A trust-contract policy - Part 1

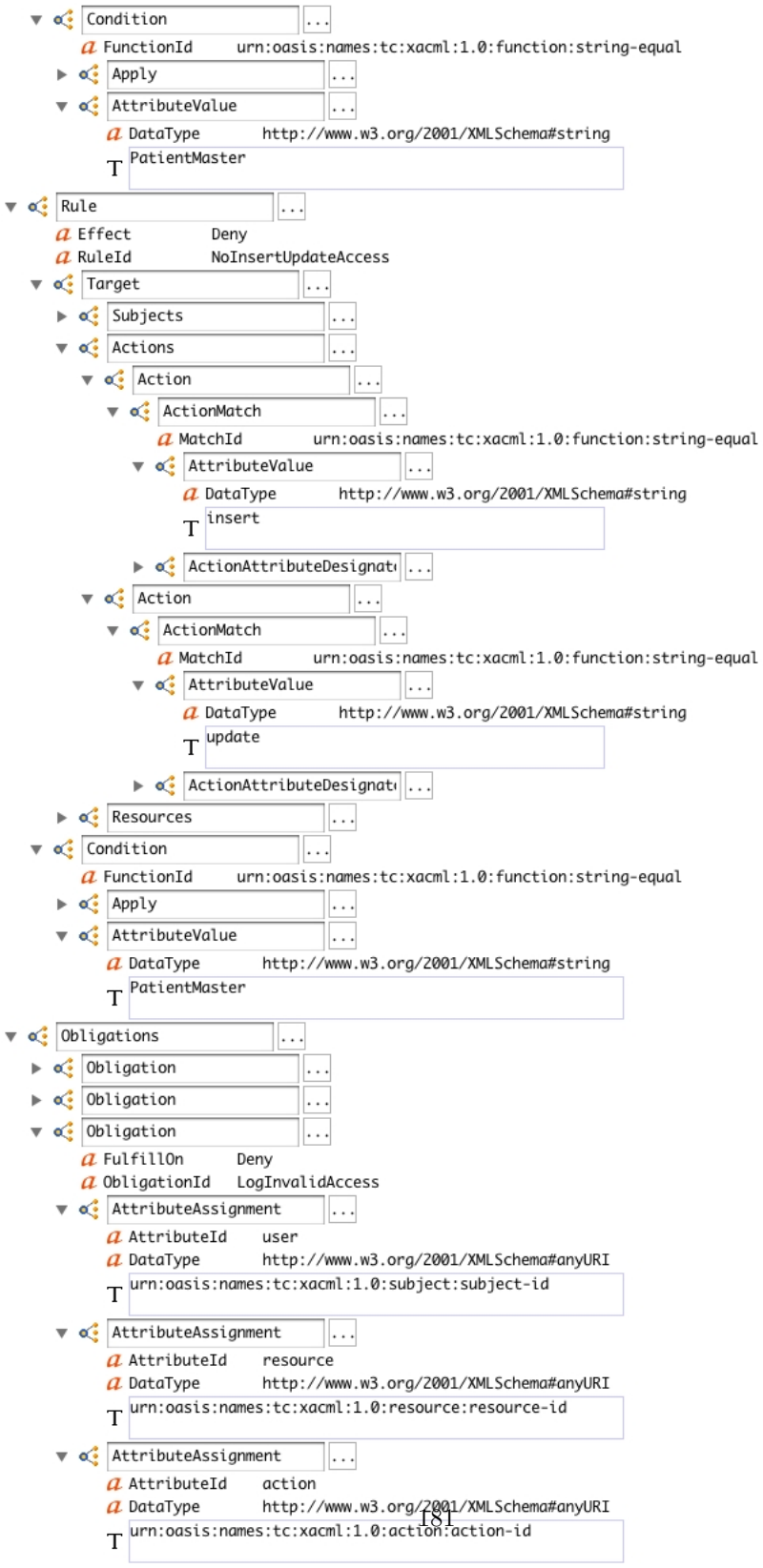


Figure A.5: A trust-contract policy - Part 2

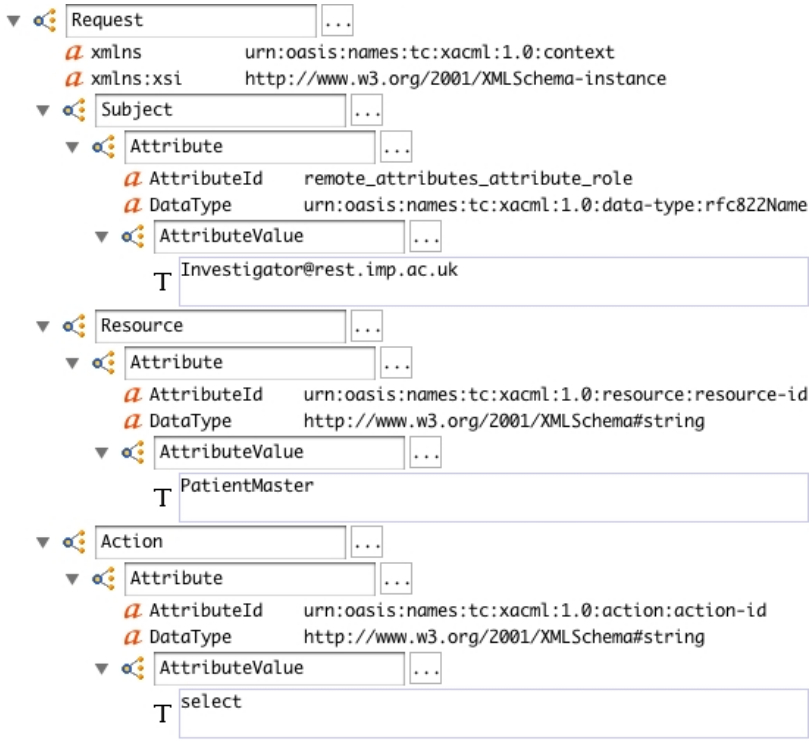


Figure A.6: A trust contract invocation request

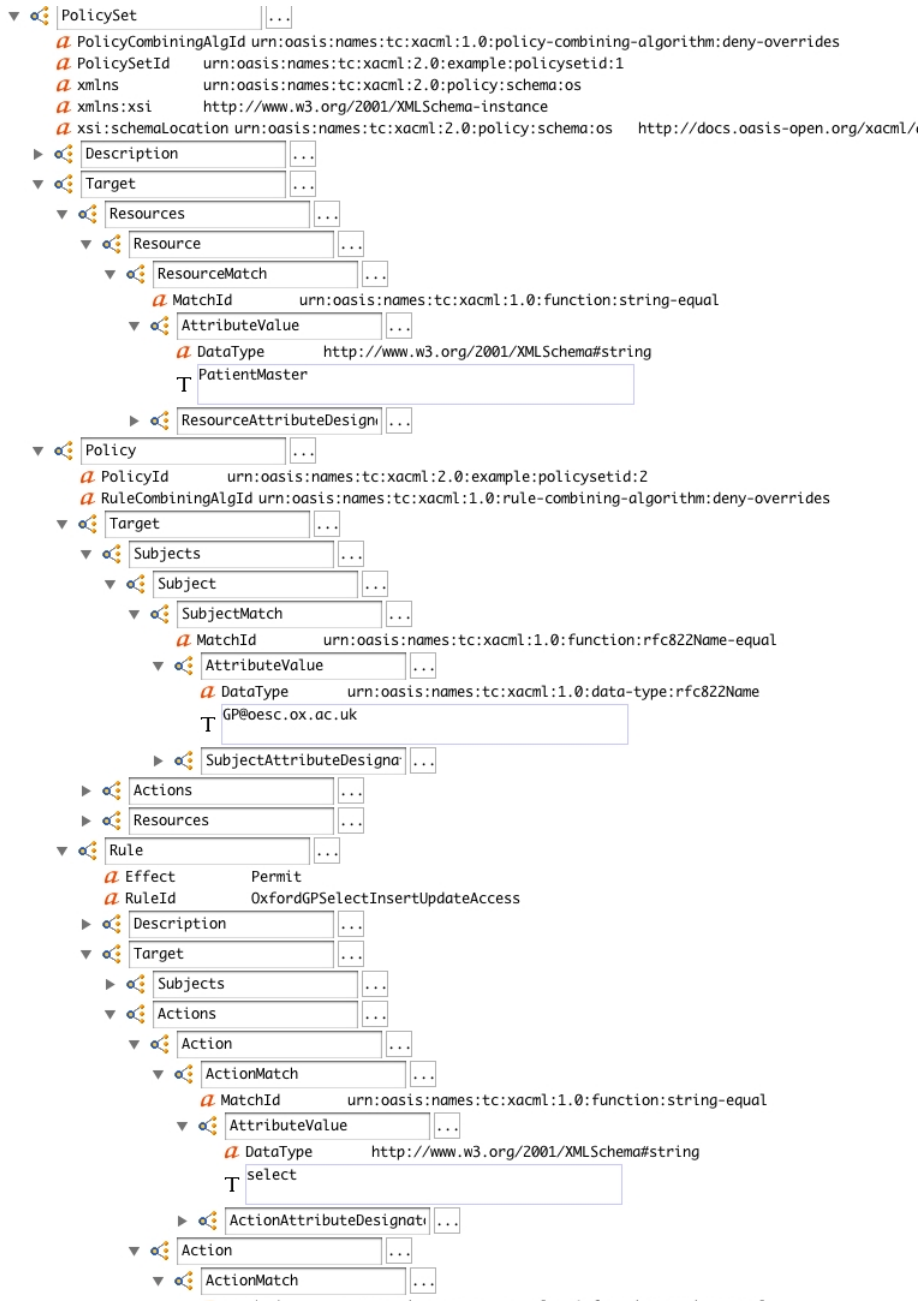


Figure A.7: Mutually exclusive trust-contract policies - Part 1

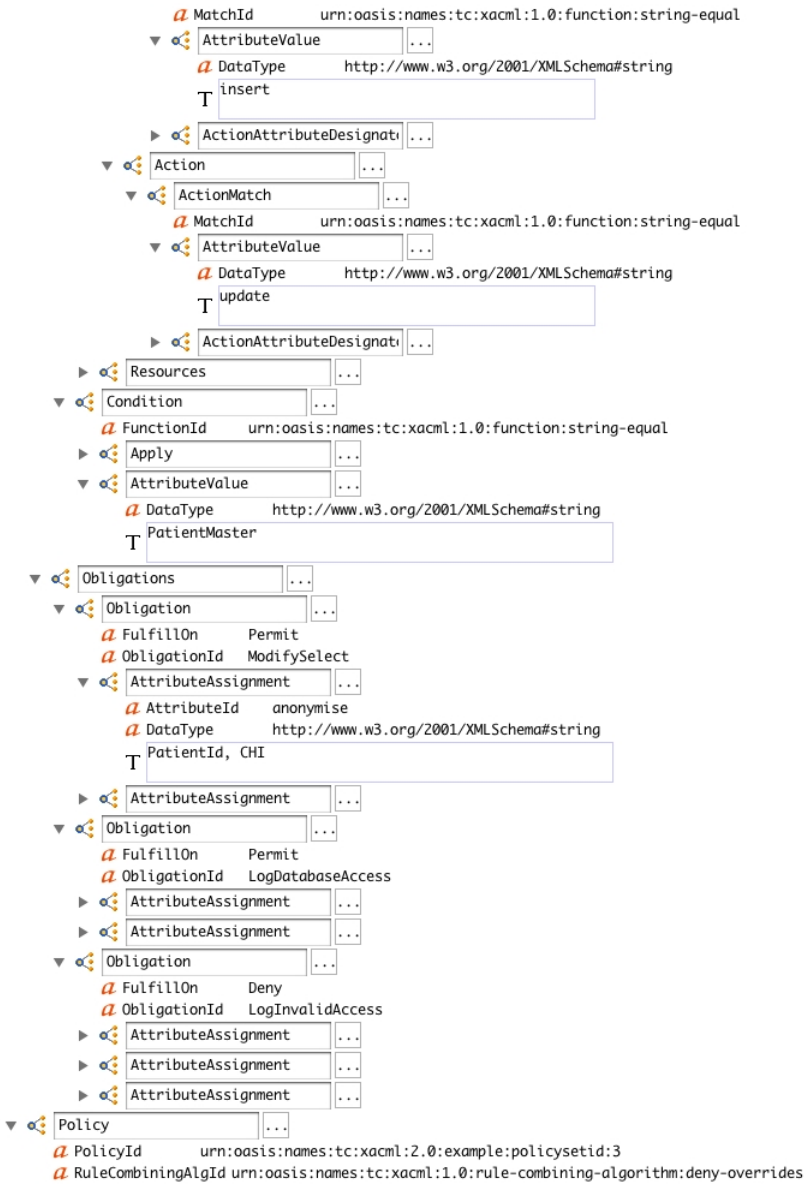


Figure A.8: Mutually exclusive trust-contract policies - Part 2

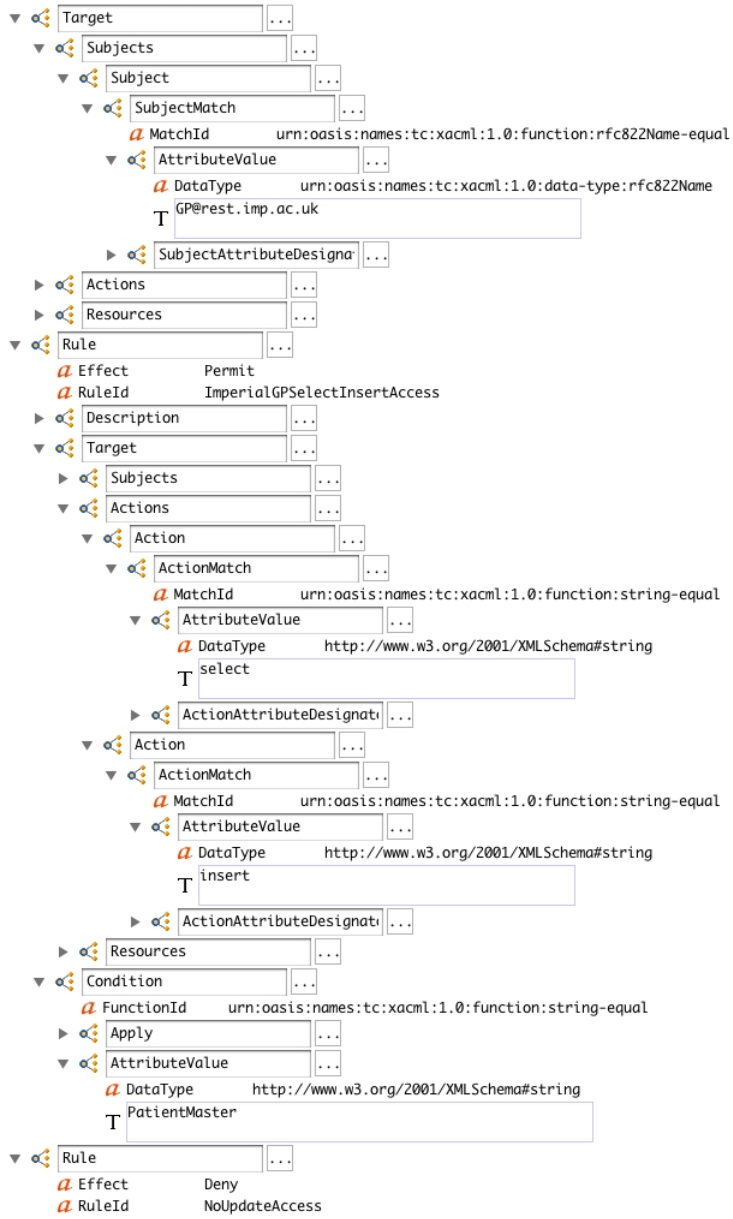


Figure A.9: Mutually exclusive trust-contract policies - Part 3

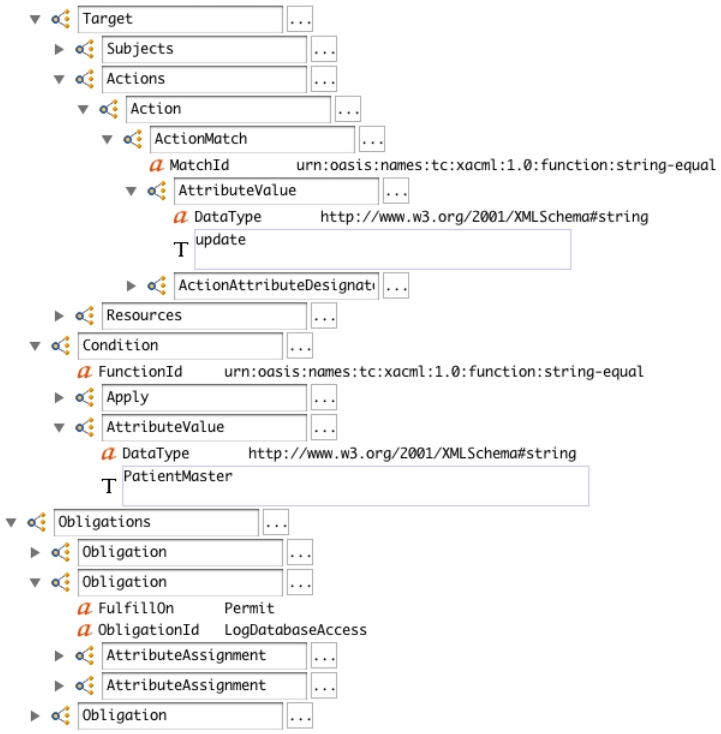


Figure A.10: Mutually exclusive trust-contract policies - Part 4

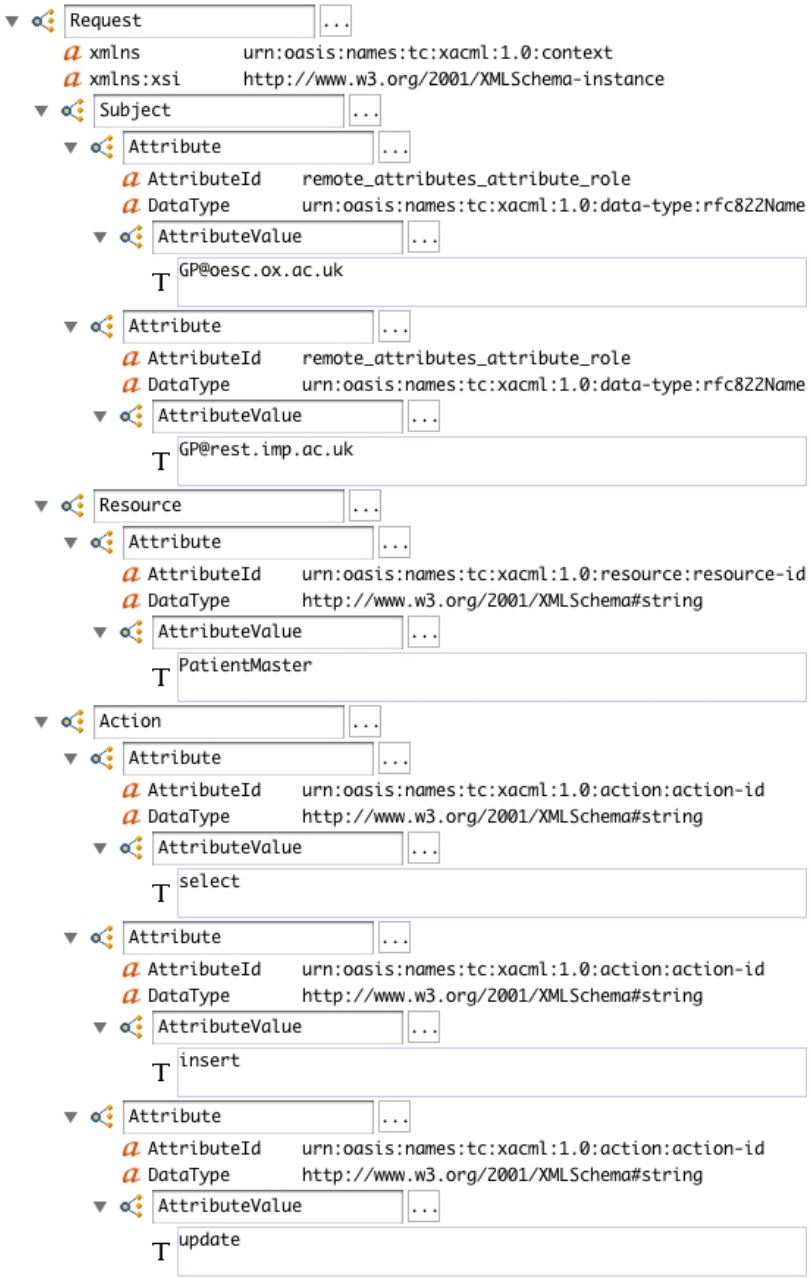


Figure A.11: A request that causes multiple-path conflict

B Background to Clinical Trial Phases and Terminology

This appendix briefly summarises the typical phases associated with clinical trials, as well as identifying and defining terms often used.

B.1 Clinical Trial Controls and Terms

Institutional Review Boards (IRBs): An IRB is typically a group of doctors, lawyers, clergymen and community members, formed to review and approve applications for clinical trials in order to protect the rights of participants. The IRB monitors trials and also has the authority to terminate on-going trials if the need arises.

Data and Safety Monitoring Boards (DSMBs): A DSMB is an independent body that reviews trial data and suggest changes to the trial process / design if necessary, e.g. disadvantages to one or more groups in a trial. DSMBs can also recommend that a trial be terminated.

Informed Consent: This is a voluntary consent given by a trial participant to show that they have understood all aspects of their involvement in a particular trial including potential risks. An informed consent form is provided by the research body and submitted to the IRB for review. The participant must be provided with information such as trials purpose, risks, treatment, benefits, procedures schedule and IRB contact information.

Placebo: Is a substance used as a trial medication which has no effect because it does not contain an active drug. It is used for comparison with an active drug to see if the active agent (drug) has any effect on the diseases. Not all trial use placebos.

Randomisation: Is a selection process where participants are anonymously divided into two groups to decide who gets an active agent or a placebo.

Double-blinded: Is a trial that use placebos, where neither the research staff nor trial participant knows which treatment is assigned to which individual in the trial.

Open-label: Is a trial where both the research staff and participant know which treatment is assigned to which individual in the trial.

Endpoints: Is a measurable event or outcome that can be used to determine if the trial or study is of benefit. A clinical trial endpoints are usually defined in the study objectives.

B.2 Clinical Trials Phases

	Phase I	**Phase II**	**Phase III**	**Phase IV**
Question	Is it a safe treatment?	Does the treatment work?	What are the long-time results in lots of people?	What is the long-term safety information?
Volunteers	Usually healthy volunteers or without comparison groups	Volunteers with the disease or condition	Both healthy & non-healthy volunteers	
Goal	Determine best doses	Get safety information	More information on safety and effectiveness	Get information about very rare side effects
Risk	High Risk	Moderate Risk	Low risk	Very low risk
Length	Few weeks/months	About a year	Up to three years	May last for years
Size	Few participants	About a hundred participants	Some hundreds of participants	Some thousands participants
Time	Before drug approval			After drug has been approved

Table B.1: Common Phases of Clinical Trials

C Datalog-based DTN Policy Function Descriptions

In the codes shown, comments are started with a '%' symbol. A statement or assertion ending with a '.' is a clause. Each term in the parenthesis of a clause is either a constant or logical variable. A logical variable is a term that begins with a capital letter. "A clause is a head literal followed by an optional body. A body is a comma separated list of literals. A clause without a body is called a fact, and a rule when it has one. The punctuation ':-' separates the head of a rule from its body. A clause is safe if every variable in its head occurs in some literal in its body" [166].

C.1 $assign(e.p(\vec{e}), r, p)$

Require: e, e_1:entity; $acts$:actions; obj:resource; r, r_1:role; $obj(acts)$:permission
Ensure: $assign(e.p(\vec{e}), r_2, obj(acts)) \leftarrow can\_activate(e, r_2), has\_invoked(e_2, r)$,
$obj(acts) \in permissions(r)$
where $p(\vec{e})$ is $can\_activate(e, r_2)$
% Let e, e_2 be instances of Entity
% Let o, n be instances of Object
% Let a, b, c be instances of Action
% Let q, r, r_2 be instances of Role
% Let $assign(U, V, W, X, Y, Z)$ represent $assign(e_2.p(\vec{e}), r, obj(acts))$
$can\_activate(e, r_2)$. % assertion communicated as a credential
$has\_invoked(e_2, r)$. % transient assertion
$permissions(r, b, n)$. % local assertion to represent $n(b) \in permissions(r)$
$permissions(q, a, n)$. % local assertion to represent $n(a) \in permissions(q)$
$permissions(q, a, o)$. % local assertion to represent $o(a) \in permissions(q)$
$permissions(r, a, o)$. % local assertion to represent $o(a) \in permissions(r)$
$permissions(r, b, o)$. % local assertion to represent $o(b) \in permissions(r)$
$permissions(r, c, o)$. % local assertion to represent $o(c) \in permissions(r)$

$assign(U, V, W, X, Y, Z)$:- $can\_activate(U, X)$, $has\_invoked(V, W)$, $permissions(W, Y, Z)$.
$assign(V, V, W, W, Y, Z)$:- $can\_activate(V, X)$, $permissions(W, Y, Z)$.

Note: U, V, W, X, Y, Z are logical variables. The rule implies U is assigned W and granted permission $Z(Y)$ if a $can\_activate(V, X)$ or $has\_invoked(V, W)$ assertion can be evaluated to be true.

C.2 $can\_activate(e, r)$

Require: e:entity; r:role
Ensure: $can\_activate(e, r) \leftarrow has\_activated(e, s)$
% Let e, f be instances of Entity
% Let r, s be instances of Role
$has\_activated(e, r)$.
$has\_activated(e, s)$.
$has\_activated(f, r)$.

$can\_activate(X, Y)$:- $has\_activated(X, s)$, $Y = r$.

Note: X, Y are logical variables. The rule implies X can activate Y if X has already activated role s. Usually $can\_activate(X, Y)$ is an assertion made by a domain and passed on to other domains in form of a credential.

C.3 *has_activated*(e, r)

Require: e:entity; r:role
Ensure: *has_activated*$(e, r) \leftarrow$ *isActive*(e, r)
% Let d, e, f be instances of Entity
% Let r, s be instances of Role
isActive(e, r). % mapping function that returns true if entity e has role r active
isActive(f, s).. % mapping function that returns true if entity f has role s active
isActive(f, r).. % mapping function that returns true if entity f has role r active
isActive(d, s).. % mapping function that returns true if entity d has role s active

has_activated(X, Y) :- *isActive*(X, Y).

Note: X, Y are logical variables. The rules implies X has already activated Y if in the domain for Y, X has role Y active.

C.4 *can_invoke*$(e_1.p(\vec{e}), r)$

Require: e_1:entity; r, r_1:role
Ensure: *can_invoke*$(e_1.p(\vec{e}), r) \leftarrow tc(r_1)$ % where *can_activate*(e, r_1) represents $p(\vec{e})$
% Let e be instance of Entity
% Let r, s be instances of local Role
% Let r_1, r_2 be instances of remote Role
contract(r_1, r). % mapping function that returns true if $tc(r_1) = r$
contract(r_2, r).
can_activate(e, r_1). % assertion communicated as a credential

can_invoke(Z, X, Y) :- contract(X, Y), can_activate(Z, X).

Note: Z, X, Y are logical variables. The rule implies Z can invoke contract for Y if Z has an assertion that says it can activate X and if a trust contract exists that map X to Y.

C.5 *has_invoked*(e, r)

Require: e:entity; r:role
Ensure: *has_invoked*$(e, r) \leftarrow$ *inSession*(e, r)
% Let e, f be instances of Entity % Let r, s be instances of Role
inSession(e, r). % mapping func. returns true if e had invoked contract for r in the current session
inSession(f, s).

has_invoked(X, Y) :- inSession(X, Y).

Note: X, Y are logical variables. The rules implies X has activated Y if in the current session state, X had invoked contract for Y.

C.6 *can_release_cred*$(e_1, e_2.p(\vec{e}), e_3)$

Require: e_1, e_2, e_3:entity; r:role
Ensure: *can_release_cred*$(e_1, e_2.p(\vec{e}), e_3) \leftarrow$ *has_invoked*(e_3, r), *cot*(e_1, e_2)
Or *can_release_cred*$(e_1, e_2.p(\vec{e}), e_3) \leftarrow$ *has_activated*(e_3, r), *cot*(e_1, e_2)
% Let *canReleaseCred*(X, Y, R, Z) represent *can_release_cred*$(e_1, e_2.p(\vec{e}), e_3$
% where *can_activate*(e_1, r) represents $p(\vec{e})$
has_invoked(e_3, r).
has_activatde(e_3, r).
cot(e_1, e_2).

can_release_cred(X, Y, R, Z) :- *has_activated*(Z, R), *cot*(X, Y).
can_release_cred(X, Y, R, Z) :- *has_invoked*(Z, R), *cot*(X, Y).

Note: X, Y, R, Z are logical variables. The rule implies R can be released to Y if Y is in X's COT and if a *has_activated*(Z, R) or *has_invoked*(Z, R) assertion can be evaluated to be true.

Bibliography

[1] R. S. Sandhu and P. Samarati, "Access Control: Principles and Practice," *IEEE Communications Magazine*, vol. 32, no. 9, pp. 40–48, 1994.

[2] M. Y. Becker, "Cassandra: Flexible Trust Management and its Application to Electronic Health Records." Published as Technical Report UCAM-CL-TR 648, University of Cambridge, Computer Laboratory, Oct. 2005.

[3] G. Aslan and D. McLeod, "Semantic Heterogeneity Resolution in Federated Databases by Metadata Implantation and Stepwise Evolution," *The VLDB Journal*, vol. 8, no. 2, pp. 120–132, 1999.

[4] A. P. Sheth and J. A. Larson, "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases," *ACM Comput. Surv.*, vol. 22, no. 3, pp. 183–236, 1990.

[5] M. Boniface and P. Wilken, *ARTEMIS: Towards a Secure Interoperability Infrastructure for Healthcare Information Systems*, pp. 181–189. From Grid to Healthgrid, IOS Press, 2005.

[6] M. Ehrig and Y. Sure, "Ontology Mapping - An Integrated Approach," in *Proceedings of the First European Semantic Web Symposium*, vol. 3053 of *Lecture Notes in Computer Science*, pp. 76–91, Springer Verlag, MAY 2004.

[7] S. Langella, S. Oster, S. L. Hastings, F. Siebenlist, T. M. Kurc, and J. H. Saltz, "Enabling the Provisioning and Management of a Federated Grid Trust Fabric," in *6th Annual PKI R&D Workshop Gaithersburg, Maryland*, Dec. 2007.

[8] C. V. Oorschot and S. Stubblebine, "Countering Identity Theft Through Digital Uniqueness, Location Crosschecking and Funneling," in *Financial Cryptography and Data Security, 9th International Conference, FC 2005*, 2005.

[9] R. Sinnott, A. Stell, and O. Ajayi, "Initial Experiences in Developing e-Health Solutions

across Scotland." Workshop on Integrated Health Records: Practice and Technology, Edinburgh, Mar. 2006.

[10] O. Ajayi, R. Sinnott, and A. Stell, "Formalising Dynamic Trust Negotiations in Decentralised Collaborative e-Health Systems," in *Proceedings of the 2nd International Conference on Availability, Reliability and Security, (ARES07), Vienna, Austria*, IEEE Computer Society, Apr. 2007.

[11] O. Ajayi, R. Sinnott, and A. Stell, "Trust Realisation in Multi-domain Collaborative Environments," in *Proceedings of 6th IEEE International Conference on Computer and Information Science, ICIS'07*, IEEE Computer Society, July 2007.

[12] S. D. Vimercati and P. Samarati, "Access Control in Federated Systems," in *NSPW '96: Proceedings of the 1996 Workshop on New Security Paradigms*, (New York, NY, USA), pp. 87–99, ACM Press, 1996.

[13] R. Sinnott, D. W. Chadwick, T. Doherty, D. Martin, A. J. Stell, G. Stewart, L. Su, and J. Watt, "Advanced Security for Virtual Organisations: Exploring the Pros and Cons of Centralised vs Decentralized Security Models," in *8th IEEE International Symposium on Cluster Computing and the Grid (CCGrid), Lyon, France, May 2008*, May 2008.

[14] W. Stallings, *Network Security Essentials: Applications and Standards*. Prentice Hall Pearson Education Inc., 2003.

[15] P. Windley, *Digital Identity*. O'Reilly, 12, Aug. 2005.

[16] M. Benantar, *Access Control Systems: Security, Identity Management and Trust Models*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[17] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1." Request for Comments RFC 4346, April 2006.

[18] S. Shirasuna, A. Slominski, L. Fang, and D. Gannon, "Performance Comparison of Security Mechanisms for Grid Services," in *GRID '04: Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing*, (Washington, DC, USA), pp. 360–364, IEEE Computer Society, 2004.

[19] R. L. Rivest, "The MD5 Message-Digest Algorithm, Internet request For Comments RFC1321." http://www.ietf.org/rfc/rfc1321.txt, 1992. (Visited December 2008).

[20] "National Institute of Standards and Technology, Secure Hash Standard (SHS), FIPS

PUB 180-1." http://www.itl.nist.gov/fipspubs/fip180-1.htm, 1995. (Visited December 2008).

[21] S. Castano, M. Fugini, G. Martella, and P. Samarati, *Database Security*. ACM Press/Addison-Wesley Publishing Co., 1994.

[22] J. Domingo-Ferrer, ed., *Inference Control in Statistical Databases, From Theory to Practice*, Lecture Notes in Computer Science, Springer, 2002.

[23] M. Handley and E. Rescorla, "Internet Denial-of-Service Considerations." http://www.ietf.org/rfc/rfc4732.txt, 2006. (Visited August 2008).

[24] R. Housley and T. Polk, *Planning for PKI: Best Practices Guide for Deploying Public Key Infrastructure*. New York, NY, USA: John Wiley & Sons, Inc., 2001.

[25] W. T. Polk, N. E. Hastings, and A. Malpani, "Public Key Infrastructures that Satisfy Security Goals," *IEEE Internet Computing*, vol. 7, no. 4, pp. 60–67, 2003.

[26] "Internet x.509 Public Key Infrastructure Certificate and CRL Profile; RFC2459." http://www.ietf.org/rfc/rfc2459.txt, 1998. (Visited December 2008).

[27] "National Institute of Standards and Technology (NIST), Data Encryption Standard (DES), FIPS PUB 46-3." http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf, 1999. (Visited December 2008).

[28] B. Schneier, "Description of a New Variable-Length Key, 64-bit Block Cipher (Blowfish)," in *Fast Software Encryption, Cambridge Security Workshop*, (London, UK), pp. 191–204, Springer-Verlag, 1994.

[29] "National Institute of Standards and Technology (NIST), Advanced Encryption Standard (AES), FIPS PUB 197." http://www.csrc.nist.gov/publications/fips/fips197/fips-197.pdf, 2001. (Visited December 2008).

[30] "ITU-T Recommendation X.509 — ISO/IEC 9594-8: Information Technology Open Systems Interconnection the Directory: Public-key and Attribute Certificate Frameworks," 3, May 2001.

[31] D. K. Smetters and G. Durfee, "Domain-Based Administration of Identity-Based Cryptosystems for Secure Email and IPSEC," in *Proceedings of the 12th USENIX Security Symposium, Washington, DC, USA*, 4, Aug. 2003.

[32] B. W. Lampson, "Protection," in *5th Princeton Conference on Information Sciences and Systems*, pp. 437–443, Princeton, NJ, 1971.

[33] D. E. Denning, "A Lattice Model of Secure Information Flow," *Communications of the ACM*, vol. 19, no. 5, pp. 236–243, 1976.

[34] R. A. Karger, "Improving Security and Performance for Capability Systems." Published as Technical Report No. 149, University of Cambridge (PhD Thesis), Computer Laboratory, Oct. 1988.

[35] B. W. Lampson, "A Note on the Confinement Problem," *Communications of the ACM*, vol. 16, no. 10, pp. 613–615, 1973.

[36] I. T. Foster, C. Kesselman, G. Tsudik, and S. Tuecke, "A Security Architecture for Computational Grids," in *ACM Conference on Computer and Communications Security*, pp. 83–92, 1998.

[37] The Globus Security Team, "Globus Toolkit Version 4 Grid Security Infrastructure: A Standard Perspective," 8, Dec. 2004.

[38] V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, and S. Tuecke, "Security for Grid Services," in *Proceedings of 12th IEEE International Symposium on High Performance Distributed Computing*, pp. 48–57, June 2003.

[39] "An Internet Attribute Certificate Profile for Authorization; RFC3281." Public-Key Infrastructure (X.509) Working Group, http://www.ietf.org/rfc/rfc3281.txt, 2002. (Visited December 2008).

[40] D. W. Chadwick and A. Otenko, "The PERMIS x.509 Role Based Privilege Management Infrastructure," in *SACMAT '02: Proceedings of the seventh ACM symposium on Access control models and technologies*, (New York, NY, USA), pp. 135–140, ACM Press, 2002.

[41] "Shibboleth - Internet2 Middleware." http://shibboleth.internet2.edu/. (Visited December 2008).

[42] D. F. Ferraiolo, J. F. Barkley, and R. Chandramouli, "Comparing Authorization Management Cost for Identity-Based and Role-Based Access Control." http://www.itl.nist.gov/div897/staff/barkley/cost061099.doc, 1999. (Visited July 2008).

[43] D. F. Ferraiolo and R. Kuhn, "Role-Based Access Controls," in *15th NIST-NCSC National Computer Security Conference*, pp. 554–563, Oct. 1992.

[44] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-Based Access Control Models," *Computer*, vol. 29, no. 2, pp. 38–47, 1996.

[45] R. S. Sandhu, D. F. Ferraiolo, and R. Kuhn, "The NIST Model for Role-Based Access Control: Towards a Unified Standard," in *5th ACM Workshop on Role-Based Access Control*, pp. 47–63, Oct. 2000.

[46] N. Yialelis, E. Lupu, and M. Sloman, "Role-based Security for Distributed Object Systems," in *WET-ICE '96: Proceedings of the 5th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'96)*, (Washington, DC, USA), p. 80, IEEE Computer Society, 1996.

[47] J. H. Saltzer and M. D. Schroeder, "The Protection of Information in Computer Systems," in *Proceedings of the IEEE*, pp. 1278–1308, 1975.

[48] D. R. Kuhn, "Mutual Exclusion of Roles as a means of Implementing Separation of Duty in Role-Based Access Control Systems," in *RBAC '97: Proceedings of the Second ACM workshop on Role-Based Access Control*, (New York, NY, USA), pp. 23–30, ACM, 1997.

[49] M. Abadi, M. Burrows, B. Lampson, and G. Plotkin, "A Calculus for Access Control in Distributed systems," *ACM Trans. Program. Lang. Syst.*, vol. 15, no. 4, pp. 706–734, 1993.

[50] E. Barka and R. Sandhu, "Framework for Role-Based Delegation Models," in *ACSAC '00: Proceedings of the 16th Annual Computer Security Applications Conference*, (Washington, DC, USA), p. 168, IEEE Computer Society, 2000.

[51] N. Yialelis, "Domain Based Security for Distributed Object Systems." Ph.D. Dissertation, Department of Computing, Imperial College, London, Oct. 1996.

[52] L. Zhang, G.-J. Ahn, and B.-T. Chu, "A Rule-Based Framework for Role-Based Delegation and Revocation," *ACM Trans. Inf. Syst. Secur.*, vol. 6, no. 3, pp. 404–441, 2003.

[53] E. Lupu and M. Sloman, "Reconciling Role Based Management and Role Based Access Control," in *RBAC '97: Proceedings of the Second ACM Workshop on Role-Based Access Control*, (New York, NY, USA), pp. 135–141, ACM, 1997.

[54] M. Lorch, B. Cowles, R. Baker, L. Gommans, P. Madsen, A. McNab, L. Ramakrishnan, K. Sankar, D. Skow, and M. Thompson, "Conceptual Grid Authorization Framework

and Classification." www.gridforum.org/documents/GFD.38.pdf, 2004. A Global Grid Forum Memo with focus on Grid Security and Authorization.

[55] "ITU-T Rec X.812 (1995) — ISO/IEC 10181-3:1996, Security Frameworks for Open Systems: Access Control Framework," 1995.

[56] D. W. Chadwick, A. Otenko, and E. Bell, "Role-based Access Control with X.509 Attribute Certificates," in *Proceedings of IEEE Internet Computing*, pp. 62–69, Apr. 2003.

[57] C. Neuman, T. Yu, S. Hartman, and K. Raeburn, "The Kerberos Network Authentication Service (V5)." Request for Comments RFC 4120, http://www.ietf.org/rfc/rfc4120.txt, Jul 2005. (Visited December 2008).

[58] R. Alfieri, R. Cecchini, V. Ciaschini, L. dell'Agnello, À. Frohmer, A. Gianoli, K. Lörentey, and F. Spataro, "VOMS, an Authorization System for Virtual Organizations," in *European Across Grids Conference*, pp. 33–40, 2003.

[59] "The Globus Project." http://www.globus.org/. (Visited December 2008).

[60] "LCAS - A Local Centre Authorization Service." http://www.dutchgrid.nl/DataGrid/wp4/lcas/. (Visited December 2008).

[61] "LCMAPS - A Local Credential Mapping Service." http://www.dutchgrid.nl/DataGrid/wp4/lcmaps/. (Visited December 2008).

[62] R. Alfieri, R. Cecchini, V. Ciaschini, L. dell'Agnello, . Frohner, K. Lorentey, and F. Spataro, "From gridmap-file to VOMS: managing authorization in a Grid environment," *Future Generation Computer Systems*, vol. 21, no. 4, pp. 549 – 558, 2005.

[63] "Integrating VOMS and PERMIS for Superior Secure Grid Management (VPMan)." http://sec.cs.kent.ac.uk/vpman/. (Visited December 2008).

[64] L. Pearlman, V. Welch, I. Foster, C. Kesselman, and S. Tuecke, "A Community Authorization Service for Group Collaboration," in *POLICY '02: Proceedings of the 3rd International Workshop on Policies for Distributed Systems and Networks (POLICY'02)*, (Washington, DC, USA), p. 50, IEEE Computer Society, 2002.

[65] J. Novotny, S. Tuecke, and V. Welch, "An Online Credential Repository for the Grid: MyProxy," *hpdc*, vol. 00, pp. 104–111, 2001.

[66] W3C, "SOAP Version 1.2 Specification." http://www.w3.org/TR/soap12, 2007. (Visited December 2008).

[67] J. Rouault, "Making Sense of the Federation Protocol Landscape." HP Dev Resource Central http://devresource.hp.com/drc/resources/fed_land/federation_landscapeHP.pdf, July 2005. (Visited February 2006).

[68] A. Bhargav-Spantzel, A. Squicciarini, and E. Bertino, "Integrating Federated Digital Identity Management and Trust Negotiation – Issues and Solutions," *Security & Privacy Magazine, IEEE*, vol. 5, no. 2, pp. 55–64, 2007.

[69] Hal Lockhart et.al., "Web Services Federation Language (WS-Federation) v1.1, December 2006." http://www.ibm.com/developerworks/library/specification/ws-fed/. (Visited December 2008).

[70] OASIS, "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard, March 2005." http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf. (Visited December 2008).

[71] T. Wason, S. Cantor, J. Hodges, J. Kemp, and P. Thompson, "Liberty ID-FF Architecture Overview." http://www.projectliberty.org/liberty/content/download/318/2366/file/draft-liberty-idff-arch-overview-1.2-errata-v1.0.pdf. (Visited December 2008).

[72] OASIS Security Services TC, "SAML 2.0 Technical Overview." http://www.oasis-open.org/committees/download.php/11785/sstc-saml-tech-overview-2.0-draft-07.pdf, 13, July 2005. (Visited December 2008).

[73] OASIS, "Bindings for the Oasis Security Assertion Markup Language (SAML) V2.0. OASIS Standard, March 2005." http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf. (Visited December 2008).

[74] OASIS, "Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard, March 2005." http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf. (Visited December 2008).

[75] OASIS, "Web Services Security v1.1." http://www.oasis-open.org/specs/index.php#wssv1.1, 2006. (Visited December 2008).

[76] OASIS Security Services TC, "SAML Executive Overview." http://www.oasis-open.org/committees/download.php/11785/sstc-saml-exec-overview-2.0-draft-06.pdf, 10, Mar. 2005. (Visited December 2008).

[77] "Liberty Alliance Project-Official Web Site." http://www.projectliberty.org/. (Visited December 2008).

[78] A. Tulshibagwale, "Federation Gateway Bridges Identity Standards." Network World http://www.networkworld.com/news/tech/2005/082905techupdate.html, 29, Aug. 2005. (Visited February 2006).

[79] OASIS, "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V1.1. OASIS Standard, September 2003." http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf. (Visited December 2008).

[80] OASIS Security Services TC, "The Official SAML FAQ." http://www.oasis-open.org/committees/security/faq.php, 20, Mar. 2005. (Visited December 2008).

[81] "BRIDGES: Biomedical Research Informatics Delivered by Grid Enabled Services." www.brc.dcs.gla.ac.uk/projects/bridges. Visited November 2008.

[82] J. Novotny, S. Tuecke, and V. Welch, "An Online Credential Repository for the Grid: Myproxy," in *Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10), IEEE*, pp. 104–111, 2001.

[83] M. Norman, "A case for Shibboleth and Grid Security: are we paranoid about identity?." Proceedings of the UK E-Science All Hands Meeting, Nottingham, UK, 2006.

[84] "eXtensible Access Control Markup Language (XACML) TC v2.0." http://www.oasis-open.org/specs/index.php#xacmlv2.0, 2005. (Visited December 2008).

[85] "Organisation for the Advancement of Structured Information Standards (OASIS)." http://www.oasis-open.org/. (Visited August 2008).

[86] S. Chapman, A. Dunlop, P. Henderson, and S. Newhouse, "OMII Grid Security Technology Overview." Open Middleware Infrastructure Institute, University of Southampton, 5, Apr. 2005.

[87] D. Power, M. Slaymaker, E. Politou, and A. Simpson, "On XACML, Role-based Access Control and Health Grids," in *Proceedings of the 2005 UK e-Science All Hands Meeting*, Sept. 2005.

[88] W3C, "Web Services Architecture." http://www.w3.org/TR/ws-arch/, 2004. (Visited December 2008).

[89] R. Srinivasan, "RPC: Remote Procedure Call Protocol Specification Version 2." http://www.ietf.org/rfc/rfc1831.txt, 1995. (Visited December 2008).

[90] OMG, "Common Object Request Broker Architecture (CORBA/IIOP)." http://www.omg.org/spec/CORBA/3.1, 2008. (Visited December 2008).

[91] Microsoft Corporation, "Distributed Component Object Model (DCOM) Technical Overview." http://msdn.microsoft.com/en-us/library/ms809340.aspx, 1996. (Visited December 2008).

[92] W3C, "XML Encryption Syntax and Processing." http://www.w3.org/TR/xmlenc-core/, 2002. (Visited August 2008).

[93] W3C, "XML Signature Syntax and Processing (Second Edition)." http://www.w3.org/TR/xmldsig-core/, 2008. (Visited August 2008).

[94] OASIS, "WS-SecureConversation 1.3, March 2007." (Visited June 2008).

[95] W3C, "Web Services Policy 1.5 – Framework." (Visited November 2008).

[96] OASIS, "WS-Trust 1.3, March 2007." http://docs.oasis-open.org/ws-sx/ws-trust/v1.3/ws-trust.html. (Visited June 2008).

[97] "Web Services Interoperability Organization." http://www.ws-i.org/. (Visited December 2008).

[98] OASIS, "Web Services Security: SOAP Message Security 1.1 (WS-Security 2004)." http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-errata-os-SOAPMessageSecurity.pdf, 2006. (Visited December 2008).

[99] J. Sabater and C. Sierra, "Review on Computational Trust and Reputation Models," *Artificial Intelligence Review*, vol. 24, no. 1, pp. 33–60, 2005.

[100] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized Trust Management," in *Proceedings 1996 IEEE Symposium on Security and Privacy*, pp. 164–173, May 1996.

[101] M. Blaze, J. Feigenbaum, and A. D. Keromytis, "The Role of Trust Management in Distributed Systems Security," in *Secure Internet Programming*, pp. 185–210, 1999.

[102] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis, "The Keynote Trust-Management System Version 2, RFC 2704." http://www.ietf.org/rfc/rfc2704.txt, Sept. 1999. (Visited August 2008).

[103] M. Blaze, J. Feigenbaum, and A. D. Keromytis, "KeyNote: Trust Management for

Public-Key Infrastructures (Position Paper)," in *Lecture Notes in Computer Science*, pp. 59–63, 1999.

[104] D. Clarke, J.-E. Elien, C. Ellison, M. Fredette, A. Morcos, and R. L. Rivest, "Certificate Chain Discovery in SPKI/SDSI," *J. Comput. Secur.*, vol. 9, no. 4, pp. 285–322, 2001.

[105] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen, "RFC 2693: SPKI Certificate Theory." http://www.ietf.org/rfc/rfc2693.txt, Sept. 1999. (Visited August 2008).

[106] R. L. Rivest and B. Lampson, "SDSI A Simple Distributed Security Infrastructure." http://theory.lcs.mit.edu/ rivest/sdsi10.ps, Aug. 1996. (Visited August 2008).

[107] N. Li, J. Mitchell, and W. Winsborough, "Design of a Role-based Trust-management Framework," in *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, 2002.

[108] N. Li, W. H. Winsborough, and J. C. Mitchell, "Distributed Credential Chain Discovery in Trust Management," *Journal of Computer Security*, vol. 11, pp. 35–86, Feb. 2003.

[109] S. Ceri, G. Gottlob, and L. Tanca, "What you always wanted to know about Datalog (and never dared to ask)," *IEEE Transactions on Knowledge and Data Engineering*, vol. 1, pp. 146–166, Mar. 1989.

[110] E. Bertino, E. Ferrari, and A. Squicciarini, "Trust Negotiations: Concepts, Systems, and Languages," *Computing in Science and Engineering*, vol. 06, no. 4, pp. 27–34, 2004.

[111] W. Winsborough and J. Jacobs, "Automated Trust Negotiation Technology with Attribute-based Access Control," in *Proceedings of DARPA Information Survivability Conference and Exposition, 2003*, vol. 02, pp. 60–62, 22-24, Apr. 2003.

[112] W. H. Winsborough, K. E. Seamons, and V. E. Jones, "Automated Trust Negotiation," *DARPA Information Survivability Conference and Exposition (DISCEX)*, vol. 01, p. 0088, 2000.

[113] W. Winsborough and L. Ninghui, "Safety in Automated Trust Negotiation," in *Proceedings of IEEE Symposium on Security and Privacy, 2004*, pp. 147–160, 2004.

[114] E. Bertino, E. Ferrari, and A. C. Squicciarini, "Trust-X: A Peer-to-Peer Framework for Trust Establishment," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 827–842, 2004.

[115] V. Bharadwaj and J. Baras, "Towards Automated Negotiation of Access Control Policies," in *Proceedings of the Fourth International Workshop on Policies for Distributed Systems and Networks (Policy 2003)*, IEEE Computer Society Press, 2003.

[116] J. Li, J. Huai, J. Xu, Y. Zhu, and W. Xue, "TOWER: Practical Trust Negotiation Framework for Grids," *2nd IEEE International Conference on e-Science and Grid Computing*, Dec. 2006.

[117] S. Ye, F. Makedon, and J. Ford, "Collaborative Automated Trust Negotiation in Peer-to-Peer Systems," in *P2P '04: Proceedings of the Fourth International Conference on Peer-to-Peer Computing (P2P'04)*, (Washington, DC, USA), pp. 108–115, IEEE Computer Society, 2004.

[118] K. Seamons, M. Winslett, and T. Yu, "Limiting the Disclosure of Access Control Policies during Automated Trust Negotiation," in *Proc. Network and Distributed System Security Symposium, San Diego, CA*, Apr. 2001.

[119] W. H. Winsborough and N. Li, "Towards Practical Automated Trust Negotiation," in *Proceedings of the Third International Workshop on Policies for Distributed Systems and Networks (Policy 2002)*, pp. 92–103, IEEE Computer Society Press, June 2002.

[120] T. Yu, M. Winslett, and K. E. Seamons, "Interoperable Strategies in Automated Trust Negotiation," in *CCS '01: Proceedings of the 8th ACM Conference on Computer and Communications Security*, (New York, NY, USA), pp. 146–155, ACM Press, 2001.

[121] T. Yu, M. Winslett, and K. E. Seamons, "Supporting Structured Credentials and Sensitive Policies through Interoperable Strategies for Automated Trust Negotiation," *ACM Trans. Inf. Syst. Secur.*, vol. 6, no. 1, pp. 1–42, 2003.

[122] A. J. Lee, M. Winslett, J. Basney, and V. Welch, "Traust: A Trust Negotiation-based Authorization Service for Open Systems," in *SACMAT '06: Proceedings of the eleventh ACM symposium on Access control models and technologies*, (New York, NY, USA), pp. 39–48, ACM, 2006.

[123] "Semantic Grid Vision." http://www.semanticgrid.org/vision.htm. (Visited September 2006).

[124] H. Stuckenschmidt and F. van Harmelen, *Information Sharing on the Semantic Web*, ch. Ontology-based information sharing, pp. 25–44. Springer-Verlag Berlin Heidelberg, July 2004.

[125] A. Doan, J. Madhavan, P. Domingos, and A. Halevy, "Learning to Map between Ontologies on the Semantic Web," in *WWW '02: Proceedings of the Eleventh International Conference on World Wide Web*, pp. 662–673, ACM Press, 7-11, May 2002.

[126] R. Sinnott, A. Stell, and O. Ajayi, "Development of Grid Frameworks for Clinical Trials and Epidemiological Studies," in *Challenges and Opportunities of HealthGrids - Proceedings of Healthgrid '06*, vol. 120, June 2006.

[127] "Virtual Organisations for Trials and Epidemiological Studies (VOTES)." http://www.nesc.ac.uk/hub/projects/votes. (Visited December 2008).

[128] I. Kotsiopoulos, J. Keane, M. Turner, P. Layzell, and F. Zhu, "IBHIS: Integration Broker for Heterogeneous Information Sources," in *COMPSAC '03: Proceedings of the 27th Annual International Conference on Computer Software and Applications*, (Washington, DC, USA), p. 378, IEEE Computer Society, 2003.

[129] F. Benchikha, M. Boufaida, and L. Seinturier, "Integration of the Viewpoint Mechanism in Federated Databases," in *SAC '01: Proceedings of the 2001 ACM Symposium on Applied Computing*, (New York, NY, USA), pp. 280–284, ACM Press, 2001.

[130] D. Heimbigner and D. McLeod, "A Federated Architecture for Information Management," *ACM Trans. Inf. Syst.*, vol. 3, no. 3, pp. 253–278, 1985.

[131] D. Kalra, P. Singleton, D. Ingram, J. Milan, J. MacKay, D. Detmer, and A. Rector, "Security and Confidentiality Approach for the Clinical e-Science Framework," 2003.

[132] A. Rector, A. Taweel, J. Rogers, D. Ingram, D. Kalra, R. Gaizauskas, M. Hepple, J. Milan, R. Powers, D. Scott, and P. Singleton, "Joining up Health and Bioinformatics: E-Science meets e-Health." Proceedings of the third UK E-Science All Hands Meeting, Nottingham, UK, 2004.

[133] "Artemis Project." http://www.srdc.metu.edu.tr/webpage/projects/artemis. (Visited December 2008).

[134] A. Dogac, G. B. Laleci, S. Kirbas, Y. Kabak, S. S. Sinir, A. Yildiz, and Y. Gurcan, "Artemis: Deploying Semantically Enriched Web Services in the Healthcare Domain," *Inf. Syst.*, vol. 31, no. 4, pp. 321–339, 2006.

[135] "Health Level 7 (HL7)." http://www.hl7.org/. (Visited March 2008).

[136] "CEN TC/251 (European Standardization of Health Informatics) ENV 13606, Electronic Health Record Communication." http://www.centc251.org/. (Visited August 2008).

[137] "ISO TC215, International Organization for Standardization, Health Informatics Technical Committee." http://www.iso.org/iso/standards_development/technical_committees/list_of_iso_technical_committees/iso_technical_committee.htm?commid=54960. (Visited December 2008).

[138] S. Castano, A. Ferrara, S. Montanelli, and G. Racca, "Semantic Information Interoperability in Open Networked Systems," in *Proc. of the Int. Conference on Semantics of a Networked World (ICSNW), in cooperation with ACM SIGMOD 2004*, (Paris, France), pp. 215–230, June 2004.

[139] "The ARTEMIS Architecture and Process." http://islab.dico.unimi.it/artemis/architecture.php. Visited December 2008.

[140] J. Ainsworth, R. Harper, I. Juma, and I. Buchan, "Design and Implementation of Security in a Data Collection System for Epidemiology," in *Challenges and Opportunities of HealthGrids - Proceedings of Healthgrid '06*, vol. 120, pp. 348–357, 2006.

[141] J. Ainsworth, R. Harper, I. Juma, and I. Buchan, "Psygrid: Applying e-Science to Epidemiology," in *CBMS '06: Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems*, (Washington, DC, USA), pp. 727–732, IEEE Computer Society, 2006.

[142] "OGSA-DQP: Open Grid Service Architecture - Distributed Query Processing." http://www.ogsadai.org.uk/about/ogsa-dqp/. (Visited December 2008).

[143] "The Cancer Biomedical Informatics Grid." http://cabig.cancer.gov. (Visited August 2008).

[144] "Web Services Resource Framework (WSRF) v1.2." http://www.oasis-open.org/specs/index.php#wsrfv1.2. (Visited August 2008).

[145] S. Langella, S. Oster, S. Hastings, F. Siebenlist, J. Phillips, D. Ervin, J. Permar, T. Kurc, and J. Saltz, "The Cancer Biomedical Informatics Grid (cabig) Security Infrastructure," in *Proceedings of the 2007 AMIA Annual Symposium*, 2007.

[146] "Ncicb Common Security Module (CSM)." http://ncicb.nci.nih.gov/infrastructure/cacore_overview/csm. (Visited August 2008).

[147] "UK BioBank." http://www.ukbiobank.ac.uk/. (Visited August 2008).

[148] P. Elliott and T. C. Peakman, "The UK Biobank Sample Handling and Storage Protocol for the Collection, Processing and Archiving of Human Blood and Urine," *International Journal of Epidemiology*, vol. 37, no. 42, pp. 234–244, 2008.

[149] "UK BioBank: Protocol for a Large-Scale Prospective Epidemiological Resource." http://www.ukbiobank.ac.uk/docs/UKBProtocolfinal.pdf, Mar. 2007. (Visited August 2008).

[150] "Scottish Health Information Platform (SHIP)." http://www.scot-hip.ac.uk. Yet to be functional as at December 2008.

[151] O. Ajayi, R. Sinnott, and A. Stell, "Trust Realisation in Collaborative Clinical Trials Systems," in *HealthCare Computing Conference HC2007, Harrogate, England*, Mar. 2007.

[152] "Eduperson Specification." http://www.educause.edu/eduperson/. (Visited August 2008).

[153] R. Sinnott, J. Watt, J. Jiang, A. Stell, O. Ajayi, and J. Koetsier, "Single Sign-on and Authorization for Dynamic Virtual Organizations," in *7th IFIP Conference on Virtual Enterprises, PRO-VE 2006, Helsinki, Finland, September 2006*, Springer, 2006.

[154] "SCI-Store - Scottish Care Information." http://www.sci.scot.nhs.uk/products/store/store_main.htm. (Visited December 2008).

[155] "General Practice Administration System for Scotland (GPASS)." http://www.show.scot.nhs.uk/gpass. (Visited December 2008).

[156] "Picture Archiving and Communications System (PACS)." http://www.connectingforhealth.nhs.uk/systemsandservices/pacs. (Visited August 2008).

[157] "Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data." http://ec.europa.eu/justice_home/fsj/privacy/law/index_en.htm. (Visited December 2008).

[158] "Health Insurance Portability and Accountability Act." Standards for Privacy of Individually Identifiable Health Information - Rules and Regulations, 2000.

[159] J. Basney, W. Nejdl, D. Olmedilla, V. Welch, and M. Winslett, "Negotiating Trust on the Grid," in *2nd Workshop on Semantics in Peer-to-peer and Grid Computing*, 2004.

[160] "Honorary Consultant Contract (England) 2003 (amended April 2007), NHS Employers: Key Documents." http://www.nhsemployers.org/restricted/downloads/download.asp?ref=471&hash=0bb1c57dd0efdfec1d7956b2b292ed3e&itemplate=_pay_conditions_3col_consult_pay-conditions-348, 2003. (Visited August 2008).

[161] "Connecting for Health (CfP) / National Programme for Information Technology (NPfIT)." http://www.connectingforhealth.nhs.uk/. (Visited August 2008).

[162] W3C, "Web Services Addressing 1.0 - Core, May 2006." (Visited August 2008).

[163] OASIS, "XACML 2.0 Core: eXtensible Access Control Markup Language (XACML) Version 2.0." http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf. (Visited December 2008).

[164] Y. Zhang, X. Li, J. Huai, and Y. Liu, "Access Control in Peer-to-Peer Collaborative Systems," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'05*, IEEE, 2005.

[165] L. Zhang, G.-J. Ahn, and B.-T. Chu, "A Rule-Based Framework for Role Based Delegation," in *SACMAT '01: Proceedings of the sixth ACM Symposium on Access Control Models and Technologies*, (New York, NY, USA), pp. 153–162, ACM, 2001.

[166] "A Lightweight Deductive Database System Written in Lua." http://www.ccs.neu.edu/home/ramsdell/tools/datalog/. (Visited November 2008).

[167] O. Ajayi, R. Sinnott, and A. Stell, "Towards Decentralised Security Policies for e-Health Collaborations," in *Proceedings of 2nd International Conference on Emerging Security Information, Systems and Technologies, (SECURWARE), Cap Esterel, France*, IEEE Computer Society, Aug. 2008.

[168] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication." Request for Comments RFC 2104, http://www.ietf.org/rfc/rfc2104.txt, Feb 1997. (Visited April 2008).

[169] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing; RFC3561." http://www.ietf.org/rfc/rfc3561.txt, 2003. (Visited August 2008).

[170] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing," *Second IEEE Workshop on Mobile Computer Systems and Applications (WMCSA)*, vol. 0, p. 90, 1999.

[171] C. Hedrick, "Routing Information Protocol." http://www.ietf.org/rfc/rfc1058.txt, 1988. (Visited August 2008).

[172] A. Schmid, "Solution for the counting to infinity problem of distance vector routing," Tech. Rep. 7–98, Universität Koblenz-Landau, Institut für Informatik, Rheinau 1, D-56075 Koblenz, 1998.

[173] J. Moy, "Ospf Version 2." http://www.ietf.org/rfc/rfc2328.txt, 1998. (Visited August 2008).

[174] P. Ferguson and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing." http://www.ietf.org/rfc/rfc2827.txt, 2000. (Visited August 2008).

[175] F. Baker and P. Savola, "Ingress Filtering for Multihomed Networks." http://www.ietf.org/rfc/rfc3704.txt, 2004. (Visited December 2008).

[176] J. Callas, L. Donnerhacke, H. Finney, D. Shaw, and R. Thayer, "OpenPGP Message Format." Request for Comments RFC 4880, http://www.ietf.org/rfc/rfc4880.txt, Nov 2007. (Visited December 2008).

[177] E. W. Lader, C. P. Cannon, E. M. Ohman, L. K. Newby, D. P. Sulmasy, R. J. Barst, J. M. Fair, M. Flather, J. E. Freedman, R. L. Frye, M. M. Hand, R. L. Jesse, F. V. de Werf, and F. Costa, "The Clinician as Investigator: Participating in Clinical Trials in the Practice Setting," *American Heart Association*, vol. 109, pp. 2672–2679, June 2004.

[178] M. D. Green, D. M. Freedman, and L. Gordis, "Reference Guide on Epidemiology." Reference Manual on Scientific Evidence, 2nd ed., Washington, DC: Federal Judicial Center, 2000.

[179] A. Monroe, "Clinical trials explained." Published by AIDS Community Research Initiative of America (ACRIA), 2000.

[180] A. G. Oliveira and N. C. Salgado, "Design Aspects of a Distributed Clinical Trials Information System," *Clinical Trials: Data Management And Study Conduct*, vol. 3, pp. 385–396, 2006.

[181] "US Food and Drug Administration. Good Clinical Practice in FDA-Regulated Clinical Trials." http://www.fda.gov/oc/gcp/default.htm. (Visited August 2008).

[182] J. Shepherd, S. Cobbe, I. Ford, and et al., "Prevention of Coronary Heart Disease with Pravastatin in Men with Hypercholesterolemia," *New England Journal of Medicine*, vol. 333, pp. 1301–1307, 3, Nov. 1995.

[183] The West of Scotland Coronary Prevention Study Group, "A Coronary Primary Prevention Study of Scottish Men Aged 45-64 Years: Trial Design," *J Clin Epidemiol*, vol. 45, no. 8, pp. 849–860, 1992.

[184] The WOSCOPS Study Group, "Screening Experience and Baseline Characteristics in the West of Scotland Coronary Prevention Study," *The American Journal of Cardiology*, vol. 76, pp. 485–491, 1, Sept. 2005.

[185] J. Shepherd, G. J. Blauw, M. B. Murphy, E. L. E. M. Bollen, and et al., "Pravastatin in Elderly Individuals at Risk of Vascular Disease (PROSPER): A Randomised Controlled Trial," *LANCET*, vol. 360, pp. 1623–1630, 23, Nov. 2002.

[186] J. Shepherd, G. J. Blauw, M. B. Murphy, S. M. Cobbe, and et al., "The Design of a Prospective Study of Pravastatin in the Elderly at Risk (PROSPER)," *American Journal of Cardiology*, vol. 84, pp. 1192–1197, 15, Nov. 1999.

[187] "General Register Office for Scotland." http://www.gro-scotland.gov.uk. (Visited August 2008).

[188] "Scottish Morbidity Records (SMR)." http://www.show.scot.nhs.uk/indicators/SMR/Main.htm. (Visited September 2006).

[189] "NHS Data Dictionary." http://www.isdscotland.org. (Visited August 2008).

[190] "SNOMED-CT." http://www.ihtsdo.org/snomed-ct/. (Visited August 2008).

[191] "OpenEHR." http://www.openehr.org/. (Visited August 2008).

[192] "International Statistical Classification of Disease and Related Health Problems (icd-10)." http://www.connectingforhealth.nhs.uk/systemsandservices/data/clinicalcoding/codingstandards/icd-10. (Visited August 2008).

[193] S. M. Eldridge, D. Ashby, and G. S. Feder, "Informed Patient Consent to Participation in Cluster Randomized Trials: An Empirical Exploration of Trials in Primary Care," *Clinical Trials: Workshop Article*, vol. 2, pp. 91–98, 2005.

[194] J. Sugarman, P. W. Lavori, M. Boeger, C. Cain, R. Edson, V. Morrison, and S. S. Yeh, "Evaluating the Quality of Informed Consent," *Clinical Trials: Ethics*, vol. 2, pp. 34–41, 2005.

[195] R. Sinnott, O. Ajayi, A. Stell, and A. Young, "Towards a Virtual Anonymisation Grid for Unified Access to Remote Clinical Data," in *6th International HealthGrid Conference, Chicago, USA*, June 2008.

[196] "Open Grid Services Architecture - Data Access and Integration (OGSA-DAI)." www.ogsadai.org.uk. (Visited August 2008).

[197] "Apache Xindice." http://xml.apache.org/xindice/. Visited December 2008.

[198] "Shibboleth Architecture Protocols and Profiles." http://shibboleth.internet2.edu/docs/internet2-mace-shibboleth-arch-protocols-latest.pdf. (Visited December 2008).

[199] "Organization for the Advancement of Structured Information Standards (OASIS)." Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, 15 March 2005.

[200] O. Ajayi, R. Sinnott, and A. Stell, "Blind Data Aggregation from Distributed, Protected Sources: The Future Model for Security-oriented Collaborations." Workshop of the UK E-Science All Hands Meeting, Edinburgh, UK, 2008.

[201] "Aaa, Middleware and DRM." http://www.nesc.ac.uk/documents/OSI/aaa.pdf. Visited December 2008.

[202] "The Network Simulator - ns-2." http://www.isi.edu/nsnam/ns/. Visited December 2008.

[203] S. Keshav, "REAL: A Network Simulator." Department of Computing Science, UC Berkeley, 1988, http://www.cs.cornell.edu/skeshav/papers/real.ps. Visited December 2008.

[204] L. Rizzo, "Dummynet: A simple approach to the evaluation of network protocols," *ACM Computer Communication Review*, vol. 27, pp. 31 – 41, January 1997.

[205] "RFC 4271, A Border Gateway Protocol 4 (BGP-4)." http://tools.ietf.org/html/rfc4271, Jan. 2006. (Visited December 2008).

[206] V. Paxson and M. Allman, "Computing TCP's Retransmission Timer." Request for Comments RFC 2988, Nov 2000.

[207] H. Zimmermann, "Osi Reference Model-The ISO Model of Architecture for Open Systems Interconnection," *IEEE Transactions on Communications*, April 1980.

[208] O. Ajayi, R. Sinnott, and A. Stell, "Dynamic Trust Negotiation for Flexible e-Health Collaborations," in *Proceedings of 15th Mardi Gras Conference, Baton Rouge, USA*, ACM Digital Library, Feb. 2008.

[209] J. Watt, R. Sinnott, O. Ajayi, J. Jiang, and J. Koetsier, "A Shibboleth-Protected Privilege Management Infrastructure for e-Science Education," in *Proceedings of 6th International Symposium on Cluster Computing and the Grid, CCGrid2006, Singapore*, May 2006.

[210] R. Sinnott, J. Watt, J. Jiang, and O. Ajayi, "Shibbolth-based Access to and Usage of Grid Resources," in *Proceedings of IEEE International Conference on Grid Computing, Barcelona, Spain*, Sept. 2006.

[211] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration." http://www.globus.org/research/papers/ogsa.pdf, Jan. 2002. (Visited September 2005).

[212] J. Kirk, "Identity Federation: Is it time to move now?." Network World http://www.networkworld.com/news/2005/091505-identity-federation.html, 9, Sept. 2005. (Visited February 2006).

[213] S. D. Vimercati and P. Samarati, "An Authorization Model for Federated Systems," in *ESORICS '96: Proceedings of the 4th European Symposium on Research in Computer Security*, (London, UK), pp. 99–117, Springer-Verlag, 1996.

[214] D. Chadwick, "Delegation Issuing Service," in *NIST 4th Annual PKI Workshop*, (Gaithersberg, USA), pp. 62–73, April 2005.

[215] K. Taylor and J. Murty, "Implementing Role Based Access Control for Federated Information Systems on the Web," in *CRPITS '03: Proceedings of the Australasian Information Security Workshop Conference on ACSW Frontiers 2003*, (Darlinghurst, Australia, Australia), pp. 87–95, Australian Computer Society, Inc., 2003.

[216] M. Gaedke, J. Meinecke, and M. Nussbaumer, "A Modeling Approach to Federated Identity and Access Management," in *WWW '05: Special interest tracks and posters of the 14th International Conference on World Wide Web*, (New York, NY, USA), pp. 1156–1157, ACM Press, 2005.

[217] M. Hadzic and E. Chang, "Role of the Ontologies in the Context of Grid Computing and Application for the Human Disease Studies.," in *ICSNW*, pp. 316–318, 2004.

[218] C. Gates and J. Slonim, "Owner-Controlled Information," in *NSPW '03: Proceedings of the 2003 Workshop on New Security Paradigms*, (New York, NY, USA), pp. 103–111, ACM Press, 2003.

[219] P. Wendel, M. Ghanem, and Y. Guo, eds., *Scalable Clustering on the Data Grid*, UK e-Science All Hands Meeting, 2005.

[220] A. M. Antonopoulos, "Identity: More than just security." Network World, 30, Aug. 2005. (Visited September 2005).

[221] M. Bouzeghoub, C. A. Goble, V. Kashyap, and S. Spaccapietra, eds., *Semantics for Grid Databases, First International IFIP Conference on Semantics of a Networked World: ICSNW 2004, Paris, France, June 17-19, 2004. Revised Selected Papers*, vol. 3226 of *Lecture Notes in Computer Science*, Springer, 2004.

[222] L. Kerschberg, M. Chowdhury, A. Damiano, H. Jeong, S. Mitchell, J. Si, and S. Smith, "Knowledge Sifter: Agent-Based Ontology-Driven Search over Heterogeneous Databases using Semantic Web Services.," in *ICSNW*, pp. 278–295, 2004.

[223] G. Saunders, M. Hitchens, and V. Varadharajan, "Role-based Access Control and the Access Control Matrix," *SIGOPS Oper. Syst. Rev.*, vol. 35, no. 4, pp. 6–20, 2001.

[224] J. Moffett and M. Sloman, "Delegation of Authority," in *Integrated Network Management II, I. Krishnan & W. Zimmer (eds)*, pp. 595–606, Elsevier Science Publishers B.V. North-Holland, Apr. 1990.

[225] B. N. Chun and A. Bavier, "Decentralized Trust Management and Accountability in Federated Systems," in *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04)*, Jan. 2004.

[226] H. Li and M. Singhal, "A Secure Routing Protocol for Wireless Ad Hoc Networks," in *Proceedings of the 39th Hawaii International Conference on System Science*, IEEE, 2006.

[227] G. Wei, X. Zhongwei, and L. Zhitang, "Dynamic Trust Evaluation Based Routing Model for Ad Hoc Networks," *Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing, 2005*, vol. 2, pp. 727–730, 2005.

[228] J. Li, J. Jannotti, D. De Couto, D. Karger, and R. Morris, "A Scalable Location

Service for Geographic Ad-Hoc Routing," in *Proceedings of the 6th ACM International Conference on Mobile Computing and Networking (MobiCom '00)*, pp. 120–130, aug 2000.

[229] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A scalable Peer-To-Peer Lookup Service for Internet Applications," in *Proceedings of the 2001 ACM SIGCOMM Conference*, pp. 149–160, 2001.

[230] Z. Liu, A. W. Joy, and R. A. Thompson, "A Dynamic Trust Model for Mobile Ad Hoc Networks," in *FTDCS '04: Proceedings of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'04)*, (Washington, DC, USA), pp. 80–85, IEEE Computer Society, 2004.

[231] N. Yialelis and M. Sloman, "A Security Framework Supporting Domain Based Access Control in Distributed Systems," in *IEEE ISOC Symposium on Network and Distributed Systems Security'96, San Diego*, pp. 26–39, IEEE, 22-23, Feb. 1996.

[232] F. Sadri, F. Toni, and P. Torroni, "Logic Agents, Dialogues and Negotiation: An Abductive Approach," in *Proceedings of the Symposium on Information Agents for ECommerce, AISB'01, March 2001*, 2001.

[233] M. Wooldridge and S. Parsons, "Languages for Negotiation," in *Proceedings of the Fourteenth European Conference on Artificial Intelligence (ECAI-2000)* (W. Horn, ed.), John Wiley & Sons, 2000.

[234] M. Wooldridge, "Semantic Issues in the Verification of Agent Communication Languages," *Autonomous Agents and Multi-Agent Systems*, vol. 3, no. 1, pp. 9–31, 2000.

[235] E. Andonoff and L. Bouzguenda, "Agent-Based Negotiation between Partners in Loose Inter-Organizational Workflow," *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, vol. 0, pp. 619–625, 2005.

[236] N. Vulkan and N. R. Jennings, "Efficient Mechanisms for the Supply of Services in Multi-Agent Environments," in *ICE '98: Proceedings of the First International Conference on Information and Computation Economies*, (New York, NY, USA), pp. 1–10, ACM Press, 1998.

[237] C. Bartolini, C. Priest, and N. R. Jennings, "A Software Framework for Automated Negotiation," in *Proceedings of (SELMAS'04): Research Issues and Practical Applications*, pp. 213–235, LNCS 3390, Springer-Verlag, 2005.

[238] N. Nagaratnam, P. Janson, J. Dayka, A. Nadalin, F. Siebenlist, V. Welch, I. Foster,

and S. Tuecke, "The security Architecture for Open Grid Services." Open Grid Service Architecture Security Working Group, Global Grid Forum, 2002, 2002.

[239] R. Sinnott, J. Watt, J. Koetsier, D. Chadwick, O. Otenko, and T. Nguyen, "Supporting Decentralized, Security focused Dynamic Virtual Organizations across the Grid," in *Proceedings of 2nd IEEE International Conference on e-Science and Grid Computing, Amsterdam, December 2006*, 2006.

[240] A. Stell, R. Sinnott, and O. Ajayi, "Secure, Reliable and Dynamic Access to Distributed Clinical Data," in *Proceedings of Life Science Grid Conference, Yokohama, Japan*, Oct. 2006.

[241] Y. Elley, A. Anderson, S. Hanna, S. Mullan, R. Perlman, and S. Proctor, "Building Certification Paths: Forward vs. Reverse," in *Proceedings of Network and Distributed System Security Symposium Conference*, 2001.

[242] H. Huang and S. F. Wu, "An Approach to Certificate Path Discovery in Mobile Ad Hoc Networks," in *Proceedings of the 1st ACM workshop on Security of Ad hoc and Sensor Networks SASN '03*, pp. 41–52, ACM Press, 2003.

[243] S. Terzis, W. Wagealla, C. English, and P. Nixon, "Trust Lifecycle Management in a Global Computing Environment," in *Global Computing (GC 2004)* (C. Priami and P. Quaglia, eds.), pp. 291–313, Spinger, 2005.

[244] W. Nejdl, D. Olmedilla, and M. Winslett, "PeerTrust: Automated Trust Negotiation for Peers on the Semantic Web," in *Proc. of the Workshop on Secure Data Management in a Connected World (SDM'04)*, August/September 2004.

[245] D. O'Callaghan and B. Coghlan, "On-demand Trust Evaluation," in *Proc. Grid'2006, Barcelona, Spain*, Sept. 2006.

[246] H. Yamaki, M. Fujii, K. Nakatsuka, and T. Ishida, "A Dynamic Programming Approach to Automated Trust Negotiation for Multiagent Systems," *Rational, Robust, and Secure Negotiation Mechanisms in Multi-Agent Systems (RRS'05)*, pp. 55–66, 2005.

[247] D. Yao, M. Shin, R. Tamassia, and W. H. Winsborough, "Visualization of Automated Trust Negotiation," in *Proceedings of the IEEE Workshops on Visualization for Computer Security VIZSEC'05*, IEEE Computer Society, Oct. 2005.

[248] I. Ford, G. J. Blauw, M. B. Murphy, and et al., "A Prospective Study of Pravastatin in the Elderly at Risk (PROSPER): Screening Experience and Baseline Characteristics," *Current Controlled Trials in Cardiovascular Medicine*, vol. 3, 20, May 2002.

[249] F. Clemens, D. Elbourne, J. Darbyshire, S. Pocock, and DAMOCLES Group, "Data Monitoring in Randomized Controlled Trials: Surveys of Recent Practice and Policies," *Clinical Trials: Policy Regulation And Law*, vol. 2, pp. 22–33, 2005.

[250] S. Claub and M. Köhntopp, "Identity Management and its support of Multilateral Security," *Computer Networks*, vol. 37, pp. 205–219, 2, Oct. 2001.

[251] E. Yuan and J. Tong, "Attributed Based Access Control (ABAC) for Web Services," in *IEEE International Conference on Web Services (ICWS'05)*, pp. 561–569, IEEE Computer Society, 2005.

[252] R. Alonso and D. Barbará, "Negotiating Data Access in Federated Database Systems," in *Proceedings of the Fifth International Conference on Data Engineering*, (Washington, DC, USA), pp. 56–65, IEEE Computer Society, 1989.

[253] J. McQuillan, I. Richer, and E. Rosen, "The New Routing Algorithm for the ARPANET," *IEEE Transactions on Communications*, vol. 28, pp. 711–719, May 1980.

[254] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2001.

[255] OASIS, "WS-SecurityPolicy 1.2, July 2007." http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/ws-securitypolicy-1.2-spec-os.html. (Visited June 2008).