Semantic Security: Specification and Enforcement of Semantic Policies for Security-driven Collaborations

R.O. Sinnott, T.Doherty,
National e-Science Centre, University of Glasgow
N. Gray[1,2], J. Lusted[1]
[1]*Department of Physics and Astronomy, University of Leicester*
[2]*Department of Physics and Astronomy, University of Glasgow*
r.sinnott@nesc.gla.ac.uk

Abstract

Collaborative research can often have demands on finer-grained security that go beyond the authentication-only paradigm as typified by many e-Infrastructure/Grid based solutions. Supporting finer-grained access control is often essential for domains where the specification and subsequent enforcement of authorization policies is needed. The clinical domain is one area in particular where this is so. However it is the case that existing security authorization solutions are fragile, inflexible and difficult to establish and maintain. As a result they often do not meet the needs of real world collaborations where robustness and flexibility of policy specification and enforcement, and ease of maintenance are essential. In this paper we present results of the JISC funded Advanced Grid Authorisation through Semantic Technologies (AGAST) project (www.nesc.ac.uk/hub/projects/agast) and show how semantic-based approaches to security policy specification and enforcement can address many of the limitations with existing security solutions. These are demonstrated into the clinical trials domain through the MRC funded Virtual Organisations for Trials and Epidemiological Studies (VOTES) project (www.nesc.ac.uk/hub/projects/votes) and the epidemiological domain through the JISC funded SeeGEO project (www.nesc.ac.uk/hub/projects/seegeo).

Keywords: authentication, authorization, OWL, RDF, policy specification, policy enforcement

1 Introduction

The vision of e-Science and the Grid in delivering environments for research – where access to and use of distributed and heterogeneous resources is made seamless and transparent to end user researcher – is beset by many challenges. Security is one of the key areas that must be addressed to realise this vision. It is the case that mainstream Grid resource providers such as the UK e-Science National Grid Service (NGS – www.ngs.ac.uk) have primarily adopted a security model based upon a public key infrastructure (PKI). In this model end user researchers are expected to acquire and manage their own X.509 based certificates. The problems with this model include: usability since non-technical communities are often put-off by the complexity of PKI based solutions and taking care of their own certificates, e.g. converting them to formats understood by the Grid middleware; overall security since there is often no security on what the end user is actually allowed to do when they access a resource such as the NGS. Thus they can upload their own codes to do arbitrary simulations for example. In many domains such access and usage paradigms would never be supported. These issues are described in more detail in [1-3].

To address this much focus has been given to security solutions that go beyond the user identification-only, i.e., authentication-based, security models, e.g. role-based access control (RBAC) solutions such as PERMIS (www.permis.org). Authorisation solutions that allow specification and enforcement of finer-grained access and usage policies have been implemented and demonstrated in a variety of domains and shown to interoperate with a variety of middleware [4-6]. However, as we describe in this paper such policy specifications and their subsequent enforcement have numerous limitations. These include the static nature of the policies themselves where possession of a particular role is often sufficient to make an access control decision. This static nature of the policies is not just in their specification, but also includes the assignment of roles to individuals. To address this many approaches are based upon pre-agreed, commonly understood roles (attributes). Two examples of this include federated access control models, e.g. based on Shibboleth (http://shibboleth.internet2.edu) where the eduPerson attributes that are agreed in advance across the UK Federation (www.ukfederation.org.uk) and passed around from Identity Providers (IdP) to Service Providers (SP) to make access control decisions. Alternatively centralised attribute authority models are common, e.g. based upon technologies such as the Virtual Organisation Membership Service (VOMS) [7]. In both of these federated and centralised solutions, pre-agreement and assignment of roles/attributes to individuals is

needed. If a user does not have the right role (or present the right role) then an access control enforcement engine will simply deny access – perhaps even though the role that the user has may actually be the same semantically but the actual string that is presented is different.

Furthermore, it is the case that the rigidity of the policy specification itself can lead to many limitations. Ideally access control mechanisms should be easy to specify and enforce, and ideally allow for inference to be used to decide upon a given access control decision. Policies will need to be refined as more roles and privileges and associations between them grow. Thus whilst RBAC allows for hierarchical reasoning to be supported in making access control decisions, it is not well suited to making access control decisions where different role hierarchies exist.

In this paper we show that semantic technologies allow for far richer policy specification and enforcement to be achieved. In particular we show how semantic reasoners and associated ontologies allow support of far richer access control specifications and decisions. To demonstrate this we focus upon two security-driven domains: the clinical domain represented through a case study which supports a clinical trial as part of the MRC-funded VOTES project, and in a case study showing access control based on access to geospatial and census data conducted as part of the JISC-funded SeeGEO project. The rest of the paper is structured as follows. Section 2 outlines authentication and authorisation offerings in mainstream use today and identifies their various limitations. Section 3 introduces semantic technologies and describes the overall architecture for how semantic-based authorisation can be achieved in a Grid-based environment. Section 4 describes the VOTES and SeeGEO projects and outlines the implementation of the case studies and the associated benefits of adopting a semantic-based security approach. Finally, section 5 draws conclusions of the work as a whole and outlines areas of future work.

2. Background to Grid-based Authentication and Authorisation

There are many authentication and authorisation infrastructures existing today. Username/password challenges responses are perhaps the simplest and most widely adopted authentication solution existing on the internet today. However one of the key tenets of the Grid is in supporting seamless single sign-on access to distributed resources, i.e., without the need for repeated authentication (username/password) challenge response. The primary way that this has been addressed by the Grid community is through PKI-based solutions.

2.1 Authentication, PKIs and Shibboleth

In essence, PKIs validate the identity of a given user requesting access to a given resource, or more precisely they validate the identity of the certificate that has been used. Through trusting the source of authority who signed the public key certificate (in the UK this is a centralised Certification Authority (CA) – www.ngs.ac.uk/ca) that a user presents, access control can be achieved based on knowledge of the user identity. For example, with the Globus toolkit (www.globus.org) solution, gatekeepers are used to ensure that signed requests are valid, i.e., from known collaborators. When this is so, i.e., the Distinguished Name (DN) of the requestor is in a locally stored and managed *gridmap* file, then the user is typically given access to the locally set up account as defined in the *gridmap* file. We note that if a user's private key is compromised then there is no easy way that a given provider can determine whether a user is the correct individual or someone masquerading as that individual. This has obvious drawbacks in terms of security models. Pushing the technologies underlying security infrastructure to the end users is very off-putting for many domains since it often requires users to convert X.509 certificates to formats suitable for Grid usage, typically with recourse to SSL libraries that are not available as default on typical PCs. To address this, the UK and numerous countries internationally are developing federated access control systems based upon the Internet2 Shibboleth technologies. With this model, users wishing to access a remote resource (SP) are redirected to their home institutional authentication system (IdP). Once authenticated with their own username and password at their own institution, signed SAML assertions are sent to the SP which can use the information provided to make an access control decision. As before, if a user's authentication credentials at their IdP have been compromised then an SP is not able to determine this. To establish and ensure the integrity of the federation it is thus essential that all IdPs in a federation take appropriate steps to ensure for example that passwords are of an appropriate strength, or that accounts are immediately revoked once a student or staff member has left the institution they are associated with.

Whilst usability is an essential consideration for any security system, neither X.509-based PKIs nor Shibboleth in themselves define meaningful security policies. Thus whilst a user can be mapped through a *gridmap* file to a local account, there is no mention of what the user is allowed to do once they have gained access to the resource. They could in principle compile arbitrary codes for good or bad reasons.

,Furthermore, for the vast majority of cases Shibboleth-based SPs simply use the authentication assertion from the IdP to allow or deny access. In the case of an eJournal SP say, this might be based upon whether that individual's institution has subscribed to that journal. Finer-grained security models are thus needed which go beyond knowledge that someone is from the University of Glasgow then this is sufficient information to make appropriate access control decisions.

Authorisation standards, infrastructures and technologies extend authentication models to support precisely such finer-grained security control when accessing and using Grid resources. One of the most refined of these and representative of many offerings in this space is the PERMIS RBAC solution.

2.2. Authorisation and PERMIS

Authentication should be augmented with authorisation capabilities, which can be considered as what Grid users are allowed to do on a given Grid end-system. This "*what users are allowed to do*" can be interpreted as the privileges that the users have been allocated on those end-systems. The X.509 standard [8] has standardised the certificates of a privilege management infrastructure (PMI). A PMI can be considered as being related to authorisation in much the same way as a PKI is related to authentication. Consequently, there are many similar concepts in PKIs and PMIs as discussed in detail in [9].

A key concept from PMI are attribute certificates (ACs) which, in much the same manner as public key certificates in PKI, maintain a strong binding between a user's name and one or more privilege attributes. The Privilege and Role Management Infrastructure Standards Validation (PERMIS) project was an EC project that built an RBAC authorisation infrastructure to realise a scalable X.509 AC-based PMI. The PERMIS RBAC system uses XML based policies defining rules, specifying which access control decisions are to be made for given VO resources. These rules include definitions of: subjects that can be assigned roles; Sources of Authority (SOA) which are typically local managers trusted to assign roles to subjects; roles and their hierarchical relationships; what roles can be assigned to which subjects by which SOAs; target resources, and the actions that can be applied to them; which roles are allowed to perform which actions on which targets; and the conditions under which access can be granted to roles.

Roles are assigned to subjects by issuing them with X.509 ACs which, with PERMIS, are typically stored in a local LDAP server along with the SOA root certificate. A typical scenario in using PERMIS is where a remote user creates a proxy-credential from their X.509 certificate and, through a remote client, tries to invoke a local PERMIS-protected service, i.e., a service which has an associated policy enforcement point (PEP) which must be satisfied by a policy decision point (PDP) before access is allowed. The client call is intercepted by the PERMIS PEP, the DN of the user extracted and passed to the PERMIS decision engine, i.e., the PDP. Based upon the roles that a user must possess to access that service (or more commonly to access the method on that service) a decision is made by the PERMIS decision engine. This decision is based upon that user having that role (AC) assigned in the local LDAP. When this is the case an "allow" decision is made by the PDP, the PEP is informed and the client invocation subsequently passed on to the service itself.

Numerous other models of interacting with clients and PERMIS-protected Grid services also exist. These can be based upon the client pushing the ACs that are need to make authorisation decisions to the service. In this case the PERMIS will validate that the ACs are correctly signed (from a trusted SOA) and when this is the case, pass them on to the decision engine for an access control decision. Alternatively it might be the case that a pull mode of operation occurs whereby the PERMIS PEP pulls the needed attributes from one or more potentially remote attribute authorities (AA). These can be based upon centralized AAs such as VOMS or decentralised AAs such as Shibboleth IdPs. The advantages and disadvantages of these models are described in [10-11].

Irrespective of the model of interaction between a client and a Grid service, it is the case that the roles themselves (which are embedded within ACs along with the target and action information associated with that role) are static. The PERMIS decision engine when broken down it its most basic functionality undertakes a string comparison, i.e., does the role string that the user presents match the one that is needed to access that service according to its local authorisation policy stored in its own local LDAP? That is, is it a role in the role hierarchy above the level needed to access that particular service? In the case where the user is not presenting any role, i.e., where the authorisation decision is based upon extracting the DN only this check is equivalent to checking that the user has a local role in a local LDAP that matches the requirements for that service. In this case the string comparison is moot since the user either has or does not have the role (AC). However when a user is pushing roles to the PERMIS-protected service or PERMIS itself is pulling roles from an remote AA then this comparison is, at its basest form (ignoring the checks on validity of the signing authority for ACs) essentially a string comparison which either matches or not.

There is no higher reasoning that takes place that could be used to derive role equivalence other than that given in the statically defined role hierarchy.

Furthermore if the local policy needs to change, e.g., to add or change a given role in the existing role hierarchy, then the whole policy enforcement and decision points themselves need to be modified. In this case the associated policy in the PDP needs to be removed, a new one created, resigned and subsequently restored in the LDAP, and the PEP subsequently informed of the new Object Identifier (OID) that has been assigned to this new policy. During this time period, no access to protected services is possible - assuming that a single PDP is used to restrict access to services for various PEPs. Solutions addressing issues related to PEP/PDP proliferation and their dependencies are described in [12].

One obvious limitation with such RBAC-based solutions is the assignment of roles and in the meaning of roles. In large scale virtual organizations involving many individuals from many institutions from multiple domains, it is not always possible or realistic for a given provider to assign roles (ACs) to individuals directly. Roles will be VO-specific and targeted to the meaning implied by a given service provider in making its own local access control decision. Knowing that someone is a “lecturer” for example can have profoundly different consequences when making access control decisions on resources. As a trivial example, a lecturer at one institution might be allowed access to unlimited numbers of books from the library; at another institution they might be restricted to ten etc. It is the interpretation and meaning of the term “lecturer” that is essential when defining and enforcing access control policies.

It is also the case that the PEP is itself static. That is, whilst a PEP might be configured to pull down further information from other AAs that are needed to make local access control decisions, the configuration of the PEP is fixed with static trust relationships to remote AAs. When a local decision cannot be made with the information provided by a client, the PEP will search other trusted AAs for information that might be used to make access control decisions. In this, the reasoning and logic that can be applied to the ACs that might be returned is primarily syntactic, i.e., does the user have a role which equates to the role that matches the requirements to invoke the particular service that is being protected.

This scenario also depends upon the AAs themselves agreeing to release the appropriate attributes to the particular service to make its own authorisation decision. A naïve approach is where a PEP requests all/any attributes associated with a given individual. Particular trust relationships may mean that given AAs will be unwilling to release all attributes, e.g., due to privacy restrictions. In a clinical context this might happen where an AA is used for particular clinical trial. In this scenario, the local attribute release policy is highly unlikely to release these attributes when an access control decision is needed to access an inter-library book loan service for example.

Given this, it is essential that richer expressivity is achieved, both in how authorisation policies can be specified, and in how they can subsequently be enforced. Semantic technologies offer one mechanism to achieve this. However we also recognise that it is necessary to capitalise upon existing working solutions that the e-Science/Grid community has adopted, i.e., it would be naïve to assume that the body of experience that has been built up in application of X.509-based PKIs and technologies such as Shibboleth can simply be ignored. Thus our work has focused upon leveraging these solutions, and proposing extensions to authorisation-based models.

3. Semantic Web and Semantic-based Security

The semantic web is an intended successor to the current web. It aims to improve on the current ‘web of strings’ by specifying and deploying the technology which will let machines ‘understand’ enough that they can service our requirements better, for example by knowing that any statements made about a ‘healthcare worker’ should be immediately taken to apply also to a ‘nurse’. The technology that allows us to articulate this relationship, and state formally that a ‘nurse’ *IsA* ‘healthcare worker’ is an ontology, and is key to semantic web technology.

In the context of the semantic web, ontologies are most usually expressed in the Web Ontology Language (OWL, http://www.w3.org/TR/owl-features/), which is based on the Resource Description Framework (RDF, http://www.w3.org/TR/rdf-concepts/). RDF is a model for representing knowledge of all types, using assertions composed of the triple of a subject, predicate and object, where subjects and predicates are named with URIs, and objects can be either URIs or literals. As an example consider the two triples:

```
<urn:example#foo>  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
   <http://xmlns.com/foaf/0.1/Person> .
<urn:example#foo> <http://xmlns.com/foaf/0.1/name> "John Smith" .
```

These two assertions use a predicate and a class from the FOAF ontology (http://xmlns.com/foaf/0.1/), which is a well-known lightweight ontology for representing people and their relations and attributes. It says that the thing which has name <urn:example#foo> is a foaf:Person, and that it has foaf:name *"John Smith"*. These can be represented somewhat more readably with the Turtle notation (http://www.w3.org/TeamSubmission/turtle/):

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
<urn:example#foo> a foaf:Person;
   foaf:name "John Smith".
```

FOAF is thus a simple ontology. It formalises well-known notions such as *'Person'* and *'name'*, and makes extra statements about those predicates. Consider the slightly more extensive set of FOAF statements (presuming the same declaration of "*foaf*:" as above):

```
<urn:example#foo> a foaf:Person;
   foaf:name "John Smith";
   foaf:mbox <mailto:john@example.org>.
<urn:example#bar> foaf:mbox <mailto:john@example.org>;
   foaf:homepage <http://example.org/john>.
```

The FOAF ontology declares that the domain of the *foaf:mbox* predicate is a *foaf:Person*, and this means that a semantic web application which has ingested the set of FOAF axioms can immediately deduce that the thing named <*urn:example#bar*> is a *foaf:Person*, although it has not been explicitly declared as such. Another of the FOAF axioms states that at most one thing can be the subject of a *foaf:mbox* predicate with a given object. It therefore follows that <*urn:example#foo*> and <*urn:example#bar*> must be alternate names for the same thing, and thus that "*John Smith*" has home page <*http://example.org/john*>. These two examples illustrate what semantic web knowledge consists of, namely the facility for a machine to combine an ontology such as the FOAF axioms and instance data such as that given above, and deduce further instance data which is not explicit.

This example illustrates one of RDF's architectural strengths, in supporting data integration. The statements about <*urn:example#foo*> and <*urn:example#bar*> may have come from completely different sources, and be translations into RDF of very different databases, such as one organisation's LDAP service and another's personnel database. The primitiveness of the RDF model means that almost anything can be turned into RDF, and once there effectively integrated.

Given this starting point, we can promptly see how we can use this framework to support much more flexible access control policy specification and subsequent enforcement. If we decide that a particular confidential resource can be seen only by members of the class *ns1:HealthcareWorker*, which consists of the union of the classes *ns1:Doctor* and *ns1:Nurse*, and further declare that the class *ns2:Krankenschwester* is a subclass of *ns1:Nurse*, then immediately anyone declared to be of type *ns2:Krankenschwester* is provably a *ns1:Nurse* and thus also a *ns1:HealthcareWorker*. Crucially, all of the components of this chain of reasoning could potentially have come from different sources.

In this scenario, the policy information is expressed in the set of available classes, the logical relationships between them, and the grant of access to members of one of the classes.

This architecture has been made manifest in the AGAST project through a PDP service: *Qadi*, which is a web service providing a RESTful interface. A client of the service, e.g., a PEP, may define a policy as an ontology, as we have illustrated above. When a user attempts to access the service, the PEP obtains or extracts information about that user as RDF, and uploads that to a newly-created *'decider'*. It then asks the decider "*is the user X in the class of entities permitted access*", to which the Qadi service can reply with a simple yes or no, which the PEP can act upon by permitting or denying the user access.

The architecture of this semantic PDP is show in Figure 1. Key to this architecture is the fact that multiple security tokens can be used/translated into a format used to make semantic decisions.

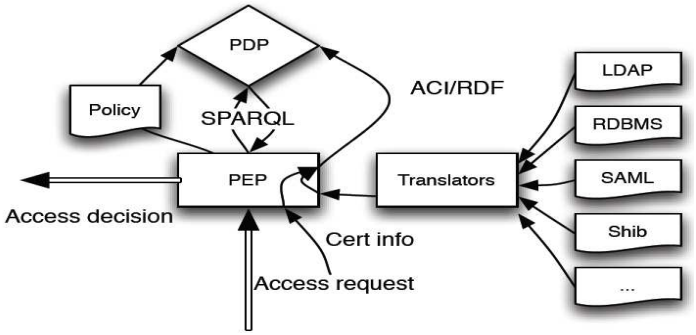

Figure 1: Semantic PDP Architecture

The PEP client can either itself generate RDF describing the user, or it can upload an X.509 user or attribute certificate proffered by the user to the Qadi service, which pulls apart the certificate and generates RDF statements which represent the information available within it, such as the certificate owner's CN or OU and the identity of the signer. The PEP subsequently interrogates the decider by posting a SPARQL query (http://www.w3.org/TR/rdf-sparql-query/). Currently the Qadi reasoner works with X.509 certificates however work is on-going to allow it to directly accept further sources of instance information.

In this model, the access control policy is therefore expressed in either or both of the ontology which is uploaded to the PDP by the PEP and the SPARQL query which the PEP subsequently makes (both of which are under the control of the PEP). The policy ontology may for example define a single 'AccessPermitted' class defined using a rich network of subclassing, equivalences and other logical apparatus, so that the subsequent SPARQL query has only to ask whether a user is a member of this class; or if it is more convenient the PSP may act as a little more than an integration point, and the PEP may effectively express the policy by using a much more elaborate SPARQL query.

4. Case Studies

To illustrate the benefits of taking a semantic-oriented approach to more flexible security policy specification and enforcement, we present two case studies based upon on-going projects at NeSC Glasgow.

4.1 VOTES Project

The MRC-funded VOTES project (www.nesc.ac.uk/hub/projects/votes) is focused upon supporting the various phases involved in clinical trials and epidemiological studies. These include recruitment, e.g. identification of individuals that may be approached for their involvement in a given trial/study; data collection throughout the course of a given trial/study, e.g. to ensure that individuals are attending clinics and their relevant details or drugs/placebos taken. Trial and study management is key to this to ensure that the right information is made available to the right individuals. As such fine-grained security is essential. A key aspect of the VOTES project is that it is not concerned with developing a single Grid infrastructure for a specific clinical trial or study, but with developing a Grid based framework through which a multitude of clinical trials can be supported. This was achieved through a framework that supported the creation of multiple different clinical virtual organisations (CVOs), each with different roles/privileges that allowed access to different clinical resources.

VOTES supported centralised security models and decentralised models based upon the Shibboleth technologies. The centralised model was supported through work on the VPman project.

4.1.1 The GT4 VOMS-PERMIS VOTES Scenario

The VPman (www.nesc.ac.uk/hub/projects/vpman) project planned to show how improved Grid based security can be achieved drawing on the strengths of both VOMS and PERMIS. Specifically it wished to integrate VOMS attribute assignment function with the PERMIS authorisation decision function. This was based upon the architecture shown in Figure 2.

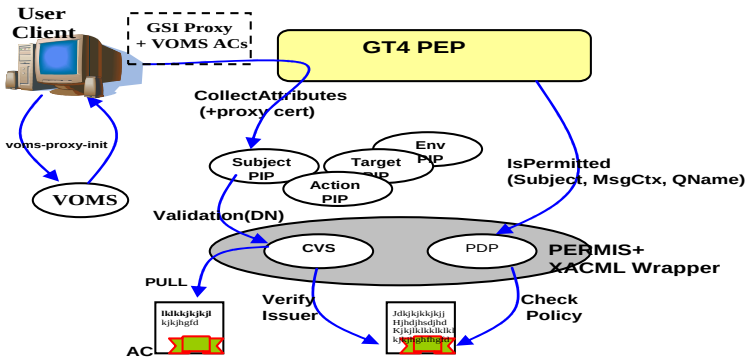

Figure 2: GT4 VOMS-PERMIS Integration

This architecture comprised an authorisation framework associated with GT4-based services. This framework provided capabilities to plug-in a series of interceptors to process each request when it is received, i.e. before it reaches the protected application. Two types of interceptors are of interest from an authorisation perspective: Policy Information Points (PIPs) and Policy Decisions Points (PDPs). The main task of a PIP is to prepare an appropriate component of the request context ready for it to be passed to the PDP for an access control decision. Typically there will be a PIP to prepare each of: the subject's attributes, the action's attributes, the resource's attributes and the environmental attributes. The relationship between PIPs, PDPs and the Policy Enforcement Point (PEP) with GT4, PERMIS and VOMS is shown in Figure 2.

VOMS is integrated with the Globus Toolkit so that the user's roles encoded as X.509 attribute certificates (ACs) can be passed around embedded in X.509 proxy certificates. A GT4 VOMS PIP allows GT4 to access and process the VOMS ACs (the Subject PIP in Figure 2). The VOMS PIP extracts the VOMS ACs from the proxy certificate, parses and stores the roles in the GT runtime so that they may subsequently be used by PDPs for making authorisation decisions.

The PERMIS Credential Validation Service (CVS) intercepts these roles and uses its policy to decide if they were correctly assigned by a trusted AA. VOMS roles may then be picked up from a VOMS SAML service given the DN of the user, or pushed from a tool such as Acacia. The valid set of user roles is passed back to the GT runtime for passing to the PDP (or other PIPs, depending upon the GT4 configuration).

To actually secure a GT4 service, it should be configured so that the required PIPs as well as a PDP must be called before access is granted. These PIPs will create the various components of an XACML request context and once all required information is collected, the PDP is passed a completed XACML request context. A protected GT4 service is configured with a security configuration and a service configuration. The former indicates the authorisation and authentication methods. In the authorisation method description, the PIPs and PDP are specified in the format of <identifier>:<java module> where *identifier* specifies a certain scope and *java module* is the full name of the java module which implements a PIP or PDP. The identifier for a PIP/PDP is used to differentiate between module instances and the parameters that need to be passed to each instance. Other services may use the same modules but with different configurations by using different identifiers. We note that the system has been designed to be extensible so that other PIPs or PDPs may be added to the authorisation chain.

A more specific explanation of the integrated VOMS-PERMIS-GT4 security solution for VOTES is shown in figure 3. The interactions between these components are as follows. Firstly, a VOTES service is deployed on a GT4 infrastructure (in this case the service was to support a type-2 diabetes trial). A user runs "voms-proxy-init" referring to the *VOTES-DiabetesVO* to generate a proxy certificate including VOMS credentials (related to their roles being either *votesdiabetes-doctor* or *votesdiabetes-nurse* in this particular trial) and tries to invoke the protected stored procedure which provides access to relevant clinical data from a variety of diabetes resources (including primary systems such as GPASS – used by 85% of GPs across Scotland, and secondary care resources such as Scottish Care Information store – used by all hospitals across Scotland for management of hospital data (inpatients, outpatients, lab data etc)). The PEP passes the user information (including proxy certificate) to the VOMS PIP. The VOMS PIP validates the credentials and passes back the VOMS Fully Qualified Attribute Name (FQAN) within the subject attributes. The PEP then calls the PERMIS PDP pushing the request information and credentials. The PERMIS PDP according to the policy then decides if this user with these attributes is authorised to access the service. Finally if successful, a stored procedure is invoked, which results in a (protected) federated query being run across various clinical resources, and results subsequently joined and returned to the end user. This joining is made possible through a unique identifier associated with the related diabetes clinical resources based upon the Community Health Index (CHI) number.

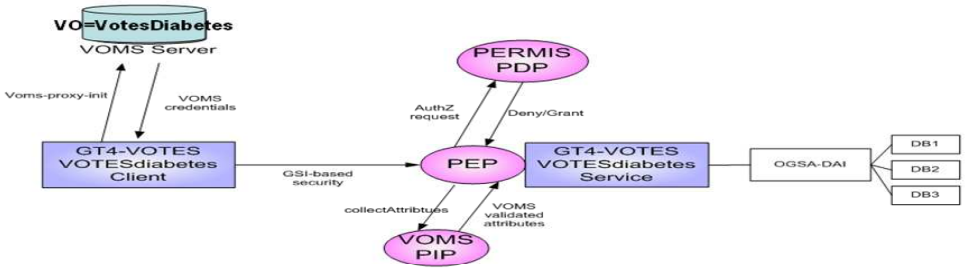

Figure 3: The GT4 VOMS-PERMIS VOTES scenario

This scenario has been implemented and supported but suffers from the limitations identified previously. Namely the PIP, PEP and PDP are statically configured to use the fixed roles (*votesdiabetes-doctor* or *votesdiabetes-nurse*) that are associated with the *VOTES-DiabetesVO*. A semantic web-based model of this system was thus produced in the AGAST project to illustrate the benefits over existing authorization approaches.

4.1.2 The GT4 AGAST-VOTES Semantic Scenario

In the AGAST project the architecture discussed in Figure 1 was constructed and applied to address limitations with existing authorisation solutions as experienced in VOTES. Specifically, the AGAST-based *VOTES-DiabetesVO* replaced the PERMIS RBAC solution with the aforementioned Qadi PDP.

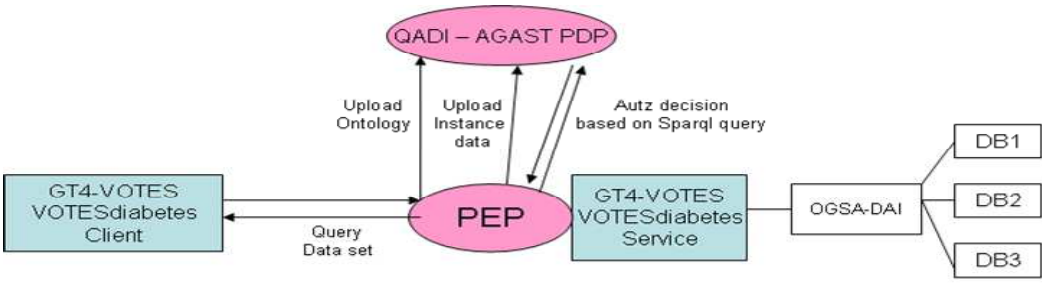

Figure 4: The AGAST-VOTES scenario

The interactions between these components to support semantic-based security are as follows. Firstly, a VOTES diabetes service is deployed on a GT4 infrastructure. An administrator then runs a PEP client to create a PDP instance on the Qadi service. Using the same client, the administrator then uploads an OWL ontology that acts as the security policy for the *VOTES-DiabetesVO*. This ontology consists of a hierarchy of roles and a hierarchy of access privileges associated with that trial. The administrator then uploads instance data to describe each user and their attributes. This can be in the form of an uploaded X.509 user or attribute certificate or just in RDF form. A *'decider'* is created. This step could also be performed by the user themselves if for example they possessed a valid VOMS proxy certificate and wanted to assert their attributes. Following this, a user invokes a query through the GT4-VOTES Client. This user must authenticate using an X.509 certificate which is used in GSI-based security on GT4. The PEP identifies the user using the DN extracted from their X.509 certificate. The PEP then interrogates the *'decider'* by posting a SPARQL query and makes an authorisation decision based on the instance data that has previously been uploaded for this user. If successful the stored procedure is invoked, the federated query run and returned results joined and returned to the end user.

4.1.3 Comparison of VOTES Scenarios

From the users perspective they must in both cases validate their identity using PKI – X.509 certificates. They can also assert their roles in both scenarios by pushing a VOMS-Proxy to the PEP, and if successfully authorised, they are returned results based on the privileges given to their respective roles. Thus there is little distinction from the end user perspective. The major difference occurs with regards to policy specification and policy enforcement. In the VOMS-PERMIS scenario, a key requirement is pre-agreement and assignment of VOMS-roles/attributes to individuals. To achieve this, static roles *votesdiabetes-nurse* and *votesdiabetes-doctor* were created and assigned to members of the *VOTES-DiabetesVO*. This resulted in PERMIS policies being created that defined what predefined datasets each VO member was authorised to access through the fixed query (or more precisely the GT4 method that invoked a stored procedure). It is also possible to make these privileges hierarchical. However RBAC-based hierarchies can in themselves be inflexible. Consider the real scenario where this infrastructure is used for an international Diabetes study. In this case, we may have an agreement made with a university hospital in Germany that their doctors and

nurses should have the ability to query the equivalent datasets. An agreement must be made that the roles *votesdiabetes-Krankenschwester* (Nurse) and *votesdiabetes-Arzt* (Doctor) must be added into the hierarchy. In theVOMS-PERMIS scenario it is not possible to derive such horizontal equivalences. Rather the RBAC hierarchy is vertical and inference of equivalences is not possible. To define that a role of *Krankenschwester* is semantically the same as *Nurse* would require a redefinition and redeployment of the static PERMIS policy since it cannot reason that one role hierarchy is equivalent to the next even though it is just named differently. This is a simple but worthwhile example that illustrates that the semantic approach is more flexible when role hierarchies begin to expand.

It is also true to say that the power and flexibility of the ontology (policy) that must be constructed and uploaded to the Qadi semantic reasoner is dependent on how well the ontology is defined in the first instance. There are tools available for this purpose such as Protégé (http://protege.stanford.edu/). PERMIS as a well established RBAC solution does allow for policies to be signed by PKI and made available on an LDAP server, which in turn allows for an authorisation decision to be made from multiple AA's. The VOM-PERMIS scenario for VOTES focused on VOMS as the AA.

Currently the Qadi reasoner works with X.509 certificates however work is on-going to allowing it to directly accept further sources of instance information. PERMIS is designed to integrate within the GT4 setup and has therefore quite a large installation overhead and maintenance of many configuration files, PKI for the policies and an LDAP server. Qadi has a relatively straightforward standalone setup being deployed on a Tomcat server.

4.2 SeeGEO Project

The JISC-funded SeeGEO project was primarily focused upon development of a Spatial Data e-Infrastructure that supported secure access to geospatial data under license to EDINA (www.edina.ac.uk) from the UK Ordnance Survey to the wider academic community. Specifically the data sets that were to be made available through the SeeGEO project included data related to UK borders, i.e. different regions such as local authorities. These areas change over time as local authorities/regions are redefined for a variety of reasons. A key aspect when conducting epidemiological and longitudinal studies is to understand the association of geo-spatially referenced data with resources such as UK border data sets.

The data sets that were explored within the SeeGEO project included the UK Census data from 1991/2001 with specific reference to data sets associated with health and well-being. These data sets were themselves geo-spatially referenced and included output areas such as partial postcodes. The scientific goal of this project was to show how it was possible to link historic clinical data from the UK Census with historical geo-spatial Borders data from EDINA.

The infrastructure that was developed was based around implementation of Open Geospatial Consortium (OGC) Web Map Service (WMS) which responds to requests by creating map images of spatial data; Web Coverage Services (WCS) which allow access to the raw data which can then be used for further analysis or for portrayal if required, and Web Feature Sets (WFS) which allow to add a range of features over a given map set. A Geo-linking Service (GLS) and client was supported that allowed to overlay various information (health variables) over mapping data.

The GLS client was accessed through a portal protected by Shibboleth (top Figure 5). To support this, a MyProxy server was used at the back end of the portal. End user invocations carry with them proxy certificate information from which the remote service policy definition point can use to extract the distinguished name (DN) of the end user. Knowing the identity of the user, the portal LDAP server can subsequently be queried for the attributes associated with that particular user. The authorization policy itself was based upon the license information a user possesses: namely whether they could see English output area information from 1991/2001 or Scottish output area information from 1991/2001. The roles used for this purpose were: *english_oa_2001, scottish_oa_2001*.

Figure 5 (bottom) also shows the mapping information associated with the health variables that were selected in the GLS client. We note that through GLS clients it is possible to overlay a whole range of information across mapping coordinates. The GT4-based GLS service was protected with PERMIS and when sufficient authorization information provided, i.e. that satisfied the licensing agreements with EDINA and hence with the commercial mapping provider Ordnance Survey, the resulting maps were rendered with specific health variables overlaid (depending upon the output area and Census years selected).

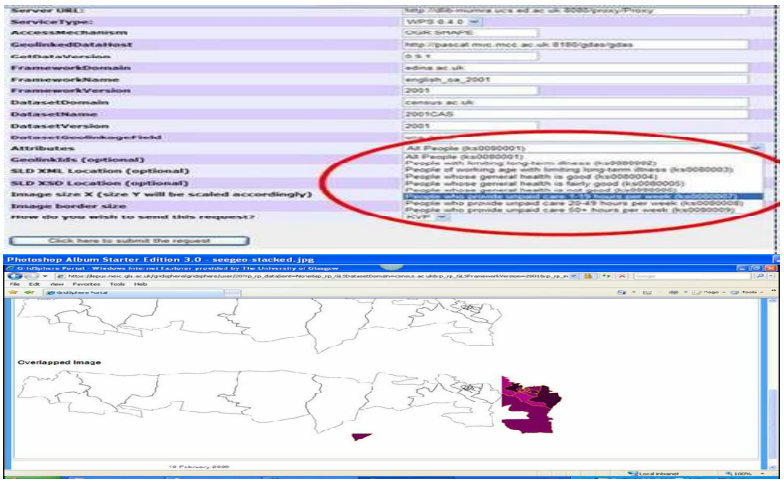

Figure 5: Geo-Linking Service Client (top) and Geo-spatial Results

The semantic web based architecture for the SeeGEO demonstrator was the same as that used within the VOTES demonstrator. The distinction here, being the roles that were used by Qadi to enforce its authorization decision were SeeGEO specific, e.g. *scottish_oa_2001*. At the time of writing, the SeeGEO semantic security based demonstrator has replicated the RBAC based security model. We note that work is progressing on highlighting richer cases studies that will demonstrate the advantages of semantic web based security. One avenue that we are pursing in this regard is how the semantic *reasoner/decider* can exploit geospatial information itself for authorization decisions. Thus a user may request a particular postcode as the output area of interest, but this might provide too much information which could potentially be used to identify the individuals themselves. In this case it might well be the case that geospatial data can only be rendered to only local area boundaries.

5. Conclusions

Security-oriented clinical collaborations and epidemiological studies require simple and robust security policy specification models and their enforcement. We have shown that semantic-based approaches can exploit a variety of authorization credentials and allow far greater reasoning to policy enforcement than existing solutions such as RBAC. The work described here has shown the proof of concept and demonstrated its applicability. The extensions to this work are numerous. We are currently exploring semantic policies for job submission and wider resource management in many other domains such as Virtual Observatories for astronomical data and nanoCMOS electronics amongst many others.

5.1 Acknowledgements

This work has been made possible by grants from the Joint Information Systems Committee (JISC) and the Medical Research Council. We gratefully acknowledge their support.

6. References

[1] J. Watt, R.O. Sinnott, O. Ajayi, J. Jiang, J. Koetsier, *A Shibboleth-Protected Privilege Management Infrastructure for e-Science Education*, 6th IEEE International Symposium on Cluster Computing and the Grid, CCGrid2006, May 2006, Singapore.

[2] R.O. Sinnott, J. Watt, O. Ajayi, J. Jiang, *Shibboleth-based Access to and Usage of Grid Resources*, IEEE International Conference on Grid Computing, Barcelona, Spain, September 2006.

[3] R.O. Sinnott, O. Ajayi, A.J. Stell, J. Watt, J. Jiang, *Single-Sign on and Authorization for Dynamic Virtual Organizations*, International Conference on Virtual Enterprises, (PRO-VE'06), Helsinki, June 2006.

[4] A.J. Stell, R.O. Sinnott, O. Ajayi, *Grid Infrastructures Supporting Paediatric Endocrinology across Europe*, UK e-Science All Hands Meeting, Nottingham, UK, September 2007.

[5] A. Schaad, J. Moffett, J. Jacob, *The Role-based Access Control System of a European Bank: a Case Study and Discussion*, Proceedings of the sixth ACM symposium on Access control models and technologies, Chantilly, Virginia, United States, 2001

[6] G. Neumann, M. Strembeck, *A Scenario-driven Role Engineering Process for Functional RBAC Roles*, Proceedings of the 7th ACM symposium on Access control models and technologies, Monterey, California, USA, 2002

[7] R. Alfieri, et al. *VOMS: an Authorization System for Virtual Organizations*, 1st European across Grids conference, Santiago de Compostela.

[8] ISO 9594-8/ITU-T Rec. X.509 (2001) The Directory: Public-key and Attribute Certificate Frameworks

[9] D.W.Chadwick, A. Otenko, E.Ball, Role-based Access Control with X.509 Attribute Certificates, IEEE Internet Computing, March-April 2003.

[10] R.O. Sinnott, D. Chadwick, T. Doherty, D. Martin, A. Stell, G. Stewart, L. Su, J. Watt, *Advanced Security for Virtual Organizations: Exploring the Pros and Cons of Centralized vs Decentralized Security Models*, 8th IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2008), May 2008, Lyon, France.

[11] R.O. Sinnott, J. Watt, D.W. Chadwick, J. Koetsier, O. Otenko, T.A. Nguyen, *Supporting Decentralized, Security focused Dynamic Virtual Organizations across the Grid*, 2nd IEEE International Conference on e-Science and Grid Computing, Amsterdam, December 2006.

[12] Q. Wei, M. Ripeanu, and K. Beznosov, *Authorization Using the Publish-Subscribe Model*, IEEE International Symposium on Parallel and Distributed Processing Systems with Applications, Sydney Australia, December 2008