David Chadwick, University of Kent
29 October, 2009

Functional Components of Grid Service Provider Authorisation Service Middleware

Status of This Document

This document provides information to the Grid community. Distribution is unlimited.

Abstract

This document describes the various components that make up the authorization decision function of a Grid service provider. It looks at the different ways in which the various components can be combined together, and data flows between the components. This document is for informational purposes only and is not intended to form a grid standard.

Contents

Abstract 1
1. Introduction 2
2. Notational Conventions 2
3. Definitions 2
4. Model 3
5. Functional Composition 4
6. The Context Handler 6
7. Relationship of CIS, CVS to STS and PIP 7
8. Security Considerations 7
9. Contributors 7
10. Intellectual Property Statement 8
11. Disclaimer 8
12. Full Copyright Notice 8
13. References 8

1. Introduction

This document describes the functional components that make up the authorization decision function of a Grid service provider. It looks at the different ways in which the various components can be combined together, and data flows between the components. This model is compared to the XACMLv2 model [XACML], and the differences noted.

2. Notational Conventions

The key words 'MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," and "OPTIONAL" are to be interpreted as described in RFC 2119 [BRADNER1]

3. Definitions

Attribute is a property or characteristic of an entity.

Attribute Authority (AA) is an entity (the issuer) that asserts attributes about another entity (the subject or holder).

Attribute Assertion is a statement made by an AA that a subject possesses a particular set of attributes.

Authorisation Credential (subsequently abbreviated to credential in this document) is an attribute assertion digitally signed by the issuer (i.e. it is a security token) so that it can be cryptographically validated.

Attribute Release Policy. A policy held by a Credential Issuing Service that says who should be allowed to request attribute assertions about whom.

An authentic attribute assertion [authentic authorisation credential] is one that was issued by the AA that purported to issue it, and has not been revoked since it was issued.

A valid attribute assertion [valid authorisation credential] is an authentic attribute assertion [authorisation credential] that is trusted by the resource's authorisation service to grant some form of access to the resource. For example, a project manager credential issued by university A is both authentic and valid for use within university A, but may only be authentic and not valid for use within bank B. Note that if an attribute assertion [authorisation credential] contains multiple attributes, it might be authentic but only partially valid.

Context Handler. The entity that is responsible for handling the communications between the PEP, the CVS, and the PDP.

Credential Issuing Service (CIS). An application independent service of an AA that issues authorisation credentials about entities.

Credential Validation Service (CVS). An application independent policy engine that validates authorisation credentials (or security tokens) and returns the valid attributes of the subject (which may be a subset of the attributes in the attribute assertion)

Grid Service Provider (SP). The application dependent and independent software that provides an authorised user with access to a grid resource or grid service.

Identity Provider – is a type of Credential Issuing Service that is a combined Attribute Authority and Authentication Service that can authenticate users and provide attribute assertions about them.

Policy Decision Point (PDP). An application independent policy engine that makes authorisation decisions based upon its policy and information about the subject and the requested mode of access.

Policy Enforcement Point (PEP). That part of an application that enforces the results returned from a policy decision point

Policy Information Point (PIP). An application independent service that acts as a source of attribute values [XACML]

4. Model

Authorisation credentials are issued by a Credential Issuing Service (CIS), in which the user is the holder/subject and the CIS is the issuer. These authorisation credentials will give the user the necessary rights to access a grid service provider (SP). These authorisation credentials may be pushed to the SP by the user (or an entity acting on behalf of the user) or pulled by the SP from the CIS, or a mixture of both. The CIS will have a policy (an attribute release policy) that will say who is entitled to receive the issued credentials. Some CISs (such as credit card issuers) may only issue credentials to their rightful holders; others (such as the Shibboleth AA) may issue them to trusted SPs. This is determined by the CIS's attribute release policy.

The SP has a policy that says which credentials are acceptable (or trusted) and which attributes are needed in order to gain access to the service. Unacceptable credentials are ignored by the SP. If a user has insufficient attributes he is denied access; if he has greater of equal to the required attributes he is granted access.

The SP software comprises application dependent and application independent code. We are only concerned with modeling the application independent code and the interfaces between the application dependent and application independent code.

The authorization decision function is application independent code. The following functional components are involved:

i) a Credential Retriever – this functional component is responsible for pulling credentials from one or more AAs when insufficient credentials are provided by the user
ii) a Credential Decoder – this functional component is responsible for parsing credentials and storing them in a local internal representation ready for passing to the credential validator. A system may have several credential decoders, in order to handle credentials in different formats e.g. SAML assertions, X.509 PKCs, X.509 ACs, proprietary credentials, etc.
iii) a Credential Authenticator – this functional component ensures that a credential is authentic, i.e. that it really was issued by the AA that claimed to have issued it and it has not been revoked since it was issued. This usually entails checking that the digital signature on the credential is valid. Since credentials come in different formats, different credential authenticators will be needed. Some credential authenticators may need access to CRLs if the credentials are long lived. Others may not, if the credentials are short lived and are guaranteed to never be revoked.
iv) a Credential Validation Policy Enforcer – this functional component is responsible for validating an authentic credential, i.e. it ensures that a credential is trusted according to the resource's policy rules.

v) a Credential Validation Service – this functional component returns a set of valid attributes for a user, optionally given a set of credentials for the user. It encapsulates the functions of the credential retriever, decoder, authenticator and policy enforcer.

vi) a Policy Decision Point – this functional component is responsible for returning an authorisation decision given the user's access request and optionally the user's valid attributes or the user's credentials. Before the PDP can make an authorization decision, it has to either be given the validated attributes of the user or validate the credentials itself. Note that the user may provide any arbitrary set of credentials, for example, member of university X, member of grid project Y, registered doctor, certified engineer, etc. issued by any arbitrary set of attribute authorities (AAs).

5. Functional Composition

The functional components can be constructed in various ways. Figures 1 to 4 show the different ways in which the CIS, CVS and PDP can be integrated with the PEP. The fundamental difference between the 4 modes of construction is how the PEP interacts with the authorization service, whether it:

1. Pushes credentials to one or more CVSs and then pushes the valid attributes to the PDP for an authorization decision (Figure 1)
2. Pushes credentials to the PDP for an authorization decision (Figure 2)
3. Passes the user's authenticated name or ID and meta-information to one or more CISs to one or more CVSs and then pushes the valid attributes to the PDP for an authorization decision (Figure 3) or
4. Passes the user's authenticated name or ID and CIS meta-information to the PDP for an authorization decision (Figure 4).

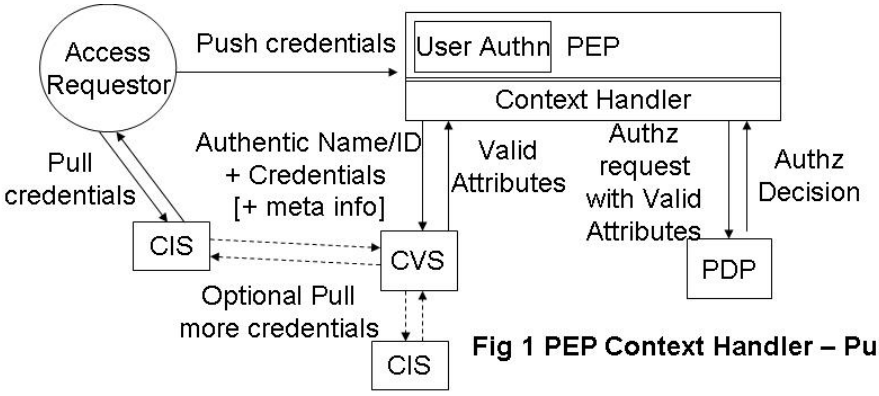


Fig 1 PEP Context Handler – Push Credentials

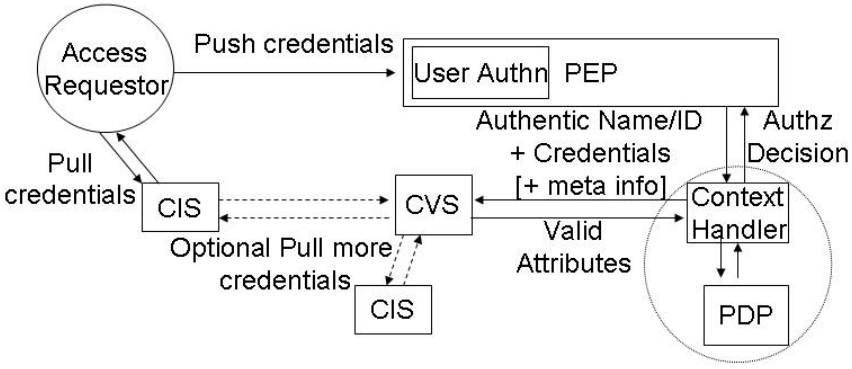


Fig 2 PDP Context Handler – Push Credentials

Examples of all 4 modes of operation are already implemented. The PERMIS authorization system has implemented all 4 modes of operation; the GGF Authz SAML protocol [GFD66] which has been implemented by several different groups, specifies figures 2 and 4; the GridShib project has implemented Figures 1 and 3; whilst Globus Toolkit 4.1+ implements the PDP functionality of Figures 1 and 3.

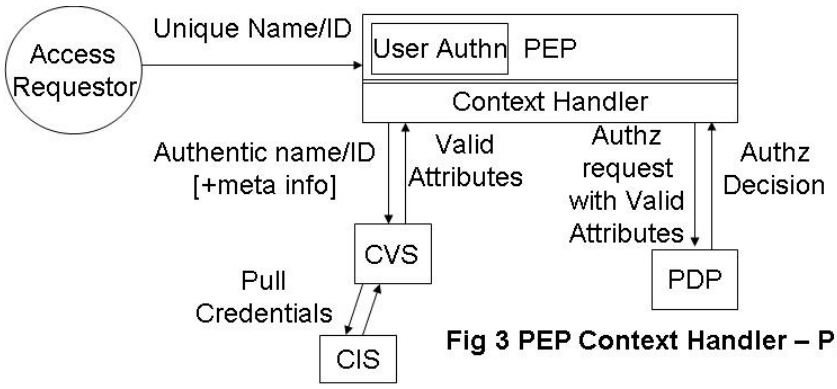


Fig 3 PEP Context Handler – Pull Credentials

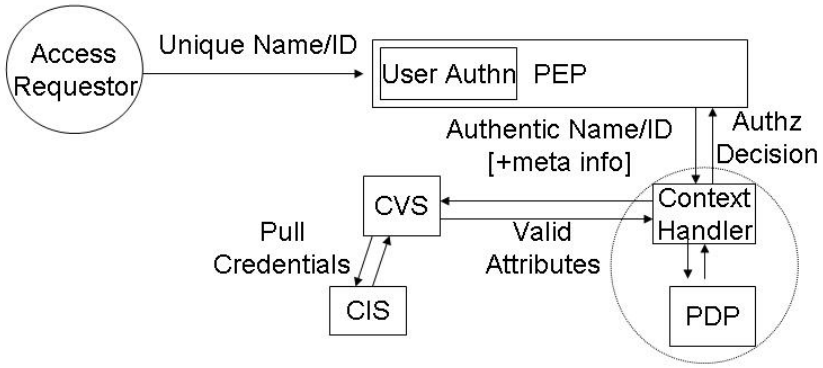
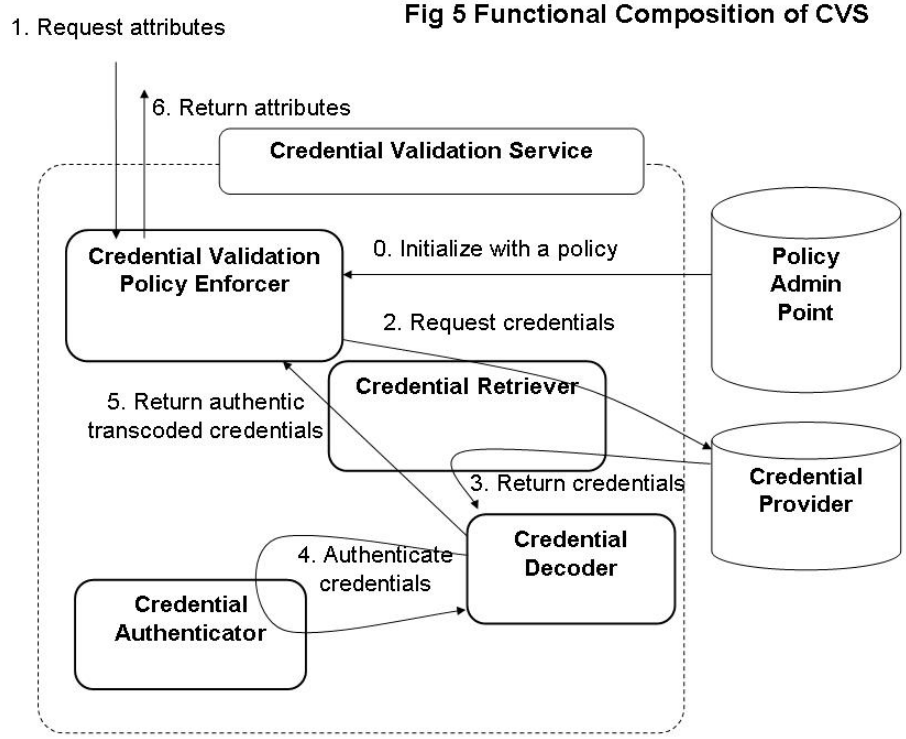


Fig 4 PDP Context Handler – Pull Credentials

Figure 5 shows how the CVS functionality can be constructed from the various functional components that operate with credentials.

Fig 5 Functional Composition of CVS



6. The Context Handler

The context handler has three interfaces, one for talking to the PEP, a second for talking to the CVS and a third for talking to the PDP. Each of these three interfaces could be APIs or open protocols. The functionality required of the three interfaces is as follows

PEP-Context Handler.
PEP→CH, the authenticated name or ID of the user, the credentials of the user (optional), meta information to one or more CISs (optional), and the user's access request
CH→PEP, the authorization decision plus optional obligations.

Context Handler-CVS.
CH→CVS, the authenticated name or ID of the user, the credentials of the user (optional), and meta-information to one or more CISs (optional)
CVS→CH, the validated attributes of the user.

Context Handler-PDP.
CH→PDP, the authenticated name or ID of the user, the validated attributes of the user and the user's access request.
PDP→CH, the authorization decision plus optional obligations.

One can see that the PEP→CH and CH→PDP protocols are very similar. The only difference is that the former optionally passes credentials and CIS meta-information, and the latter passes validated attributes. These could easily be combined into one protocol if there is a way of signaling the difference between an attribute, a credential and CIS meta-information. This is the approach taken in the OGF protocol profile **Use of XACML Request Context to Obtain an Authorisation Decision** [OGFXACML].

The CH→CVS protocol is quite different to the previous protocol, and this is the subject of a separate OGF profile **Use of WS-TRUST and SAML to access a CVS** [OGFCVS].

7. Relationship of CIS, CVS to STS and PIP

WS-Trust [WSTRUST] is a proposal from Microsoft, IBM and others[1] that enables security token interoperability by defining a request/response SOAP protocol whereby clients can request from some trusted authority that a particular security token be exchanged for another one. The security token service (STS) is the trusted authority that responds to WS-Trust requests.

Madsen[2] identifies that an STS actually has three different functionalities, namely: security token exchange, security token issuing and security token validation. The last two functions are special simplified cases of the first. In this document we are interested in the two simplified functions, security token (or credential) issuing and security token (or credential) validation. Therefore we have decided to give these specialized functions their own names – the credential issuing service (CIS) and credential validation service (CVS) – rather than the generic name STS, since STS implies a much greater functionality than that which is required here.

XACML [XACML] is a proposal from OASIS that defines a language for expressing access control policies in XML. XACML has nothing to say about security tokens or credentials. The nearest it comes is to define a Policy Information Point (PIP) as the system entity that acts as a source of (asserted) attribute values. Since the CVS described in this document is a source of attribute values that are ready to be passed to an XACML conformant PDP, then one can consider that the CVS is a specialized type of PIP that can process credentials and/or security tokens according to a credential validation policy, and that can return valid attributes in exchange for the input credentials.

The only difference between an attribute assertion and a credential is the digital signature of the latter. If there is a trusted connection between a CIS and the PEP or PDP, then the digital signature isn't needed, and the CIS could issue unsigned attribute assertions ready for consumption by the PDP. In this case, the CIS is acting as a PIP, since it is a source of attributes. In grid environments it is not usually the case that we have trusted connections between entities, and therefore credential issuing and credential validation services will usually be needed. The OGF protocol profile for fetching credentials from a CIS is specified in [OGFAA].

8. Security Considerations

This entire document is concerned with security.

9. Contributors

Author: David W. Chadwick
The Computing Laboratory
University of Kent
D.W.Chadwick@kent.ac.uk

[1] The WS-Trust specification is available from ftp://www6.software.ibm.com/software/developer/library/ws-trust.pdf
[2] Paul Madsen "WS-Trust: Interoperable Security for Web Services" Available from http://www.xml.com/pub/a/ws/2003/06/24/ws-trust.html

The author would like to thank members of the OGSA-Authz group for their contributions to this document, and in particular: Yuri Demchenko, Tom Scavo and Richard Sinnott.

10. Intellectual Property Statement

The OGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the OGF Secretariat.

The OGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the OGF Executive Director.

11. Disclaimer

This document and the information contained herein is provided on an “As Is” basis and the OGF disclaims all warranties, express or implied, including but not limited to any warranty that the use of the information herein will not infringe any rights or any implied warranties of merchantability or fitness for a particular purpose.

12. Full Copyright Notice

13. References

[BRADNER1] Bradner, S. Key Words for Use in RFCs to Indicate Requirement Levels, RFC 2119. March 1997.
[BRADNER2] Bradner, S. The Internet Standards Process – Revision 3, RFC 2026. October 1996.
[CATLETT] Catlett, C. GFD-1: Grid Forum Documents and Recommendations: Process and Requirements. Argonne, Illinois: Global Grid Forum. April 2002.
[GFD66] Von Welch, Rachana Ananthakrishnan, Frank Siebenlist, David Chadwick, Sam Meder, Laura Pearlman. “Use of SAML for OGSI Authorization”, GFD.66. March 2006,
[OGFAA] V. Venturi, T. Scavo, D.W. Chadwick, “Use of SAML to retrieve Authorization Credentials”, OGF GWD-R-P, 25 June 2009

[OGFCVS] David Chadwick, Linying Su. “Use of WS-TRUST and SAML to access a Credential Validation Service”. OGF GWD-R-P, 25 June 2009
[OGFXACML] David Chadwick, Linying Su, Romain Laborde. “Use of XACML Request Context to access a PDP”, OGF GWD-R-P, 25 June 2009
[SAML] OASIS. “Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0”, OASIS Standard, 15 March 2005
[SAMLPROF] OASIS “Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0”, OASIS Standard, 15 March 2005
[WSTRUST] Anthony Nadalin (Editor) “Web Services Trust Language (WS-Trust)”, Feb 2005 available from ftp://www6.software.ibm.com/software/developer/library/ws-trust.pdf
[XACML] “OASIS eXtensible Access Control Markup Language (XACML)” v2.0, 6 Dec 2004, available from http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml