

White Paper on Enterprise Authorization

Large organizations typically operate through a federated structure, with control over knowledge, process, system and data decentralized across various business units. Federated operation is often an essential ingredient for success of these enterprises. However, large enterprises would also like some assurance that access to all its federated systems are controlled and that access is available only to the right users for the right reasons. The problem really is finding a balance between providing this enterprise level assurance, without disturbing the essential federated nature of these enterprises.

True federated authorization envisages a loose federation of independent organizations, which have an association of sorts that calls for sharing of systems and data. The issues involved in such federations, where diverse interests have to be catered to closely approximates to a business entity that has chosen a federated mode of operation. This paper addresses the access authorization issues involved in such federated entities.

In most traditional organizations, data access security is typically governed by application-specific mechanisms, each with its own set of users, roles and access control policies. More often than not, authorization is hardwired in a disjointed way into organizational silos and segmented across domains and applications. This fragmentary, inconsistent, and possibly incomplete authorization mechanism carries a significant level of risk. An SOA environment magnifies this risk, since data sources are typically rendered more visible through the organization. The picture below depicts the required state, the current state, the gap and the issues to be primarily addressed to achieve enterprise assurance of authorization.

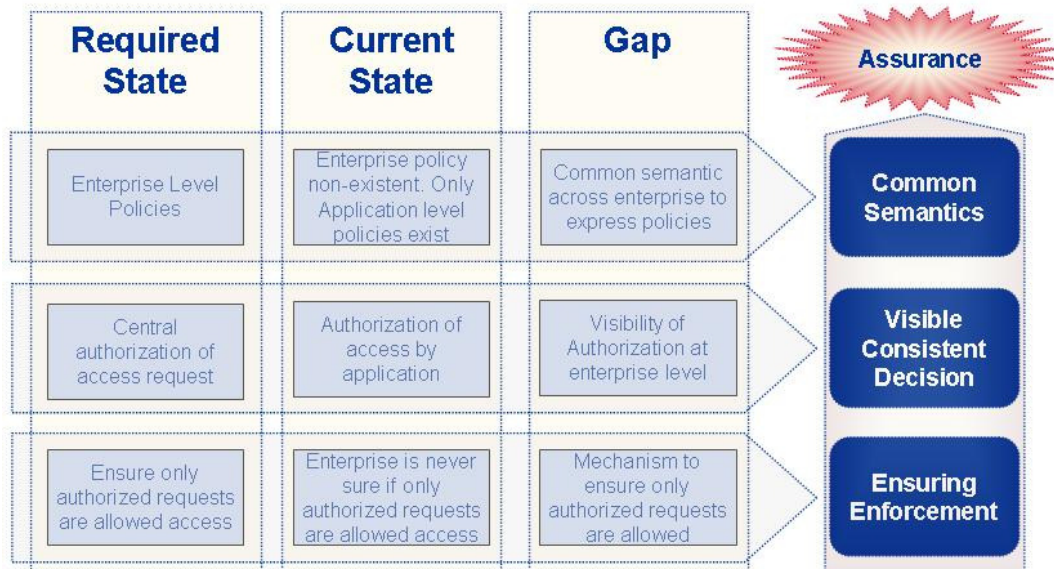


Figure 1 – High Level Perspective of Common Authorization

The issues that require address are briefly discussed below:

- **Common semantics** - A common authorization semantic across the enterprise could serve as the cornerstone for a well founded security strategy for a federated organization. Introduction of such semantics and a well defined authorization interface for services consumers and providers could significantly drive down the security related costs and effort involved in implementation of business changes. One of the first orders of business to implement federated authorization is to generate a common understanding/definition of authorization across the federation and create unified semantics through which to express authorization policies.

- Visible Consistent Decision – The decision making process for authorization should be consistent across the enterprise and should be visible at the enterprise level. This clearly makes a case for the authorization decision to be centralized, (at least to extent required to generate assurance of authorization) and not delegated to the individual applications.
- Ensuring Enforcement – The authorization decision taken centrally should be enforced by each application, i.e. only authorized requests can be executed.

Central Authorization

To achieve assurance of authorization, it is essential that all access requests are routed through an authorization decision point and that we have mechanisms in place to ensure that these decisions are enforced. An effective way of achieving this would be to have a central authorization controller, which ensures that all access requests are subject to authorization decision. We could then generate an authorization token or a virtual user profile, which accompanies the access request to the application. The application will enforce the authorization decision by allowing only such access requests that are accompanied by the authorization token. The picture below represents a logical view of such an implementation.

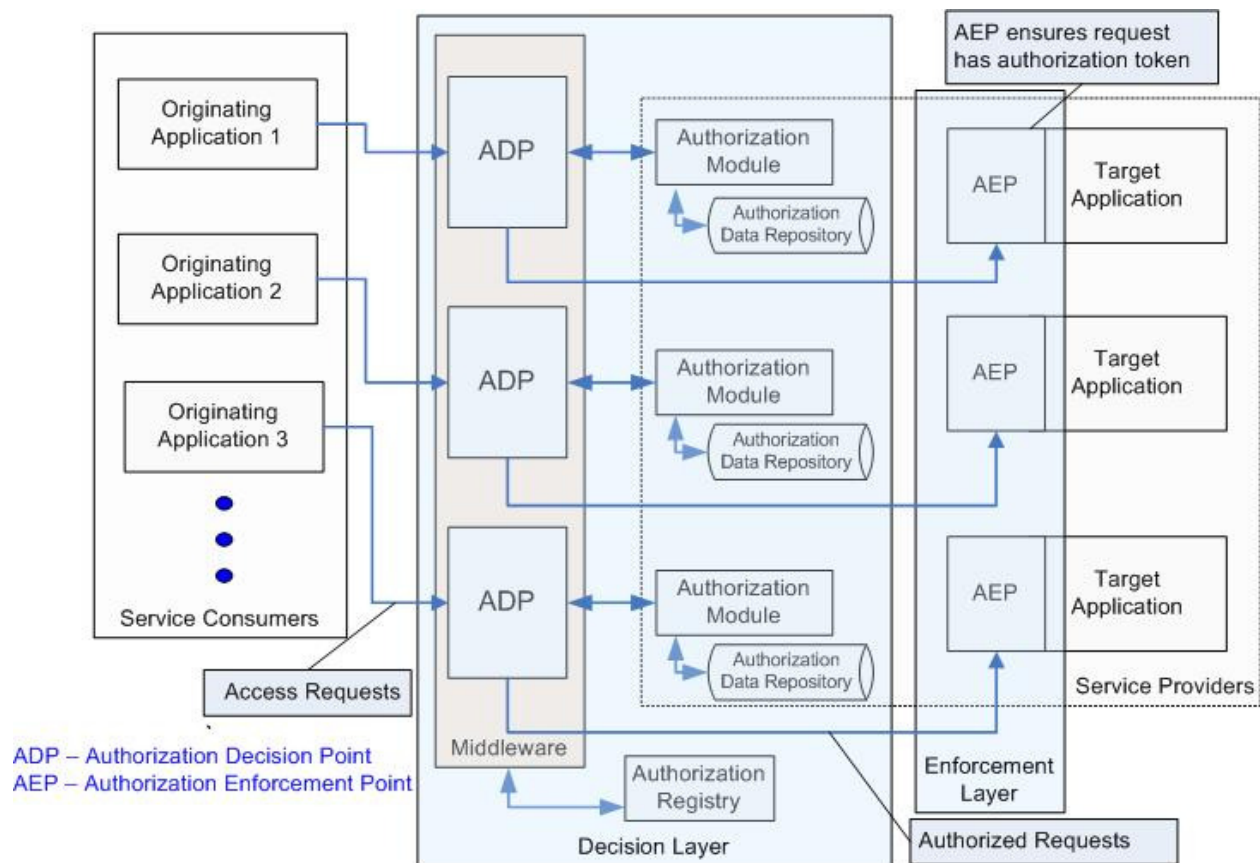


Figure 2 - Central Authorization - Logical View

The layers/components of the above diagram are discussed below:

- Decision Layer - Access to all applications necessarily goes through a formal decision layer, thus ensuring that the authorization decision is not bypassed.
- ADP – Authorization Decision Point - ADP is responsible for locating the authorization module, obtaining authorization and directing the authorized request to the target application. ADP also generates an authorization token, which accompanies the authorized request.
- Enforcement Layer – All applications in the secured domain are protected by an enforcement layer, which ensures that only requests that are authorized by the decision layer are permitted to execute.
- AEP – Authorization Enforcement Point - AEP is an application-specific component that is responsible for ensuring that authorization carried out by ADP is enforced. AEP typically reviews the authorization token, accompanying each request for this purpose.

While it is clear that the enterprise should move towards central provisioning of access authorization, the question that has to be addressed is the level of centralization. At one end of the spectrum, we could conceive total central control of access authorization, where individual applications have no accountability for authorization. In practical terms, this is very difficult to administer as the enterprise does not have easy access to application data to allow for a fine grained authorization decision. There would be performance penalties to pay if we incorporate convoluted access to application data in the access request life cycle. Further, such an extreme step could impact the federated nature of the enterprise.

The other possibility is layering, which conceives two tiers of access authorization. At the enterprise level, we could have a relatively coarse-grained first-tier authorization, which is adequate to meet the requirements of enterprise level assurance. The individual applications can then impose finer grained second-tier authorization, which may be more domain-specific. A typical example would be a trader, who is authorized to place orders by the enterprise, but is subject to restrictions on the volume of trade by the individual application or domain.

The demarcation of authorization between the enterprise and the application could be contentious, specially, if change is involved from a legacy mode of operation. The guidelines for deciding enterprise involvement in authorization has to be built into the enterprise policies.

Modeling to Arrive at Common Semantics

We suggest a model to create the authorization constructs that will allow us a unified way of representing the authorization semantics and that would cover most of the normal authorization scenarios. The typical authorization request has a syntax, which is very similar to English grammar, in that there is a subject and a predicate. The subject deals with who is making the request and the predicate deals with what the request is. The predicate is also often referred to as permission. Besides the subject and predicate, the other typical elements of the model are 'Protected Resources' and 'Constraints'.

Subject

The subject is the representation of the user, e.g. login details. Roles are assigned to the user and permissions are assigned to roles, and inherited by the user. Each role implies a set of tasks and responsibilities. To effectively carry out these tasks and responsibilities, the user will require a set of access privileges or entitlements, which translate to permissions, assigned to the user.

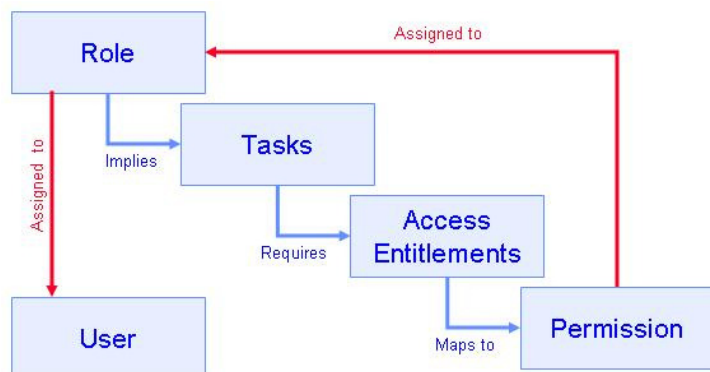


Figure 3 - Permissions for Roles

Predicate

The predicate is a composite term, which has action and object as its major constituents. The action is typically a single worded verb, such as 'Enter', 'Delete', 'Update' etc. The object answers to the question "Action what?" For instance, the predicate could be 'update account profile', where 'update' would be the action and 'profile' would be the object, with 'account' being an attribute of the object. We may have to extend the predicate with additional object qualifiers, e.g. 'Update account profile for trade limit' and such extensions ('for trade limit') could enhance the granularity of authorization.

Protected Resource

The protected resource typically answers to the question – 'which business object does this access permission impact?' One of the first questions to be answered while authorizing access authorization will be "Is the user allowed to access this protected resource?" Once this question is answered in the affirmative, the specific permission can be authorized.

Constraints

Besides the basic permission, there will be constraints, which have to be addressed. Some examples of constraints are

- A user is permitted to trade, but there are volume restrictions – while trading is considered as permission, the volume restrictions are considered as constraints
- A user is allowed to update an instrument, but only if the related account is in a 'Pend' status – again the basic permission is for the updating the instrument, but the status of the related account is an additional restriction, which is a constraint.

SecureUML model

The above syntax is well-articulated using a SecureUML model, which is based on RBAC, a synonym for 'Role Based Access Control'. The fundamental concept of RBAC lies in insulating the user from the permission, by assigning permissions to roles, which are then assigned to users.

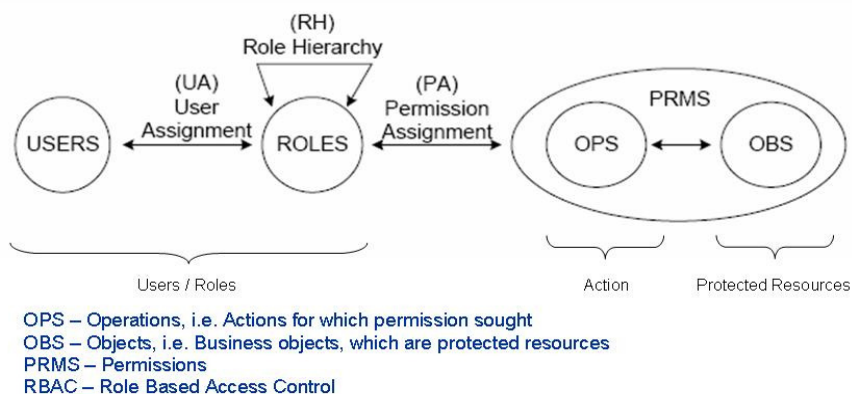


Figure 4- RBAC Model

Once we decide on the model, individual permissions can be analyzed to arrive at a common semantic, which can then be used to express enterprise level authorization policies.

Conclusion

We have to strike the right balance between the federated nature of large enterprises and assurance of authorization, which could call for central involvement. In traditional enterprises, data access is typically controlled by individual applications. This results in fragmentary, inconsistent, and possibly incomplete authorization mechanisms. In the context of a service oriented enterprise, the risks of such a scenario would be magnified.

Central authorization could be a solution to the remedy, but a total central implementation could be an overreach. A good via media could be two-tiered authorization, where the enterprise carries out coarse-grained authorization for assurance, while individual applications impose finer restrictions as dictated by business.

The essential ingredients to achieve assurance of authorization are modeling and generating common semantics, creating enterprise policies, and architecting decision / enforcement layers. Creating enterprise policies calls for common semantics, i.e. a unified mode of expressing the policies. The decision layer ensures that the authorization decision is not bypassed and the enforcement layer ensures that the authorization decisions taken are enforced.

Common semantics is an excellent starting point for instituting central authorization. The first order of business to generate common semantics is deciding on a model that would comprehensively cover the range of permissions that the enterprise deals in. SecureUML, based on role-based access control model (RBAC) is a competent alternative to generate common semantics.

About Object Edge

Object Edge (<http://www.objectedge.com>) is a high-end business & technology services company with proven track record in enterprise/application architecture, IT strategy and integration services. We work at the junction of business and technology. The above paper is a high-level summary of the company's experiences in the area of federated authorization. For details, please contact Ram Subramanian (sram@objectedge.com) or Mohan Parthasarathy (mohan.parthasarathy@objectedge.com).