resourcemove_scenario

Scenario: Moving Resources Between Hosts (Pending)

Submitted by: Maciej Machulak

In a rapidly developing Web environment there is a wide spectrum of various Web applications that combined together provide users with a powerful set of features. New interesting Web applications are being offered on a daily basis by corporations, small start-ups or single developers.

As new applications are being delivered, it is often the case that Web users wish to test functionality of those applications. Often they decide that a new application meets their needs and should substitute their old application. Apart from a better functionality, an application may additionally compete on the price which plays an important role when choosing which Web application to use.

In a situation where a user decides to switch between two or more Web applications then resources need to be transferred between those applications. Most commonly, the user will need to download all resources from an old application and upload them to a new one. In some cases, tools are provided to support users with this time consuming process. However, security settings and access control policies need to be manually set up on a new Web application and cannot be reused from an old one.

When a user wishes to transfer resources from one application to another, be it manually or automatically, then the problem of access control can be easily resolved if those applications support User-Managed Access. Once resources are transferred a user may simply plug in their existing and already configured Authorization Manager to a new application. Then a user may apply already composed access control policies for the same set of resources which is hosted by a new Web application.

In the next section we present how a User-Managed Access can be used to support users when resources are moved between Web applications (Hosts). We show how our approach allows a user to compose access control policies for their resources once and apply them independently of Web applications that host those resources.

(**NOTE:** All references to real Web apps are hypothetical.)

Use Case: Moving Resources Between Web Applications (Pending)

Submitted by: Maciej Machulak

Caroline is using the Web mostly to share short video clips and pictures of herself with her friends. She is using YouTube and Picasa Web Albums for that purpose. She is very happy with those services and is particularly pleased with their reliability and the fact that those services are free to use for non-commercial users.

Caroline is very security conscious. She wants all of her video clips and pictures to be well protected and only accessible by legitimate users (i.e. her friends). She is fine with spending some time every week to make sure that access control policies are in place to protect her resources. She is using an Authorization Manager for that purpose. All of her access control policies are composed centrally and are applied to her resources hosted by YouTube and Picasa.

Apart from pictures and video clips, Caroline stores her documents online as well. She uses Office Live Workspace as it is integrated with her word processor. She can access her documents over the Web from almost any place in the world. She configured Office Live Workspace to delegate access control to her Authorization Manager - the same which is used for Picasa and YouTube.

After some time, Caroline decides to use a single Web application to host her pictures and video clips. However, she does not want to change the Office Live Workspace application as she finds it very good. Caroline finds a PhotoBucket service. This Web application allows storing pictures and video clips and sharing them with other users of the Web.

Caroline reads reviews on the Web and decides to try the PhotoBucket service. She creates an account on the service. She then configures her Authorization Manager to allow PhotoBucket access her resources hosted on YouTube and Picasa. The PhotoBucket service then acts as a consumer of those resources and retrieves them. Caroline does not have to be actively involved in that process and she does not have to download any of her pictures and videos from YouTube and Picasa and upload them to PhotoBucket by herself.

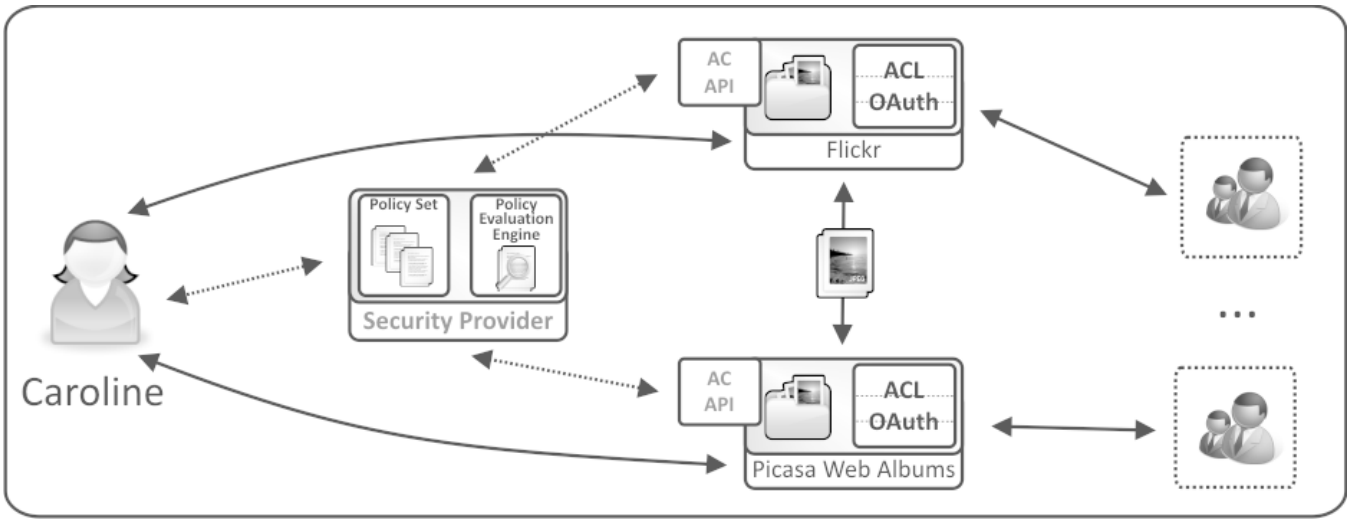
The PhotoBucket service supports a User-Managed Access so Caroline configures it to use her already tested Authorization Manager. By default, all newly uploaded resources to PhotoBucket are made private. However, Caroline logs in to this service and clicks on the **Administration** option. She is then presented with a menu where she can perform various administration tasks. Among all of the options she can choose to apply security to all of her resources. Once she clicks on that option, the PhotoBucket contacts her Authorization Manager so that policies can be applied to resources hosted by this application.

Typically, when Caroline defines access control rules for her resources (i.e. pictures, video clips and documents), she logs in to the application that hosts those resources and clicks on an **Access Control** link next to a resource. She is then redirected to an Authorization Manager of her choice. Under the hood, the application contacts the Authorization Manager with information about the resource and operations supported by this resource. When Caroline is redirected to her Authorization Manager then she sees that it waits to compose an access control policy for this resource. She uses the interface of her AM to specify access control rules that will be later applied to this resource.

In the following case, however, Caroline does not click on the **Access Control** link. Instead, she clicks on a link to apply security for all of her resources. Similarly, she is redirected to her Authorization Manager and sees that a group of resources waits for policies to be specified. Her AM detects that access control policies were previously applied for those resources and informs Caroline about that. What she needs to do at this moment is to confirm that those policies can be reapplied. Once she does that, her resources remain protected in the same way when those resources were hosted by YouTube and Picasa Web Albums.

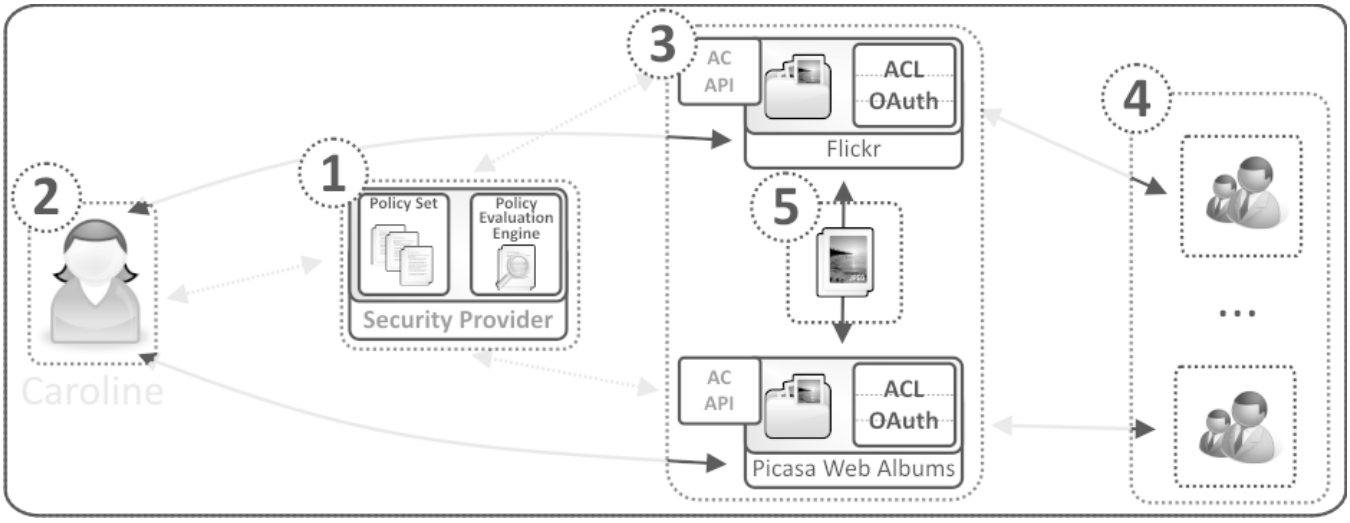
Architecture

The architecture for a User-Managed Access for the provided scenario is depicted below.



A user delegates access control for a set of resources from a Web application to an Authorization Manager. If a user decides to move this set of resources from one application to another then the same set of policies can be reapplied to those resources at the new application. Policies stored and evaluated by an Authorization Manager are therefore application agnostic. A user is able to define access control rules to resources independently of the application that hosts those resources.

View of the actors presented in this scenario with regards to the generic architecture of a User-Managed Access is depicted below. Presented diagram shows an Authorization Manager (1), a User (2), a set of Hosts (3), Requesters (4) and a Resource (5).



An access control policy protects resources independently of Web applications (Hosts). As such, if a resource is moved from one application to another application, the same access control policy can be easily reapplied.

Discussion

The following scenario shows how a user is able to reapply already composed access control policies to resources if those resources are transferred from one Web application to another Web application. Typically, in such situation it would be necessary to define access control policies from scratch or to transform policies from one application to the format used by another application. However, in case of a User-Managed Access where policies are stored in a central location, it is possible to simply apply those policies to the same set of resources that is hosted by a different Web application.

Reapplying an access control policy to a set of resources does not differ much from applying a single policy to a resource hosted by a Web application. Typically, when a user creates a resource on the Web and wants to protect it then a Web application contacts an Authorization Manager so that either a new access control policy can be composed or an already defined policy can be applied. In a situation where a resource is transferred from one application to another, the new application perceives such resource as newly created. Therefore, this new Web application contacts an Authorization Manager as usual. It is an Authorization Manager that detects that a policy for this resource has been previously defined and applied. Therefore, this Authorization Manager proposes to reapply the same policy to protect this particular resource.

An important issue that needs to be resolved in the discussed use case is the possibility of having different operations that are supported by different applications for the same type of resources. An example of that is when one application allows downloading, writing, deleting and shrinking a picture while another application allows downloading, writing, deleting and transforming pictures. When an application contacts an Authorization Manager it sends information about a resource along with operations supported on this resource. An Authorization Manager may detect that a policy has been already specified for such resource and may propose such policy to be reapplied. However, a set of operations described by a new Web application may differ from the set of operations that are defined in a security policy. This can be either a subset of original operations, a superset of those operations or a different set of operations.

If a new Web application supports only a subset of operations that were originally supported by the previous Web application, then rules for those operations that exist in an access control policy are simply removed. In case a new application supports a superset of operations then all rules from an access control policy are retained. New rules for newly supported operations can be easily added to the policy. In case the set of operations differs from the operations as defined in an access control policy, a human intervention may be required to map names of old operations to the names of new operations.

Dimensions

- Scope: This use case touches the notion of Scope is so far as the moved resource is to be assigned the same Scope values.
- Cardinality: This use case involves multiple Hosts and multiple AMs, and as such it may have a high degree of cardinality.
- Nature of access to protected resource: This use case may require the nature of access to be determined. (In the diagram, an API method is indicated.
- Person-to-Self: This use case may be implemented in a manner that involves a Person-to-Self transaction, in the sense that the User that authorizes the move is also the same User that authorizes the new Web Application to accept the existing AM currently used by the User.