distributed_services_scenario

Scenario: Distributed Services (Pending)

Submitted by: Christian Scholz

The story

(this text is just to describe one possible environment in which distributed or mass authorization of services is useful)

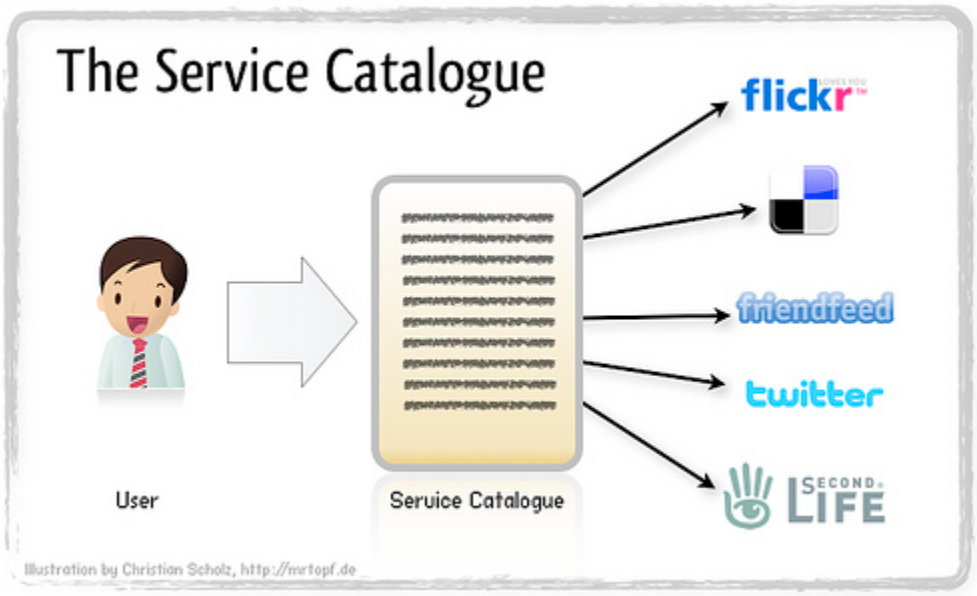
If you look at the social networking scene today one thing is obvious already: There is lot of data online on various services and much of this data is redundant because it is available in various copies which are usually not synced. The main area for this problem is probably profile and friendship/contact information. On each social network or service you register you usually have to enter your profile information again and try to find your contacts.

With the advent of more and more of such social services the amount of redundant data will grow even more and this will lead to a acceptance problem.

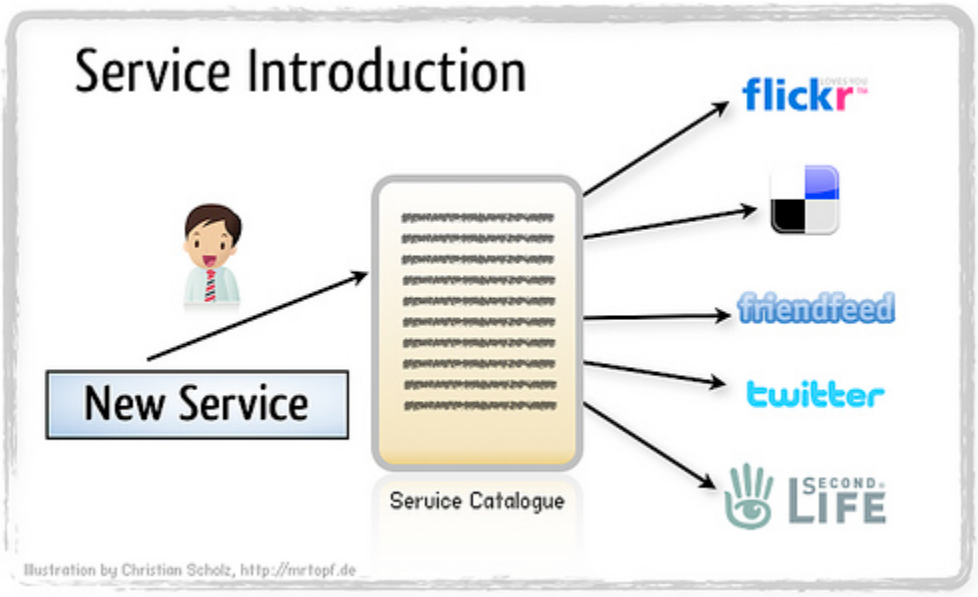
The Service Catalogue idea

It is unlikely that users will centralize all their data in one place. It's more likely that data will be distributed even more. So one problem might already be to manage all the places where data is stored about you or where services can provide functionality on your behalf. One solution to this might be a concept called "Service Catalogue" which came up in discussions in the open web/DiSo/DataPortability communities. The basic idea is to have a list of all these places stored under your control which can be queried by services.

Another point is that for reducing the amount of copies of your data it is necessary to link to your data instead of copying it (or even worse asking the user to type it in again). The Service Catalogue can serve basically as such a link list where each service/type of data is marked up with a location (URI) and type (probably another URI). Obvious things to link to are your profile and contact list but other things make also sense, like photos, videos, blog posts, recommendations, your attention profile, travel information and much more.



Having this catalogue you can easily tell a new service which other services you already use by simply pointing it to the Service Catalogue:



Note though that this Service Discovery is out of scope for the work at UMA and only serves as an example of how to obtain a list of services to authorize later on. Another method apparently is the user typing in various URLs which is not that user friendly though.

The result is in any case a list of services you want to authorize.

Distributed Authorization

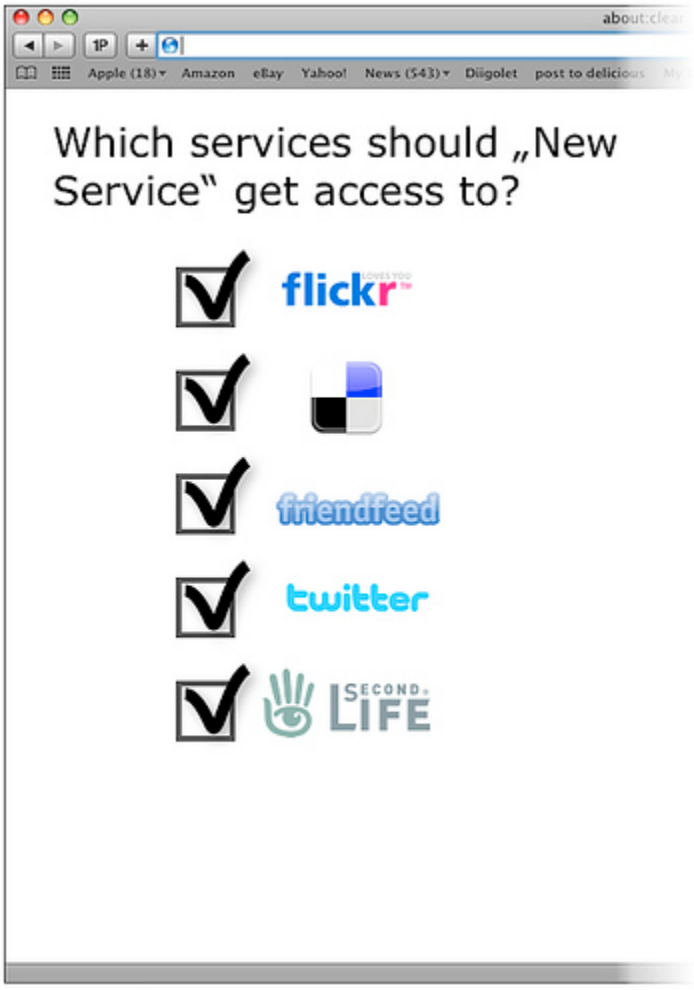
The problem is how you authorize that new service to get access to all the other 3rd party services. OAuth is one possible solution but at least if the default mechanism for retrieving a token is used this means that the user has to be redirected to each of these 3rd party services in order to give consent for the new service to use that data.

Moreover OAuth does not contain a mechanism to define what permissions should be used on the service endpoints. This can be done individually by each service but having a central place for such policy decisions and being able to store policies and share them among services might be beneficial as well.

An example for such profiles would be that you can filter which fields of your profile a PortableContacts endpoint actually gives out to a certain consumer.

For the sake of usability what we want is a single page where you can define the relationships between that new service and all the other services you have access to.

This could look like this:



Additionally a user should be able to quickly revoke tokens again in a central location as well as getting an overview of which services have access to which other services under which policies.

Dimensions

- Scope: This use case involves the Scope dimension, as the new service added to (or through) the Service Catalogue may need to be given the Scoping information pertaining to all the existing services already listed in the Catalog. Furthermore, the syntax and semantics of the scopes (scoping rules) will need to be interoperable between all the the services listed in the Catalogue.
- Resource Discovery: This use case relates to resource discovery in that the Service Catalog acts as a service directory from which external entities can learn about services relevant to the user. (NB. privacy issues).