Information Trust Exchange Governing Association
http://www.itega.org
https://docs.google.com/document/d/10NshZq6UMyWL52WOb7nQFgP6P5r3xZEbiks02rPXcYg/edit

A discussion of a Single Sign On network authentication service featuring unique, network, anonymized common IDs

This is in the ite-TECH-DESIGN folder
File: ite-single-sign-on-network-description-v1-RL-01-19-17

Author: Richard Lerner, Clickshare Service Corp.
Comments in RED from Bill Densmore

REFER TO PAGE SEVEN OF THIS:
http://www.newshare.com/charts/charts.pdf

Here is a first shot at a description of how the SSO authentication/access will work. This does NOT cover the Customer Profile Network component. The main questions I have are:

1) Is this clear? If not, what parts are unclear?

2) Does this meet the needs of the ITEGA demonstration? If not, what else is required?

3) Is there any functionality you would like to add, or see described in more detail?

Thanks,
Rick

Comments in this color from **bill densmore 01-20-17 1:50 p.m.**
Let's refer to this charge for terminology:
http://newshare.com/ite-demo/ite-architecture-DIAGRAM-v2-01-17-17.pdf

17 Jan 2017 - updated for ITEGA

7 May 2012 - initial version

Implementing a Network for NECommon / ITEGA

Overview

The CSNetwork allows customers from multiple sites to authenticate on all sites within the network, using the account on their home site. Only the home Site (Identity Service Provider) ever receives or authenticates a customer's username and password. Each member site can collect and store personalization information for remote accounts, as may be required by the site. The personalization information can come from any of three sources:

1. Directly from the customer via a form submit to the remote site (first-party data)
2. Copied from common data sent from the home server. (exchanged via ITEGA protocols)
3. Copied from the Customer Profile Network (Data/Demographic Aggregator) server. (which only has information supplied from the home base server and only liked to an anonymous key)

The latter two allow the customer to provide information that follows them around the network. If #3 (DDA) can distinguish between network clients and third-party clients, we may be able to forgo #2 and have all personalization information pass through the Customer Profile Network. Otherwise, we need to define a common set of demographics to pass through the CSNetwork server.

This seems like duplicate work to me. (Good discussion to have -- seems to me key is the freshness of the information. If the DDA is constantly updated by the IdSP whenever sharable profile information is changed at the IdSP, then there is no need for the Auth-Logging Service to have any dynamic data on the user. But there may be some very basic information about a user (such as their globally unique anonymous user ID) that needs to be available real time universally to address billing and login issues, but not demographic issues. It strikes me that should be part of what is authenticated and then stored on a session-basis by the auth/logging service.

The particular DDA where with the IdSP is affiliated -- and there may be at scale a pluraility of DDAs for different topical/geographic/use networks -- can hold the detailed demographic data but on one side -- the side which works with IdSP's, it is specific to a unique user ID -- on the other side -- the side which deals with Profile Usage Agents (advertisers) it is anonymized.

The home server (hIdSP) maintains a mapping from its internal userIds to networkUserIds assigned uniquely for each remote server the customer authenticates with. Yes So, if I am registered on HOME with userId 12345, and I attempt to authenticate on REMOTE-1, HOME will assign a new networkUserId, say "NNN-987643", where NNN is the id of the REMOTE-1 server, to send to REMOTE-1 to identify my account. If I then attempt to authenticate on REMOTE-2, HOME will assign a different networkUserId to

send to REMOTE-2. The next time I authenticate on either REMOTE-1 or REMOTE-2, HOME will send the corresponding networkUserId it generated for my first authentication on that remote server. **This allows the home server to unlink a customer from one or all networkUserIds, should there be a desire to do so, and blocks attempts by REMOTE-1 and REMOTE-2 to combine data about me.** Yes, this seems ideal.

The first remote authentication request follows the following sequence:

1. The customer clicks on a link for an article on a remote, in-network, server (e.g., from a LifeSteam email - link actually goes first to Li).

2. The remote content server redirects to its authentication server to authorize the request.

3. The remote authentication server puts up its standard login page, with a new "Network Login" button.

4. The Network Login button invokes a Clickshare controller on the remote authentication server to initiate a remote authentication (networkAuthRedirect.do).

5. The networkAuthRedirect controller saves information about the customer's request (e.g., which article they were asking for) and sends a network authentication request to the CSNetwork server via a browser redirect to its networkAuthStart controller.

6. The CSNetwork server looks for a cookie it drops with the browser's home server selection. If not found, the CSNetwork server displays a page asking the customer to indicate their home server.

7. The CSNetwork server redirects the browser to the home server network authentication controller (networkAuth.do).

8. The home server authenticates the customer in its normal manner (which may or may not show a login page) records any information it needs and either finds or generates the appropriate networkUserId for the request. It computes a networkGroupId, based on the customer's current effectiveGroupId. Finally, it redirects back to the CSNetwork server's networkLoginStart controller.

9. The CSNetwork server drops its home server cookie and redirects to the remote authentication server's networkLogin controller.

10. The remote authentication server locates the original request information, creates a local account, if necessary, maps the networkGroupId to a local effectiveGroupId and continues with its standard authentication process. If the effectiveGroupId is sufficient to authorize the original content request, the remote authentication server redirects to its content server to display the content.

Once the customer has authenticated on the remote server, the remote authentication server can choose to remember this authentication for some period of time, granting subsequent access to content without consulting the CSNetwork server.

If the customer has already logged into their home server and has previously authenticated on any remote server so that the CSNetwork server has dropped its homeSite cookie, the redirects for authentication will be transparent to the customer. From the customer's perspective, they clicked on an article link in the email and the article appears. In the worst case scenario, such as a browser whose cookies have been cleared, the customer would see the following sequence of pages once they clicked on the article link in the email:

1. The remote authentication server's login page with the "Network Login" button.
2. The CSNetwork server's "Select Home Site" page with a "Submit" button.
3. The customer's home server's login page, where they submit their username and password.
4. The remote authentication server's customer information page, if the site wants to ask the customer for more information.
5. The requested article.

This all sounds right but would like to see a chart depicting entities and sequence to be sure I get it.

The NetworkGroupId

The CSNetwork defines a collection of common access properties that servers within the network use to determine which customers have access to which content. Yes, exactly.

The home servers are responsible for mapping from their own access groupIds to the network groupIds. The remote authentication servers are responsible for mapping network groupIds to their local access groupIds.

The system initially supports the following group Ids, which can be combined as appropriate:

Bit	GroupId	Name	Description
-	0	Anonymous Customer	Customer accessing the site without having logged in (e.g., using a pre-reg meter)
0	1	Group Account Customer	Customer logged in using a group locked account (e.g., IP or accessKey access)
1	2	Registered Customer	Customer logged into an individual non-locked account
2	4	Print Subscriber	Subscriber to print edition
3	8	Digital Subscriber	Subscriber to replica digital edition
3	8	Web Subscriber	Subscriber to online content
4	16	Data Subscriber	Subscriber to special content

10	1024	Comp Subscriber	Subscription is complementary
11	2048	Controller Subscriber	Subscription is granted as a controlled (free) subscriber
12	4096	Paid Subscriber	Subscription requires payment
13	8192	Trial Subscriber	Temporary trial subscription
14	16384	Site Subscriber	Subscription as part of a group (e.g, corporate, university, or library access)

These make sense but plan to reflect on whether there are any other groups we should support out of the box.

Questions

1. Should all of the member sites have access to the single uniqueId of a customer?
   I think they should because their use of data can be constrained by contract. Or, like third party profile clients, should they receive a temporary userId? My preference is to use a fixed networkUserId for identifying remote customers, rather than the internal userId, but that the networkUserId would be long-lived for each remote server. The networkUserId would be different for each report server. This allows the home server to unlink a customer from one or all networkUserIds, should there be a desire to do so.

2. Should networkGroupId be a bitmap, like our effectiveGroupId values? Or, should it be text based? Also, should it allow comp,paid, etc. to be modifiers on Print, Digital, etc.?
   I don't understand the significance of this issue.