Identity Federation System (idFS)

Version 1.2

April 2004

Introduction

The Passive Requestor Profile (PRP) is part of the WS-Federation specification. It describes, how Web services (security token services/STS) specialized on issuing security-relevant claims (security tokens/ST) can cooperate in federated scenarios, where a passive requestor (e.g. a user with a Web browser) is involved. The definition covers the way how the requestor is redirected between the corresponding services and how the necessary data (like the security token) is conveyed.

The idFS based on C#/.NET consists of a collection of HTTP Modules for resources, STS and identity providers (IP; a special STS used for authentication). These modules allow the construction of corresponding federation architectures with minimal development effort.

idFS also comes with a demo of two sample Web sites requiring users to sign-in. For that purpose, they both use the same resource STS that sends the user to one of two requestor IPs with login forms. The demo features single sign-out (SSO) and the dynamic allocation of identity providers according to requestor characteristics (e.g. IP-addresses).

Single Sign-Out Scenario

In this scenario, a user trying to access any resource of the demo always gets redirected to the same identity provider. Once signed-in, she is not required to re-enter her authentication data, even when accessing another site of a different trust realm.

When she performs a sign-out, all sites are automatically informed, ending all of her sessions at once. Figure 1 gives an overview of the interaction occurring between the three HTTP Modules and the login form belonging to the IP. Arrows with boxes represent HTTP redirects, i.e. the browser is redirected to another URL and the PRP parameters in the boxes are passed as POST- or GET-variables.

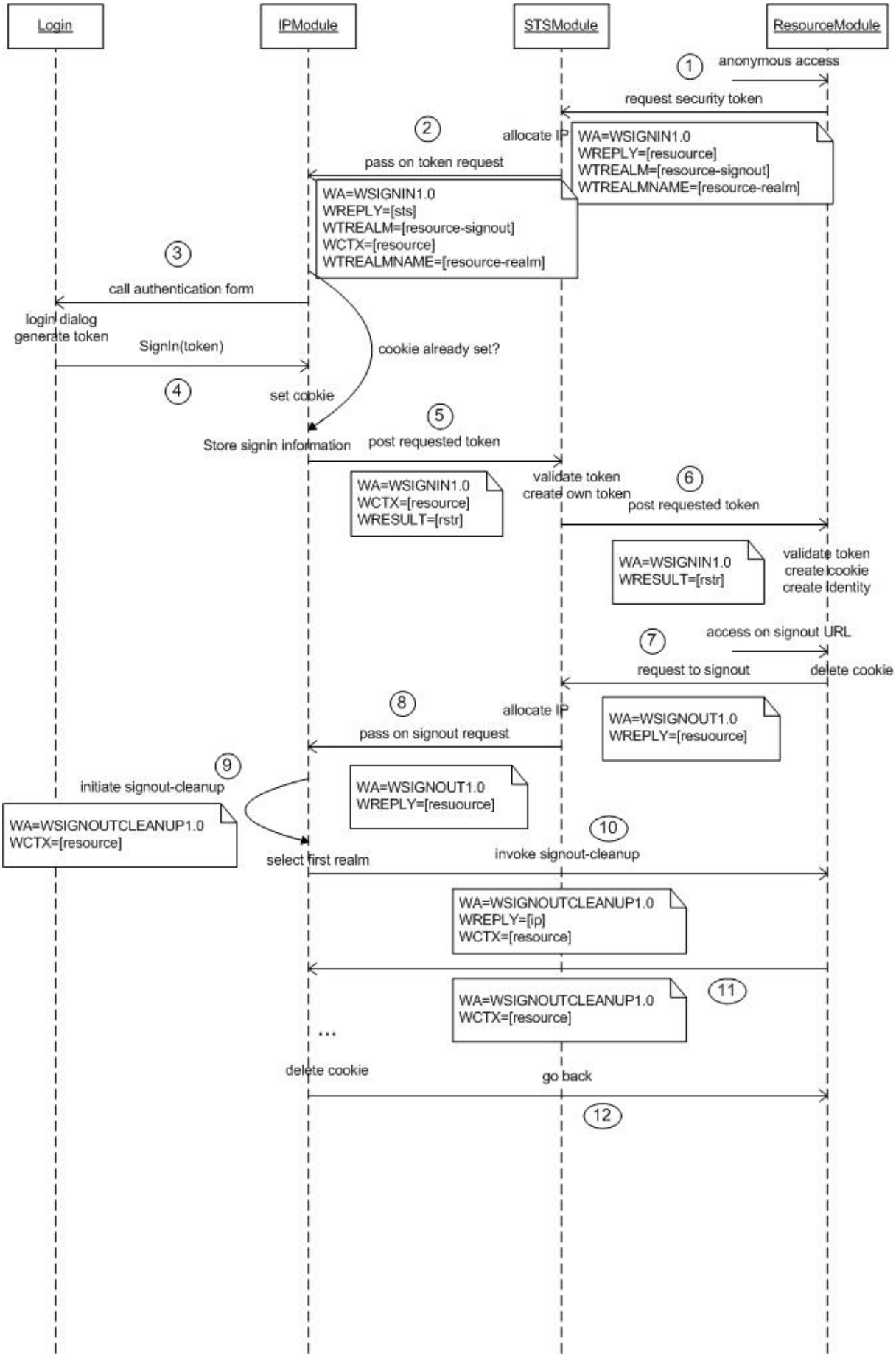

Figure 1: Interaction between services during sign-in and sign-out

Anonymous accesses are automatically redirected to the STS responsible for providing a security token that authorizes the use of a resource (1). Figure 2 depicts the output of the module in demonstration mode. Normally, the redirection occurs transparent to the Web user.

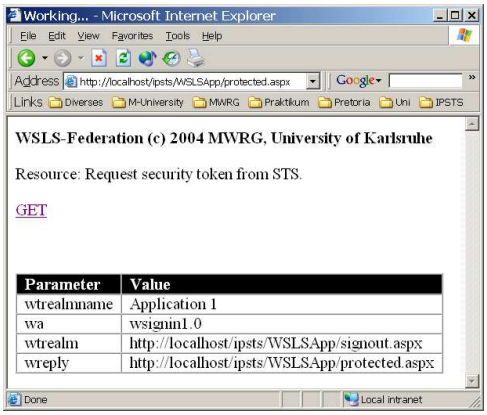

Figure 2: Demo mode output during sign-in procedure

The STS needs to determine the identity of the requestor first. Therefore, the STS allocates the IP responsible for the user and performs another redirect, requesting a security token about her identity (2). The IP checks if the user is already signed in, in which case it directly returns the security token issued earlier without any interaction. Otherwise, the login form displayed in Figure 3 is invoked (3).

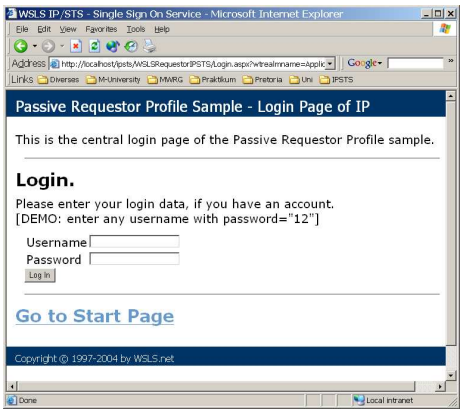

Figure 3: Login form of IP

After the user has supplied her credentials, a security token is generated and passed on to the module with a simple method call (4). The IP creates some persistent data to remember the security realm and the issued token, sends a cookie to the browser for future identification and redirects back to the calling resource STS (5).

The redirect transports the token within a SOAP envelope signed with an X.509 certificate (a.k.a. Request Security Token Response, RSTS). The receiving STS can verify the origin of that message, as it disposes of a list of public keys of trusted services. It generates its own token, which is passed on to the resource in a similar fashion (6). Now the user is authorized and given a cookie, allowing her to access all pages of the site without further consultation of the STS.

After having worked with the site and possibly others, the user decides to sign-out and calls a certain configured logout URL. This task is delegated to the STS (7) and from there to the IP (8), where a sign-out-clean-up-process is initiated (9). The IP knows the security realms at which the user is currently signed-in. The browser is redirected to the corresponding sign-out-URLs of all these sites, causing her specific

sessions to end immediately (10,11). Finally, she is also singed-out at her IP and sent back to where the sign-out procedure was initiated (12).

Multiple IP Scenario

Another scenario realized within the same demo concerns the operation of two IPs, each one delegated to a different group of administered user identities. Figure 4 describes the general setup.

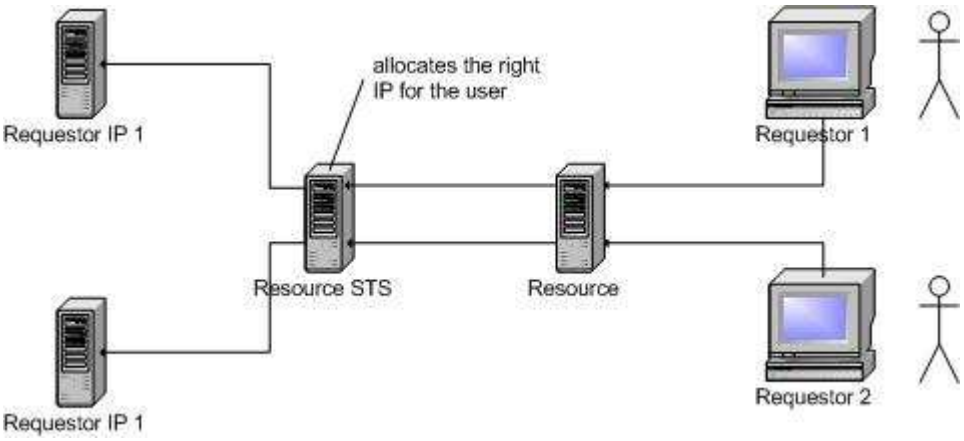

Figure 4: Scenario with two different identity providers

As demonstrated in the previous section, anonymous users get redirected to the resource STS. In order to determine the correct IP, a set of configuration rules, as can be seen in Figure 5, is compared to the user's incoming HTTP request.

```
...
<STSAllocationRules>
  <add key="IP=*.*.*.*" value="http://default-ip.company-a.com/Login.aspx" />
  <add key="IP=172.20.40.*" value="http://sales-ip.company-a.com/Login.aspx" />
  <add key="IP=172.20.41.*" value="http://management-ip.company-a.com/Login.aspx" />
</STSAllocationRules>
...
```

Figure 5: Example of STS configuration rules

In this case, the IP-address is used as the key condition. Browsers running on machines within the IP-address-range of the sales department are automatically forwarded to one IP (in this example the sales-ip), whereas management machines are directed to another. Also, there is a third IP for requestors not belonging to either one of the groups. This concept allows the decentralization of the authentication process, eliminating the need for a single central authority knowing all users.