WS-Federation

Web Services Federation

- WS-Federation (Web Services Federation) is a protocol used for enabling secure, trusted, and interoperable identity and authentication mechanisms in web services environments.

- Web Services Federation (WS-Federation or WS-Fed) is part of the larger WS-Security framework and an extension to the functionality of WS-Trust which are in turn part of WS policy.

- The features of WS-Federation can be used directly by SOAP(Simple Object Access Protocol) applications and web services.

Web Services Federation: Goal

- The main goal of WS-Federation is to facilitate single sign-on (SSO) scenarios across different web applications or services that may be owned and managed by different organizations.

- It allows users to authenticate once with an identity provider (IDP) and then access multiple web services or applications without having to re-enter their credentials each time.

Web Services Policy

- WS-Policy is an interoperability standard that is used to describe and communicate the policies of a web service so that service providers can export policy requirements in a standard format.

- Clients can combine the service provider requirements with their own capabilities to establish the policies required for a specific interaction.

- Service provider shares the policy configuration in Web Services Description Language (WSDL), in WSDL that is obtained by a client by
  - using an HTTP GET request,
  - using the Web Services Metadata Exchange (WS-MetadataExchange) protocol.
- The WSDL is in the standard WS-PolicyAttachments format.

WS Security policy and WS Trust

- WS-Security Policy: It is a security add-on to the WS-Policy. It defines a set of security policy assertions and a framework for allowing web services to express their constraints and requirements.

- WS-Trust : It builds on the WS-Security base by
  - Providing additional mechanisms for working with security tokens and defines the communication with a Security Token Service (STS).
  - WS-Trust clients can make different types of calls for the exchange of security tokens.

Federation Metadata

- Federation metadata describes settings and information about how a service is used within a federation.

- Metadata Endpoint reference (MEPR): It allows the requestors to obtain all requirement of metadata about a service like metadata, communication policies, WSDL etc.

- Communication policy and federation metadata can be embedded within WSDL.

Framework

- The WS-Federation framework builds on three main specifications.

- This standard provides for the following to Simple Object Access Protocol (SOAP) messages:
  - Integrity (XML Digital Signature with SOAP)
  - Confidentiality (XML Encryption with SOAP)
  - Transmitting identity tokens actors (Username Tokens, SAML2, Binary Security Tokens, etc)

Request Profiles

- **Passive request profiles** use this to create a base for using WS-Trust for further communication.
  - A passive requestor is an HTTP browser capable of broadly supported HTTP (e.g. HTTP/1.1).

- **Active request profiles** are requestor directly deal with WS-Trust and WS-Security.
  - An active requestor is an SOAP enabled application (can be a web browser) that is capable of issuing Web services messages such as those described in WS Security and WS-Trust.

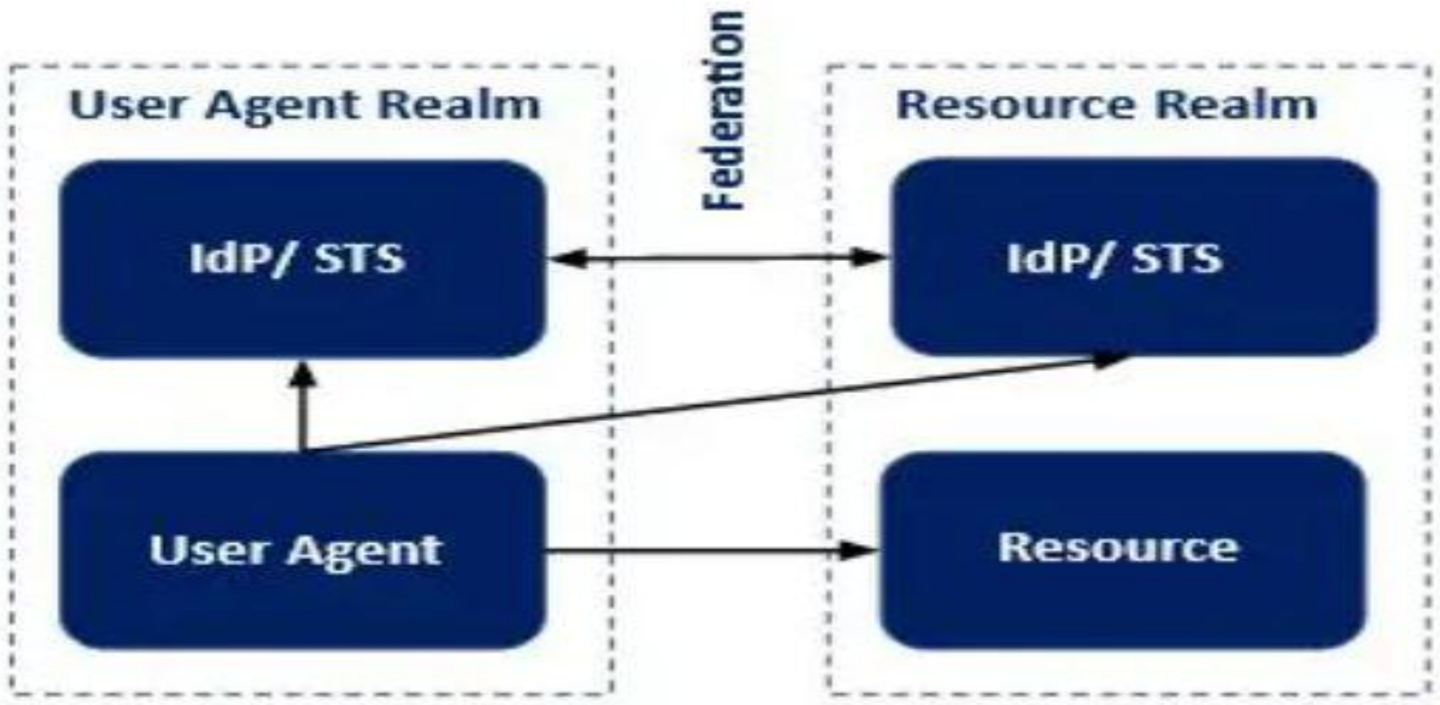
Terminology

- **Realm or Domain** - A *realm* or *domain* represents a single unit of security administration or trust.
- **Federation -** A federation is a collection of realms that have established trust. The level of trust may vary, but typically includes authentication and may include authorization.
- **Security Token Service (STS)** - A *Security Token Service* is a Web service that provides issuance and management of security tokens .
  - Any Web service can, itself, be an STS simply by supporting the WS-Trust specification.
- **Identity Provider (IP)** – Entity that acts as an authentication service to end requestors (an extension of a basic STS).
  - User agent obtains an identity security token from its IP and then presents/proves this to the STS for the desired resource.

Scenario

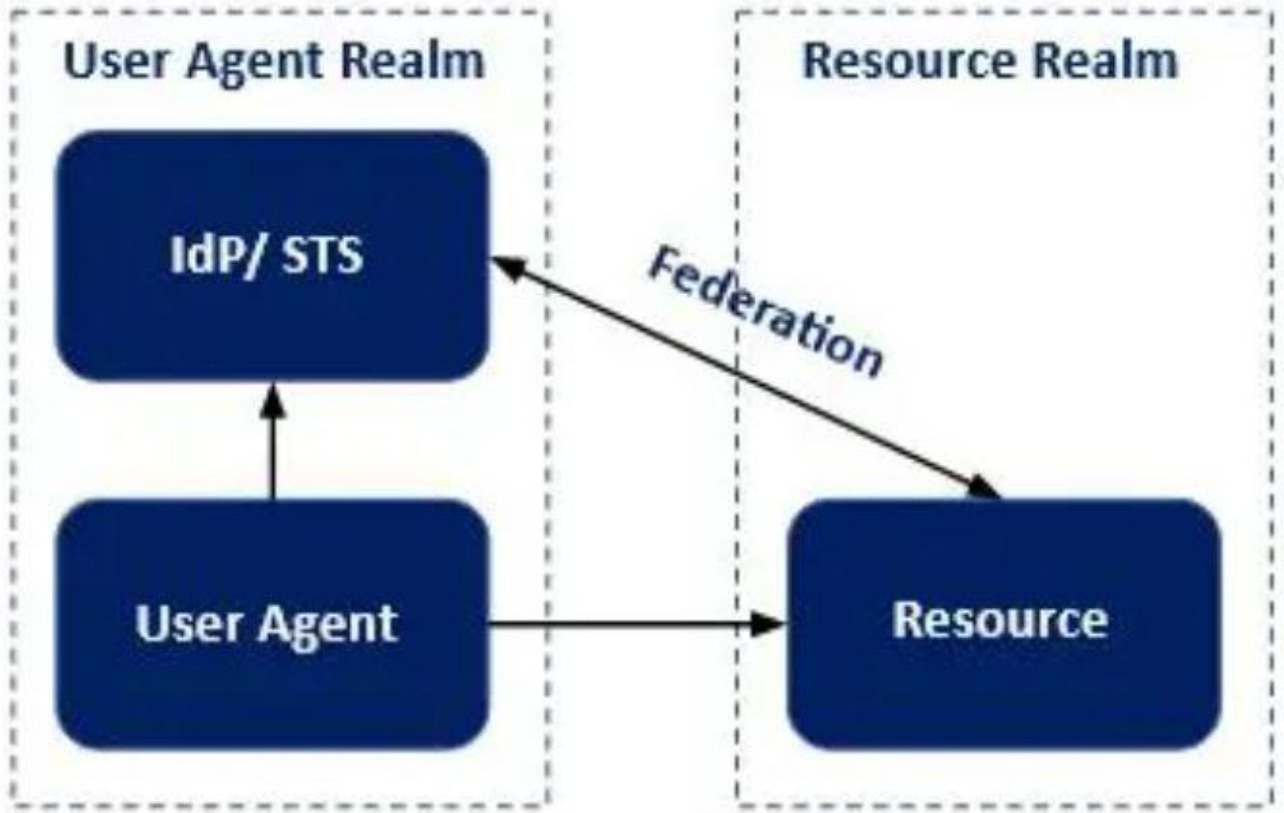
- According to the number of IdPs on either side of the communication, there can be two scenarios:
  - Both User Agent and Resource use an IdP.
  - Only User Agent uses an IdP

User Agent and Resource use IDP



User Agent and Resource use IDP

Only User Agent uses an IdP



Only User Agent uses IDP

WS-Federation Model

1. Trust and Security Token Issuance
2. Identity Providers
3. Pseudonyms and Attributes

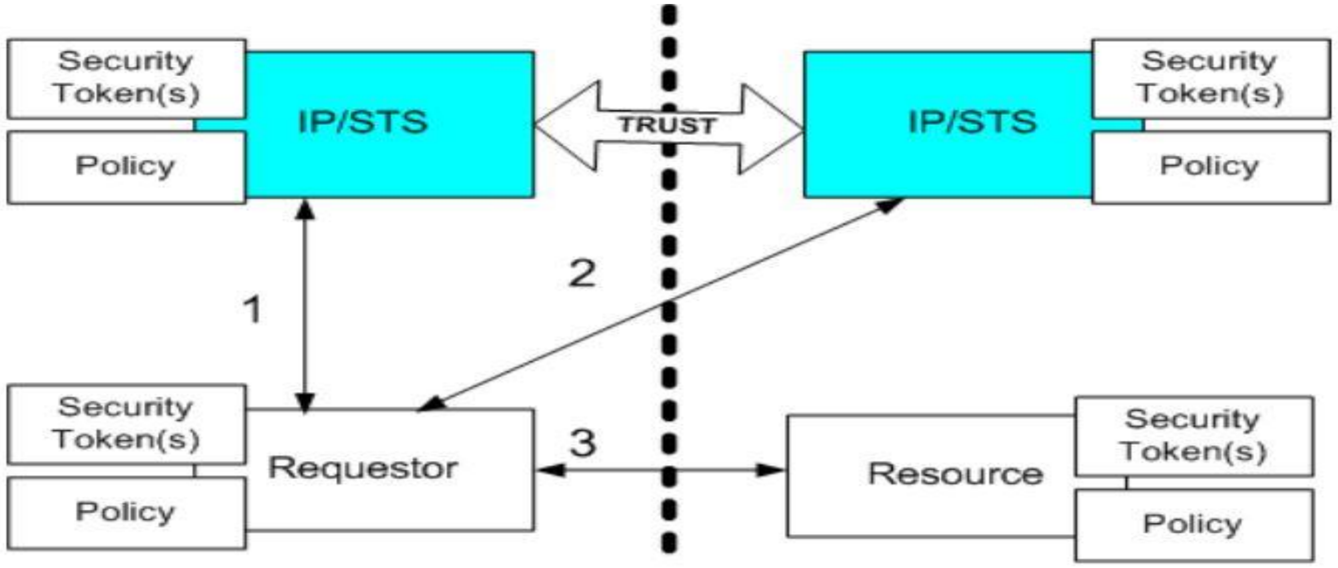
Different Trust Scenario in W

- Direct Trust
  - 1. Basic STS
  - 2. Alternate STS
- Indirect Trust
- Multiple Trust Domains
- Delegations

Direct Trust: Basic STS

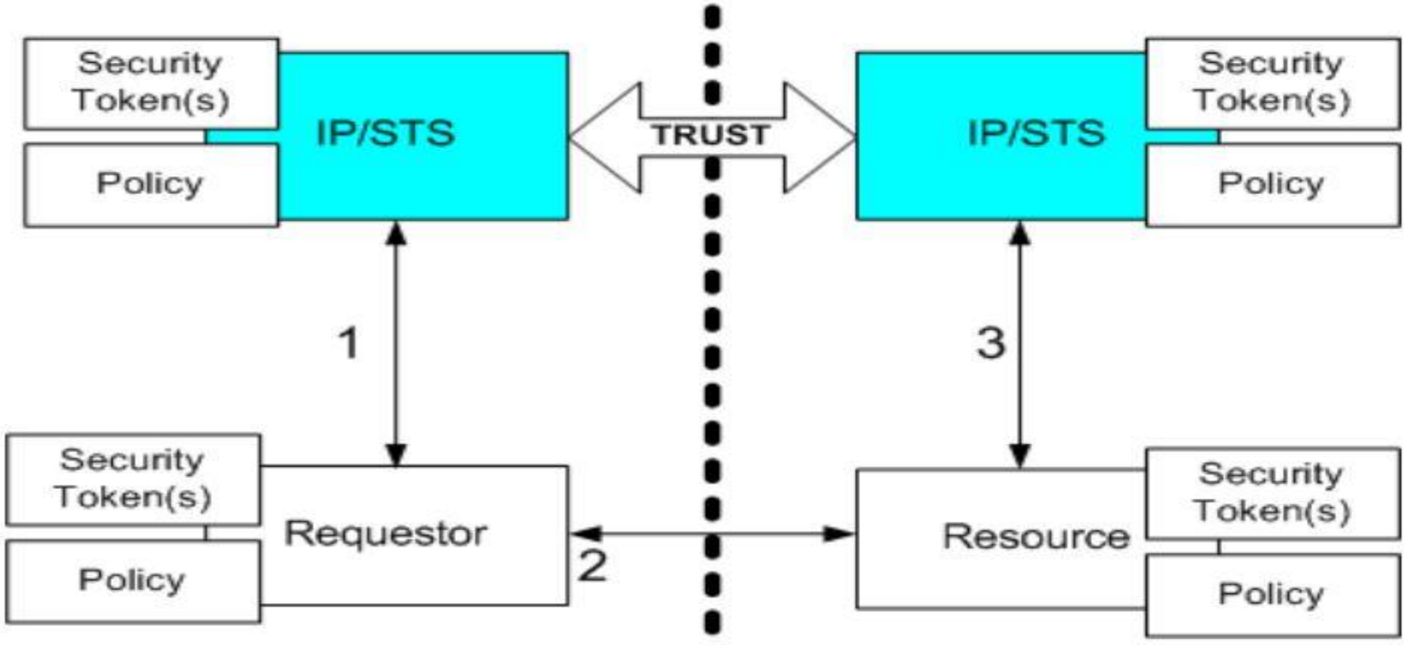
- Here security tokens (1) from the requestors trust realm are used to acquire security tokens from the resources trust realm (2) in order to access the resource/service (3).
- A token from one STS is exchanged for another at a second STS or possibly stamped or cross-certified by a second STS.

.



Direct Trust: Alternate STS

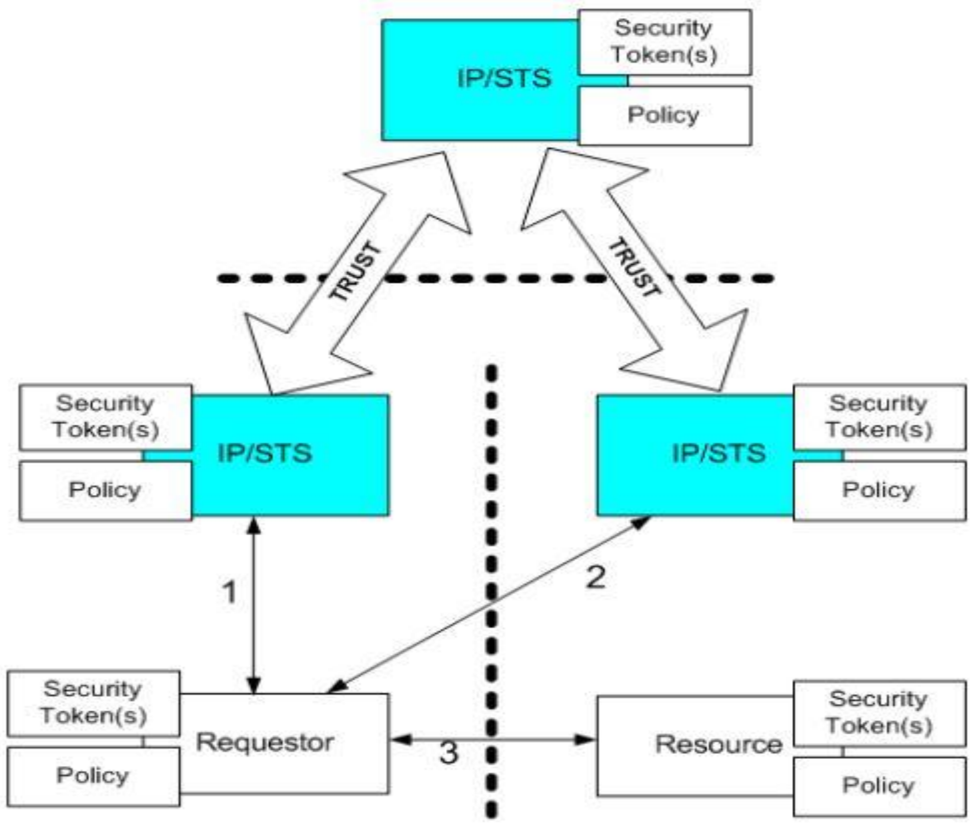
- In this scenario, the resource provider (3) uses its security token service to understand and validate the security token(s) received from the requester (1,2).
- The validity information is returned as a security token (since it includes authentication and/or authorization data).



•

Indirect Trust

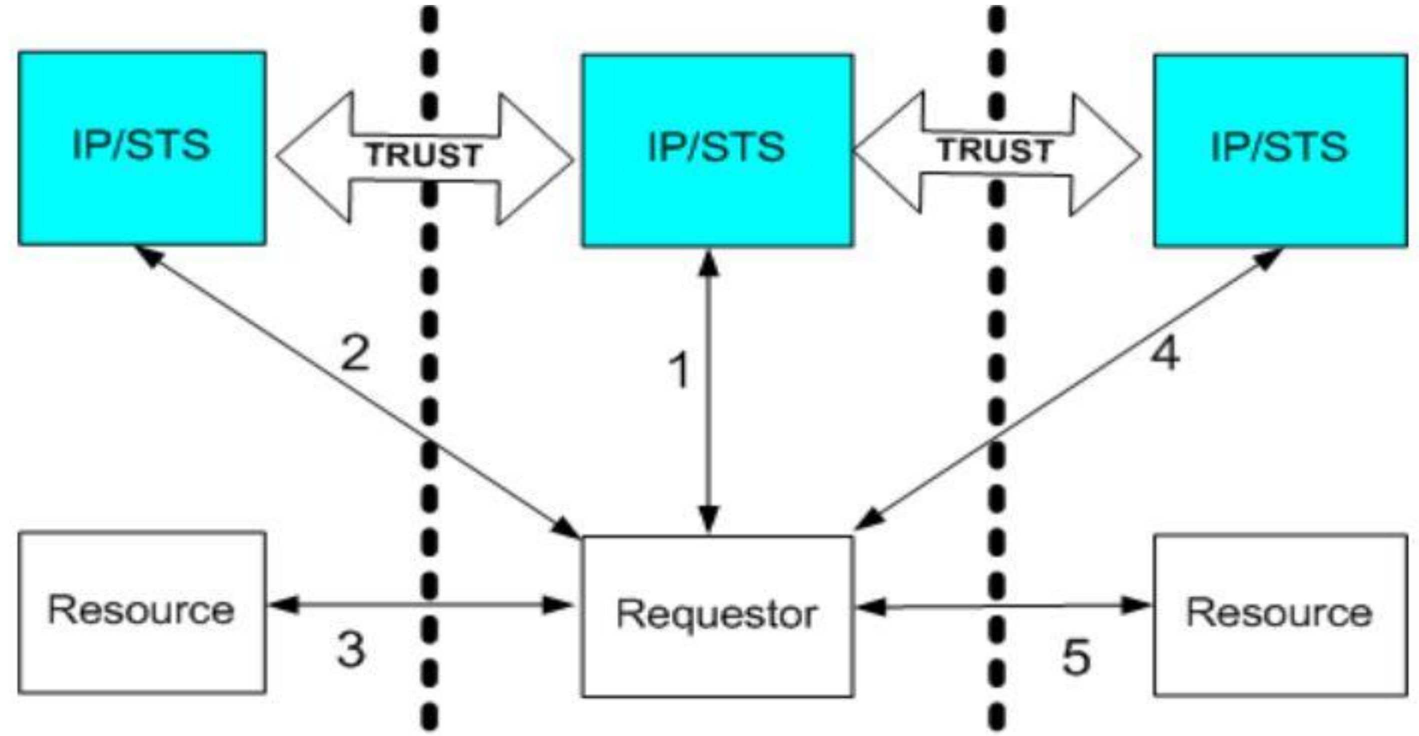
- There may not be a direct trust relationship between token services, but an indirect trust relationship that relies on a third-party to establish and confirm trust.



.

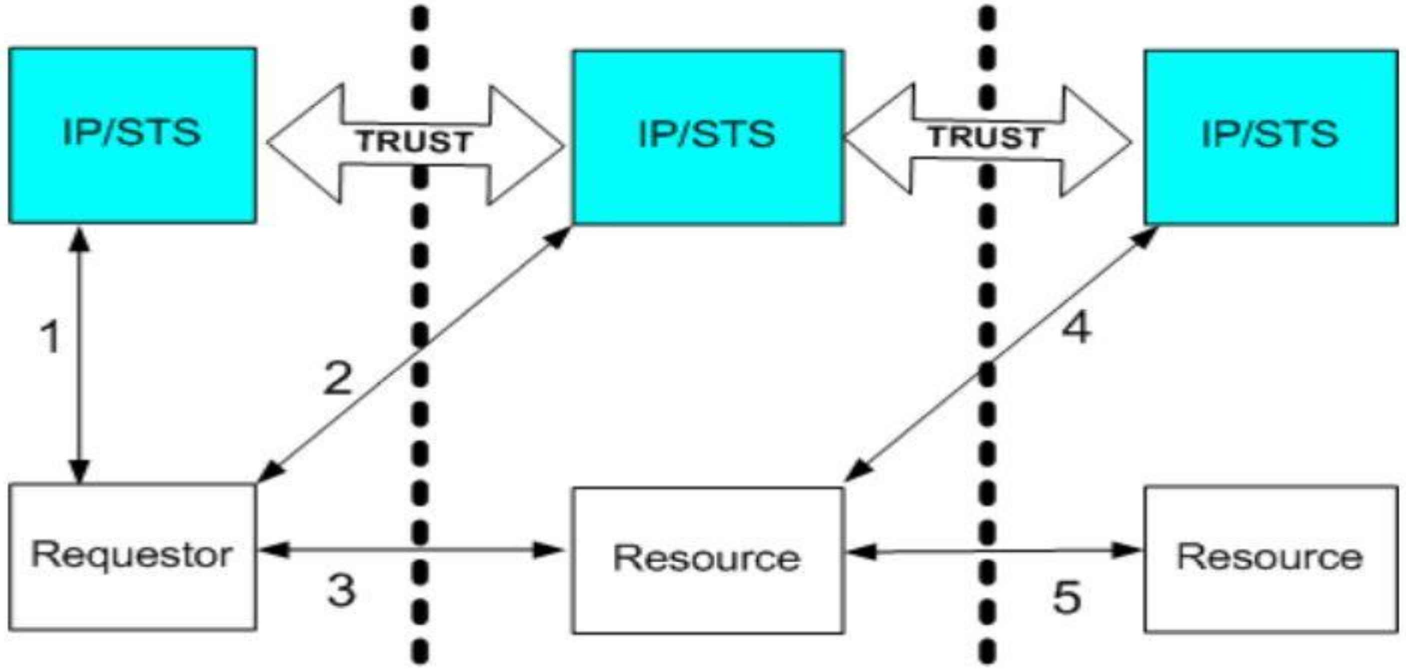
Multiple Trust Domains

- In practice, a requestor is likely to interact with multiple resources/services which are part of multiple trust realms as illustrated in the figure below
- .



Delegation Trust

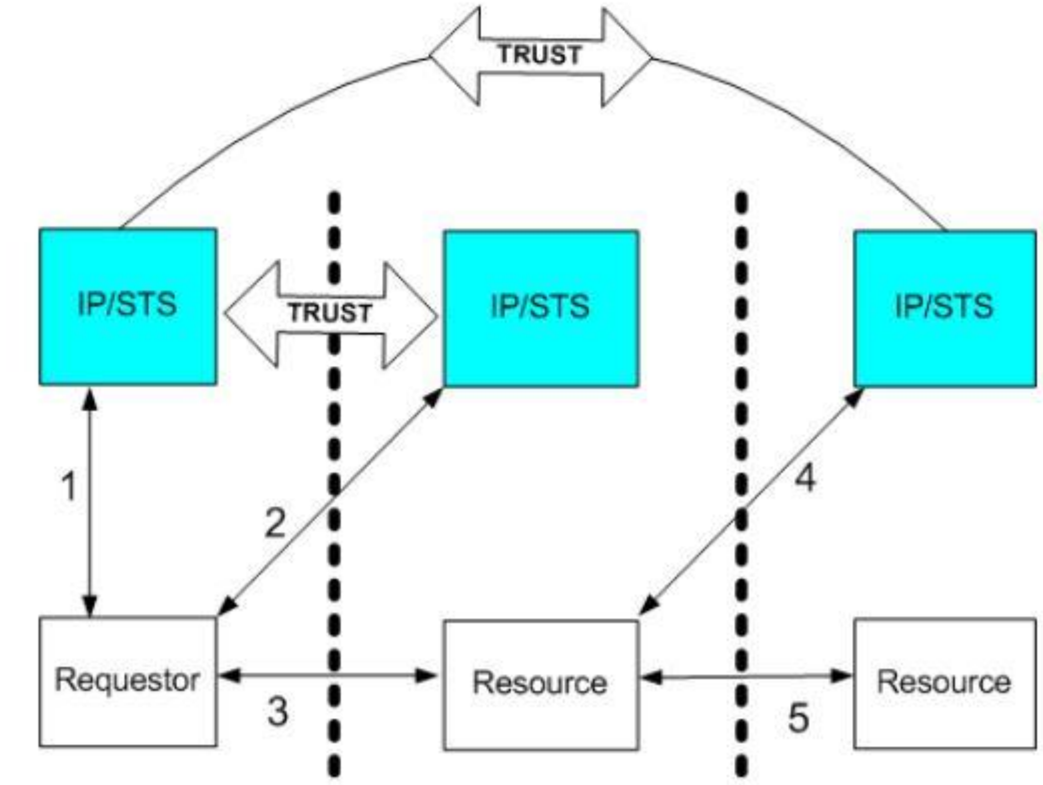
- In response to a request a resource/service may need to access other resources/service on behalf of the requestor.
- In such cases the (2)requestor provides security tokens to allow/indicate proof of delegation.
- .



Delegation Trust

- The security token service for the final resource may only have a trust relationship with the token service from the original requestor (illustrated below), as opposed to the figure above where the trust doesn't exist with the original requestor's STS.
- Requestor (1) provide security tokens to allow/indicate delegation proof.

.



Identity Providers

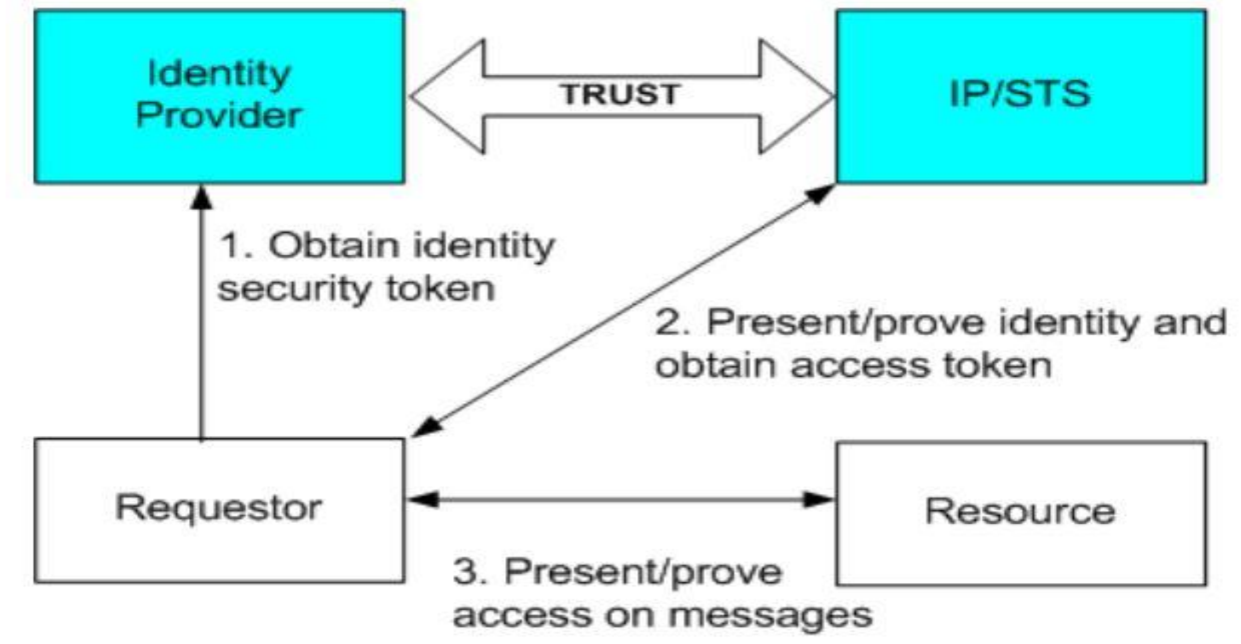
- A security token service (STS) is a generic service that issues/exchanges **security tokens** using a common model and set of messages.
- There are different types of security token services which provide different types of functions.
  - For example, an STS might simply verify credentials for entrance to a realm or evaluate the trust of supplied security tokens.

- Another function of a security token service is to provide identities – an Identity Provider (IP).
  - This is a special type of security token service that performs peer entity authentication and can make identity claims in issued security tokens.

Identity Providers

- A requestor (1) obtains an **identity security token** from its IP and then presents/proves this to the STS for the desired resource.
- If successful (2), and if trust exists and authorization is approved, the STS returns an **access token** to the requestor.
- The requestor (3) then uses the access token on requests to the resource or Web service.

.


Identity Provider
TRUST
IP/STS
1. Obtain identity security token
2. Present/prove identity and obtain access token
Requestor
Resource
3. Present/prove access on messages

Attributes services

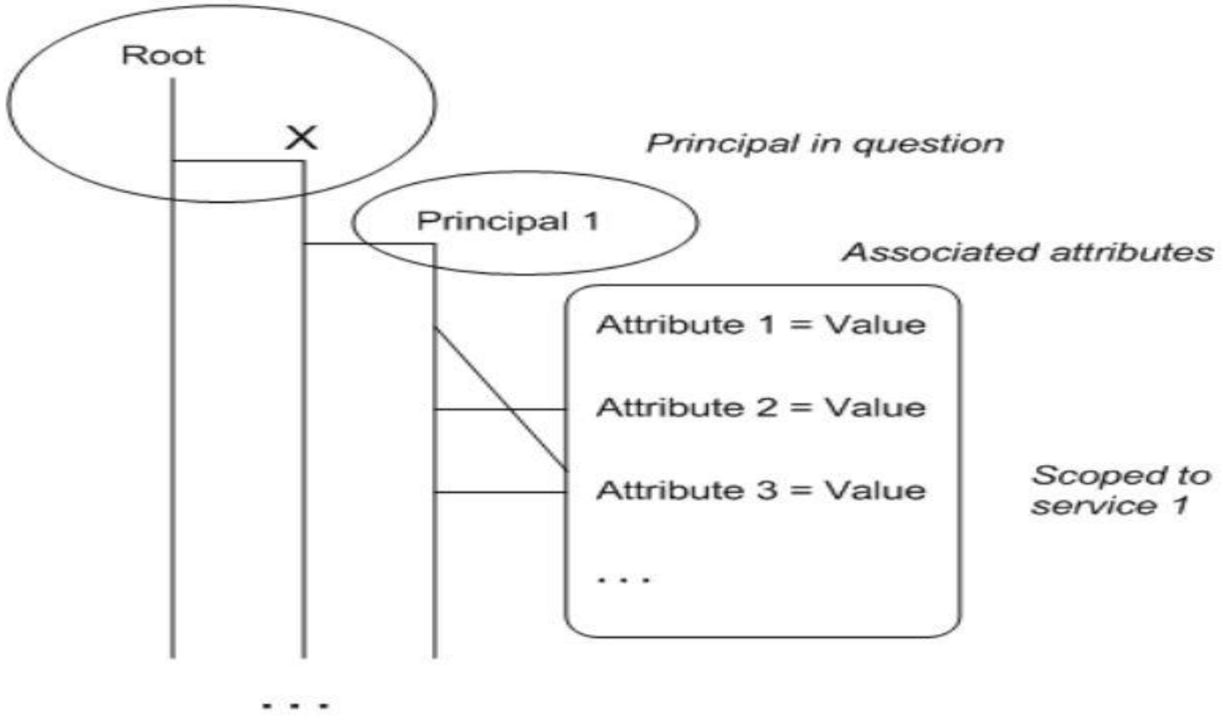
- When requestors interact with resources in different trust realms (or different parts of a federation), there is often a need to **know** something about the requestor in order to personalize the experience.

- A service, known as an **attribute service** may be available within a realm or federation and such a service can be used to obtain authorized information about a principal.

- This allows the sharing of data of a principal (users or resource) between authorized entities.

Attributes services

- An attribute service MAY leverage existing repositories and may provide some level of organization or context.
- Principals represent any kind of resource, not just people.



.

Pseudonyms Service

- To facilitate single sign-on where multiple identities need to be automatically mapped and the privacy of the identity need to maintained, a pseudonym service is used.

- A pseudonym service allows a principal to have different aliases at different resources/services or in different realms.

- Principals can also have also optionally have pseudonyms change per-service or per-login.

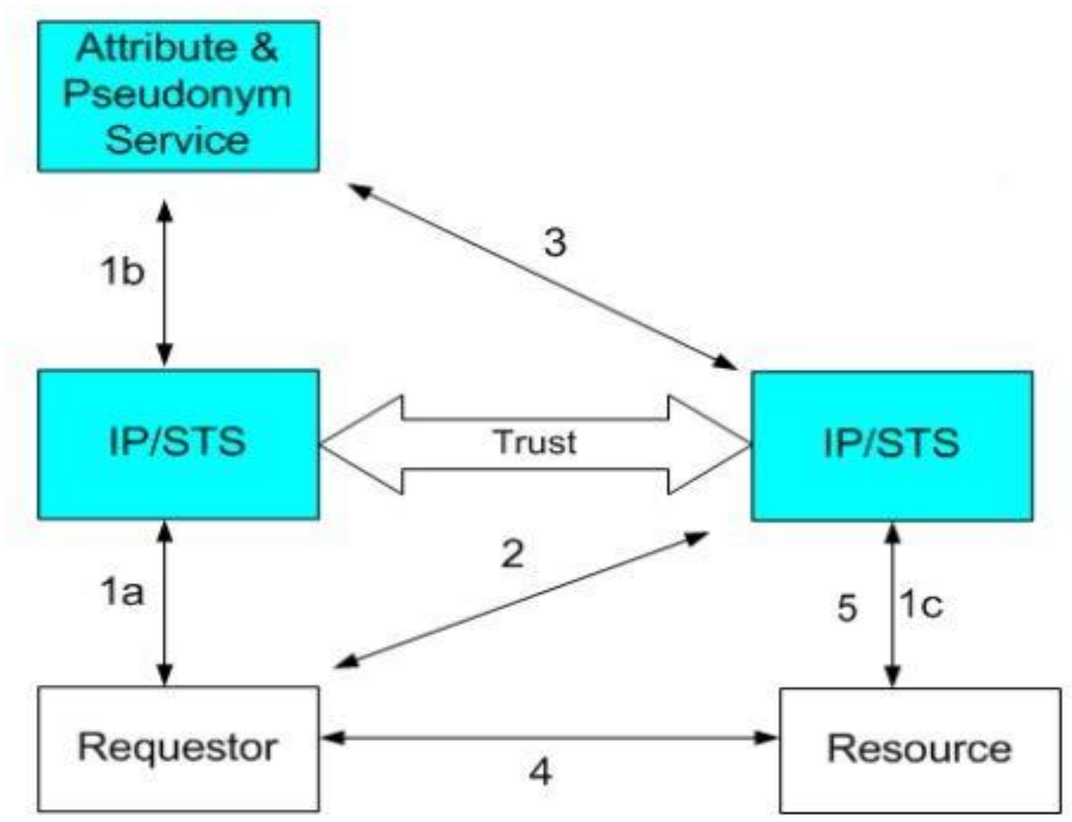
Pseudonyms and Attributes

- IdP/STS use Attribute and Pseudonym services which provide specific attributes needed for authentication(attribute service) or different aliases(pseudonym service) associated with the user agent .

- Many a time they are combined with IdP/STS.

- There can be scenarios where the Attribute and Pseudonym services are separated from the IdP/STS service where an extra step of getting the aliases or attributes from a different system is added.

Pseudonyms and Attributes



Pseudonym and Attribute Working

- Initially, a requestor has knowledge of the policies of a resource, including its IP/STS.
- The requestor obtains its identity token from its IP/STS (1a) and communicates with the resource's IP/STS (2).
- The resource IP/STS has the registered pseudonym from the requestor's pseudonym service (3).
- The requestor accesses the resource using the pseudonym token (4).
- The resource can obtain information (5) from the requestor's attribute service if authorized based on its identity token (1c).

- Alternatively requestor's IP/STS may automatically obtain pseudonym credentials for the resource (1b) if they are available. In such cases, steps 2 and 3 are omitted.

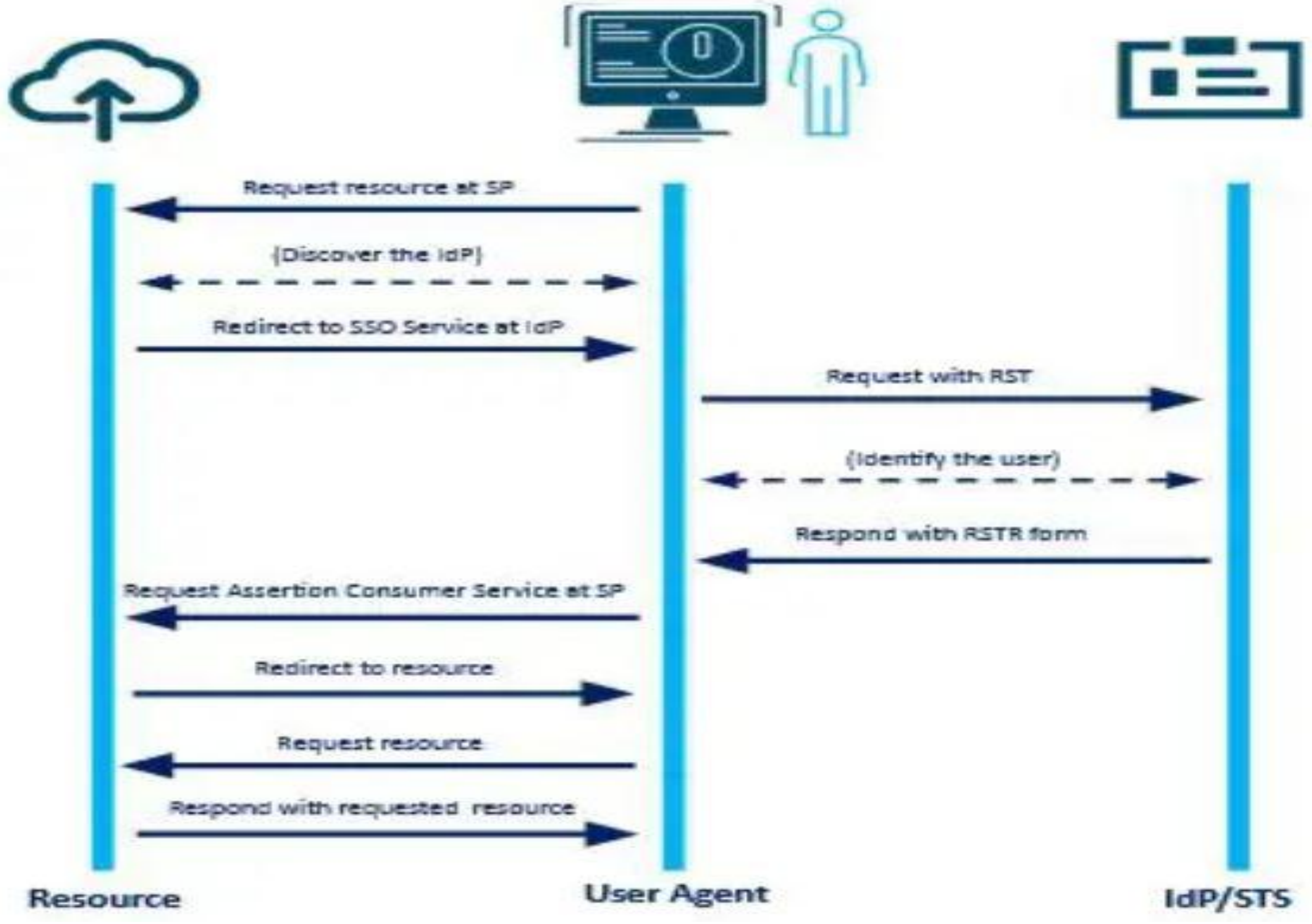
Working of WS-Fed

The working is very similar to that of SAML and only the parameters used for building the trust and tokens are different .

Steps:

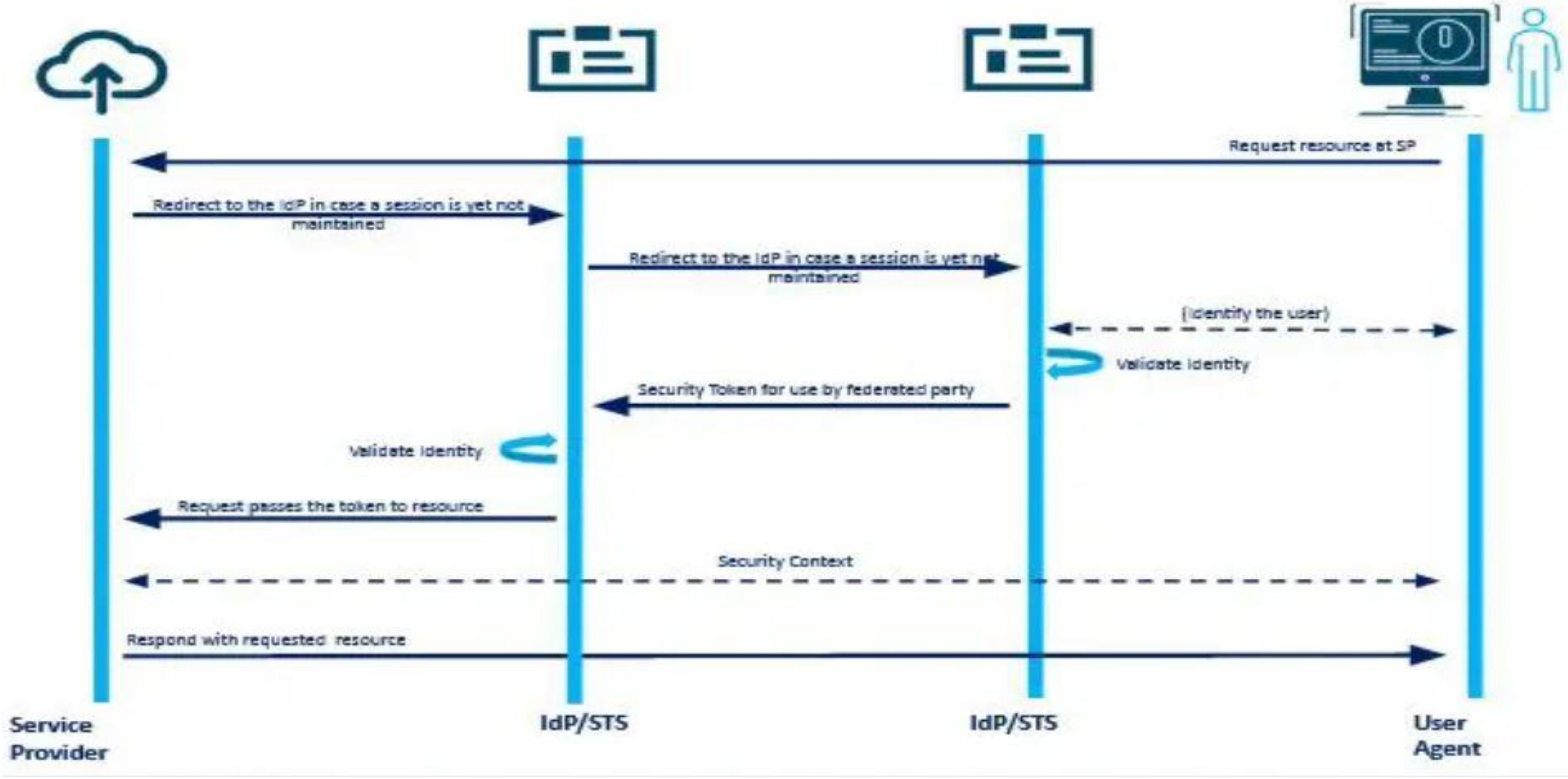
1. User Request the target resource (SP).
2. SP Determines the IdP and redirects to the SSO Service at the IdP containing the RST(Request Security Token).
3. For validating the user agent, IdP parses the RST.
4. The IdP then generates a Request Security Token Response (RSTR) and sends it back to the resource.
5. The assertion consumer processes the response and redirects the user agent to the resource (login done) at SP.
6. Requesting the specific target resource again (specific page or document on the resource)
7. Response of the resource at SP for the request from the user agent.

Only User IdP/STS



Both User and Resource IdP/STS

- If both the resource and user agent are using IdP/STS, then the sequence of events will be different as depicted in the figure below:


Request resource at SP
Redirect to the IdP in case a session is yet not maintained
Redirect to the IdP in case a session is yet not maintained
(Identify the user)
Validate Identity
Security Token for use by federated party
Validate Identity
Request passes the token to resource
Security Context
Respond with requested resource
Service Provider
IdP/STS
IdP/STS
User Agent

Federated Sign-Out

- The purpose of a federated sign-out is to clean up any cached state and security tokens that may exist within the federation.

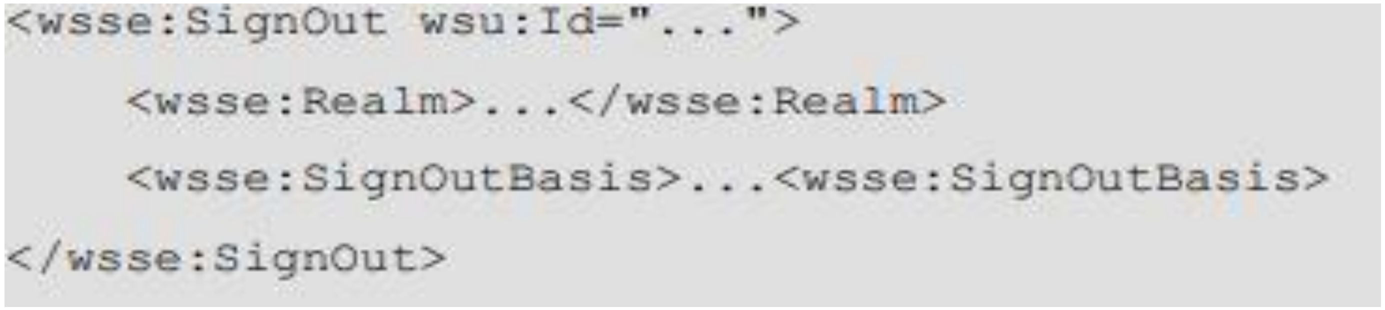
- That is, sign-out notification serves as a hint that it is OK to flush cached data (such as security tokens) or state information.

- The sign-out mechanism allows requestors to send a message to its IP/STS indicating that the requester is initiating a termination of the SSO.

Sign-Out-Message

- The sign-out mechanism allows requestors to send a message to its IP/STS indicating that the requester is initiating a termination of the SSO.
- For SOAP, the action of this message is as follows:
  - http://schemas.xmlsoap.org/2003/07/Federation#SignOut

<signout> syntax



```
<wsse:SignOut wsu:Id="...">
    <wsse:Realm>...</wsse:Realm>
    <wsse:SignOutBasis>...<wsse:SignOutBasis>
</wsse:SignOut>
```

- **SignOut:** This element represents a sign-out message.
- **Realm:** This optional element specifies the "realm" to which the sign-out applies and is specified as an Endpoint Reference.
- **signOutBasis:** The contents indicate the principal that is signing out. Content can include **security token reference** or **user name token**.
- **wsu:Id :** It include web service user id.

SAML vs WS-Federation

- SAML and WS-Federation are both standards that allow users that have already logged into one site to access another site without logging in again.

- WS-Federation is primarily championed by Microsoft Corporation which has invested heavily into incorporating WS-Federation into its products.

- SAML is an older specification that is well supported by many identity management vendors.

- However, most vendors, including Microsoft, are moving to support both standards.

SAML vs WS-Federation

- In both WS-Federation and SAML, the client is redirected by the SP to an IP with a HTTP 302 error.
- The authentication request is transmitted by the following URL parameters:
  - For SAML: parameter *SAMLRequest* containing an XML document of *<samlp:AuthnRequest>* type.
  - For WS-Federation: parameter *wa* containing the *wsignin1.0* value and parameter *wreq* containing an XML document of *<wst:SecurityTokenRequest>* WS-Trust type.
- A valid authentication response is conveyed by a HTTP POST with the following parameters:
  - For SAML: parameter *SAMLResponse* containing an SAML assertion.
  - For WS-Federation: parameter *wresult* contaning an XML document of *<wst:SecurityTokenRequestResponse>* WS-Trust type which itself contains an SAML assertion

SAML vs WS-Federation

SAML	WS-Federation
The web application sends a *SAML request* to the identity provider. *<samlp:AuthnRequest>* XML documnet type	The web application sends *Query parameters* in a *Request Security Token* (RST) as the request to the identity provider. *<wst:SecurityTokeRequest>* XML WS-Trust type.
After verifying the user's identity, the identity provider returns a *SAML response*. Inside that SAML response is a *SAML assertion*.	After verifying the user's identity, the identity provider returns a *Request Security Token Response* (RSTR). Inside that RSTR is a *SAML assertion*. The WS-Trust can itself contain SAML assertion.
You can specify to sign the SAML assertion, the SAML response, or both.	RSTRs are always signed.