Seguridad en SSO

02/27/06

José Ramón Palanco jose.palanco@hazent.com

Hazent Systems SL

Antecedentes

- Internet = negocio + interacciones personales.
- La identidad en internet ahora está fragmentada a través de gran cantidad de proveedores de identificación.

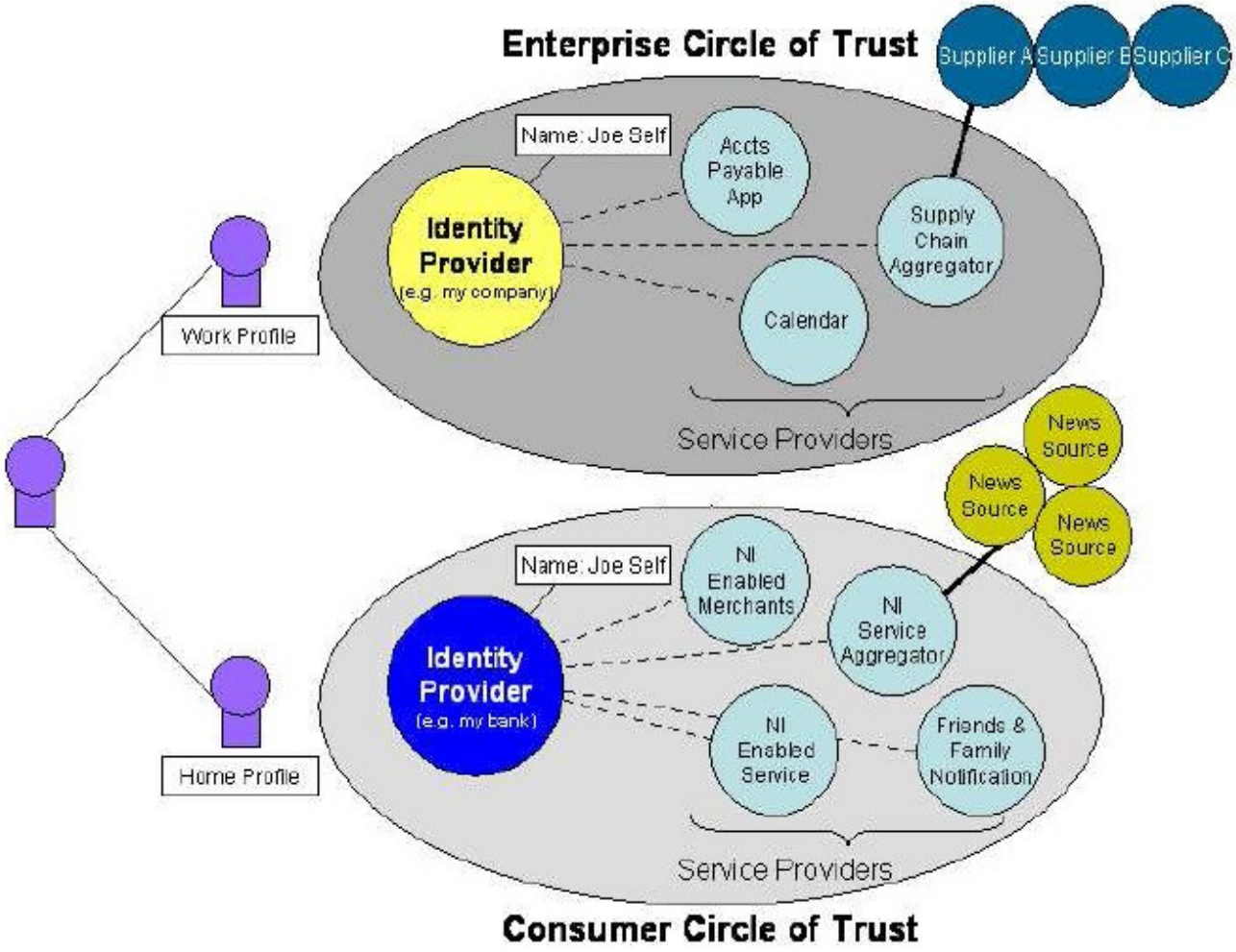
Intento de cohesión

- Redes federadas de identificación: reducción de pasos, creación de nuevos modelos y oportunidades de negocio...
- Un usuario tendrá una identidad en línea, perfil personal, configuraciones personalizadas, hábitos de compra, preferencias, .. que serán administradas por el usuario y compartidas con las organizaciones que el usuario elija.
- Un modelo de red de estas características debe asegurar que esta información realmente solo sea accesible para entidades apropiadas.

Identidad de Red

- Objetivos de identidad de red:
  - Protección y seguridad de datos privados.
  - Facilitar el comercio a través de internet.
  - Creación de una infraestructura accesible.
- Para ello es necesario crear circulos de confianza utilizando la tecnología como Liberty.

Círculos de confianza



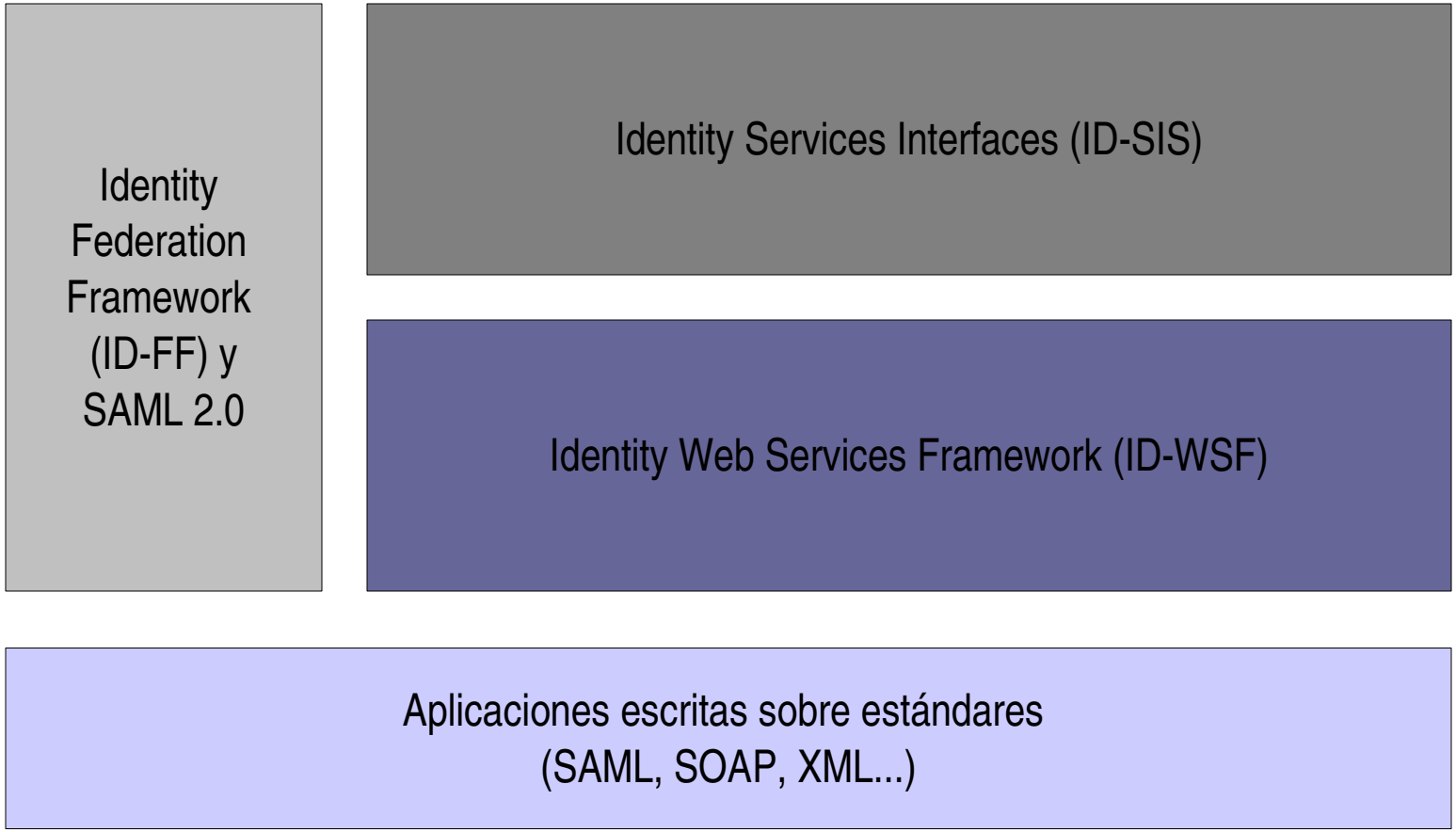
Conceptos

- **Principal**: usuario (persona que puede ser autenticada)
- **IdP**(Identity Provider): servidor de autenticación y aserción de la identidad de un principal.
- **SP** (Service Provider): Proveedor de servicios.
- **Federación**: establecer una relación entre dos entidades.
- **SSO** (Signgle-SignOn): la posibilidad de que un principal se autentique ante un IdP y esa autenticación valga también para otros sistemas (proveedores de servicio principalmente).
- **Círculo de confianza**: un grupo de IdP's y SP's que comparten relaciones comerciales basadas en la arquitectura Liberty.

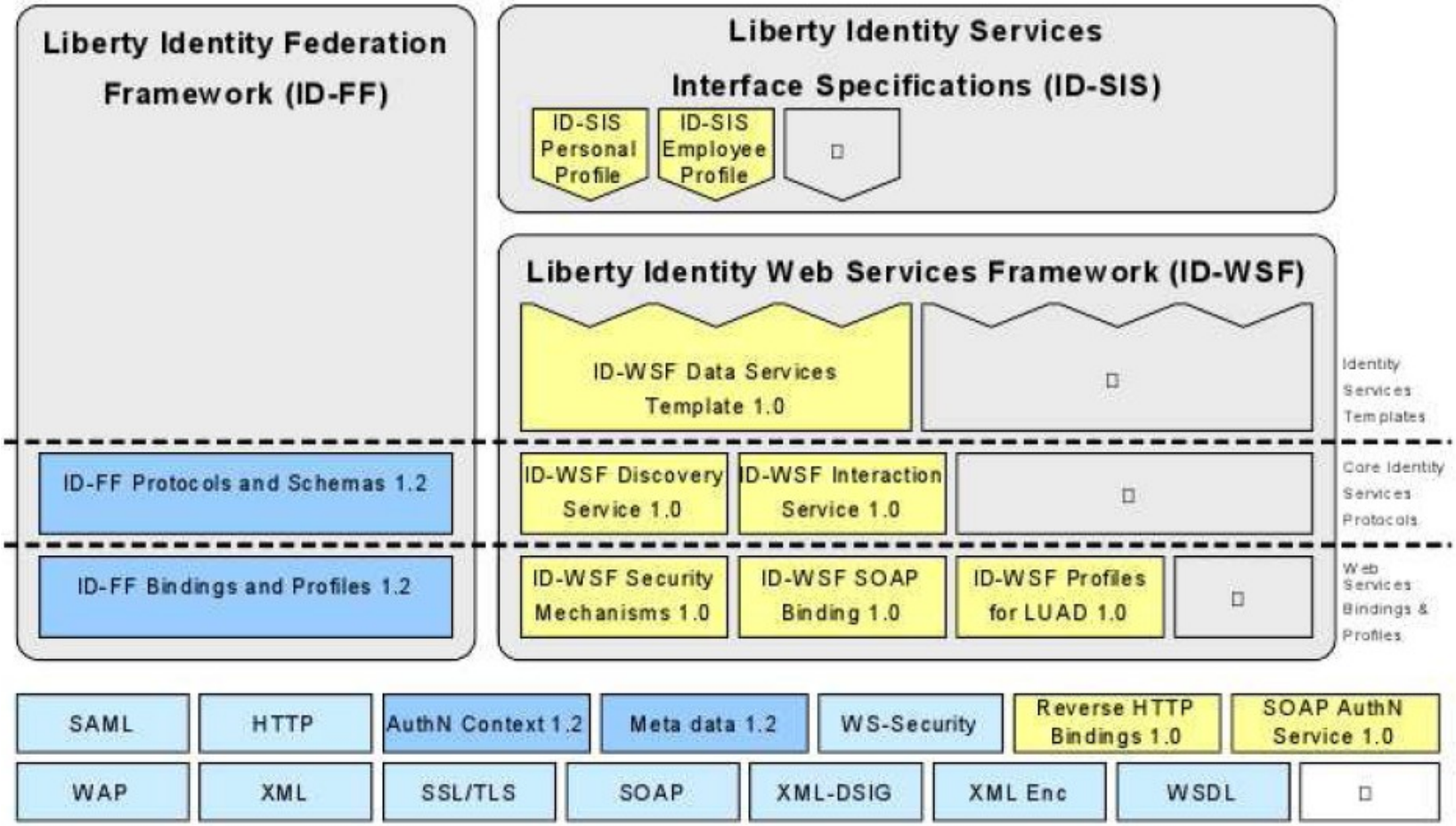
Framework de Servicios de Identidad (arquitectura de protocolos)

- Define una arquitectura por capas usando SOAP y sin definir el contenido en el cuerpo del mensaje SOAP con el fin de permitir el desarrollo de servicios de identidad

Arquitectura de Liberty



Módulos de Liberty



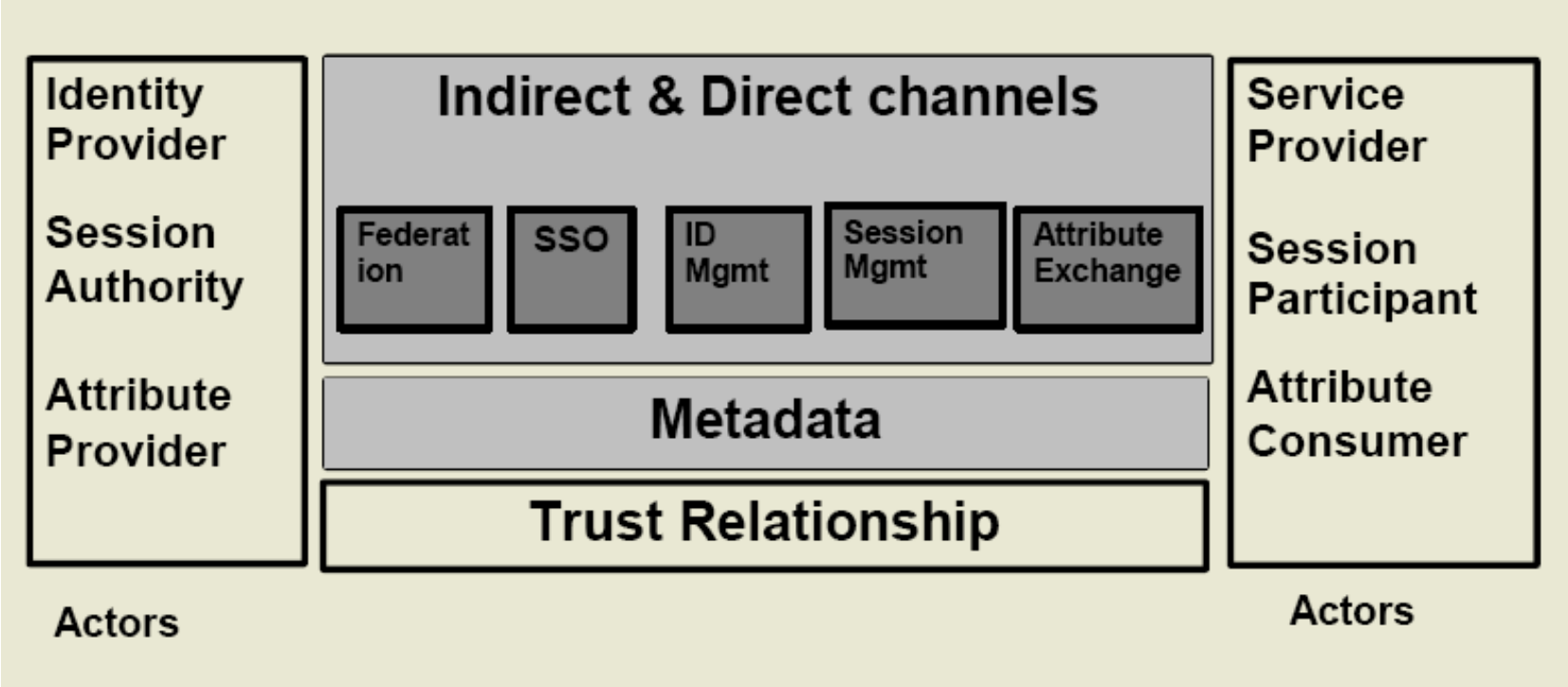
ID-FF

- Posibilita la federación de identidades mediante características tales como enlace de cuentas/identidades, single-signon y manejo simple de sesiones.
- Amplía los estamentos de autenticación de SAML, añadiendo conceptos como sesión o contexto de autenticación.
- Crea un protocolo de autenticación.

Security Assertion Markup Language

- Desarrollado por Oasis.
- Define aserciones que pueden llevar estamentos. Confirma que un usuario es quien dice ser.
- Provee de un protocolo de emisión de aserciones entre autoridades y terceros.
- Define un conjunto de perfiles que simplifican SSO a través de la Web.

Actores y mecanismos de SAML 2.0



Componentes de SAML

- Definición de estructura y contenido de las **aserciones**.
- Esquema XML del **protocolo**.
- Definición de **bindings** (canales de comunicación: HTTP, SOAP, ...)
- **Perfiles** creados a partir de la combinación de protocolos y bindings.

Aserciones

- Las aserciones son declaraciones de hechos, de acuerdo con el emisor.
- Son el cómputo de varios tipos de estamentos sobre un sujeto:
  - autenticación
  - atributo
  - decisión de autorización
- Se puede extender SAML para cubrir las necesidades de nuestros propios estamentos.
- Las aserciones pueden firmarse digitalmete usando firmas XML.
- Pueden contener información sobre el emisor y el sujeto.
- Consejos adicionales (para informar de como o porque una aserción fue emitida)
- Pueden ser usadas como tokens para asegurar los mensajes SOAP.

Ejemplo de aserción

```
<saml:Assertion
Version=”2.0” ID="buGxcG4gIL" IssueInstant="2002-06-19T17:05:37.795Z">
<saml:Issuer Format=”entity”>IDP.com</saml:Issuer>
<ds:Signature></ds:Signature>
<saml:Conditions NotBefore="2002-06-19T17:00:37.795Z"
NotOnOrAfter="2002-06-19T17:10:37.795Z"/>
<saml:Subject>
<saml:NameID NameQualifier=http://www.acompany.com
Format="persistent">jsdyfhs</saml:NameID>
<SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer"/>
</saml:Subject>
<saml:AuthnStatement
AuthnInstant="2002-06-19T17:05:17.706Z">
<AuthnContext>
<AuthnContextClassRef>Password</AuthnContextClassRef>
</AuthnContext>
</saml:AuthnStatement>
</saml:Assertion>
```

Protocolos

- **Petición de autenticación**: define un mensaje <AuthnRequest> que produce un <Response> con una o más aserciones que son devueltas al Principal (emitido por el SP con el resultado de la respuesta del IdP).
- **De artefacto**: nos permite acceder a una aserción creada con anterioridad mediante una referencia, esta referecia es conocida también como artefacto.
- **De Single Logout**: define la petición que inicia el proceso de cierre de sessión (ya sea iniciado por el Principal o por timeout de la aplicación).
- **De adm. de nombre identificador**: provee de macanismos o métodos que permiten cambiar el valor o el formato del nombre del Principal.
- **De mapeo de nombre identificador**: permite a un SP pedir a un IdP un identificador mapeado comprensible por otro proveedor.
- **De petición y emisión de aserciones**: define un conjunto de peticiones que permiten devolver una aserción existente.

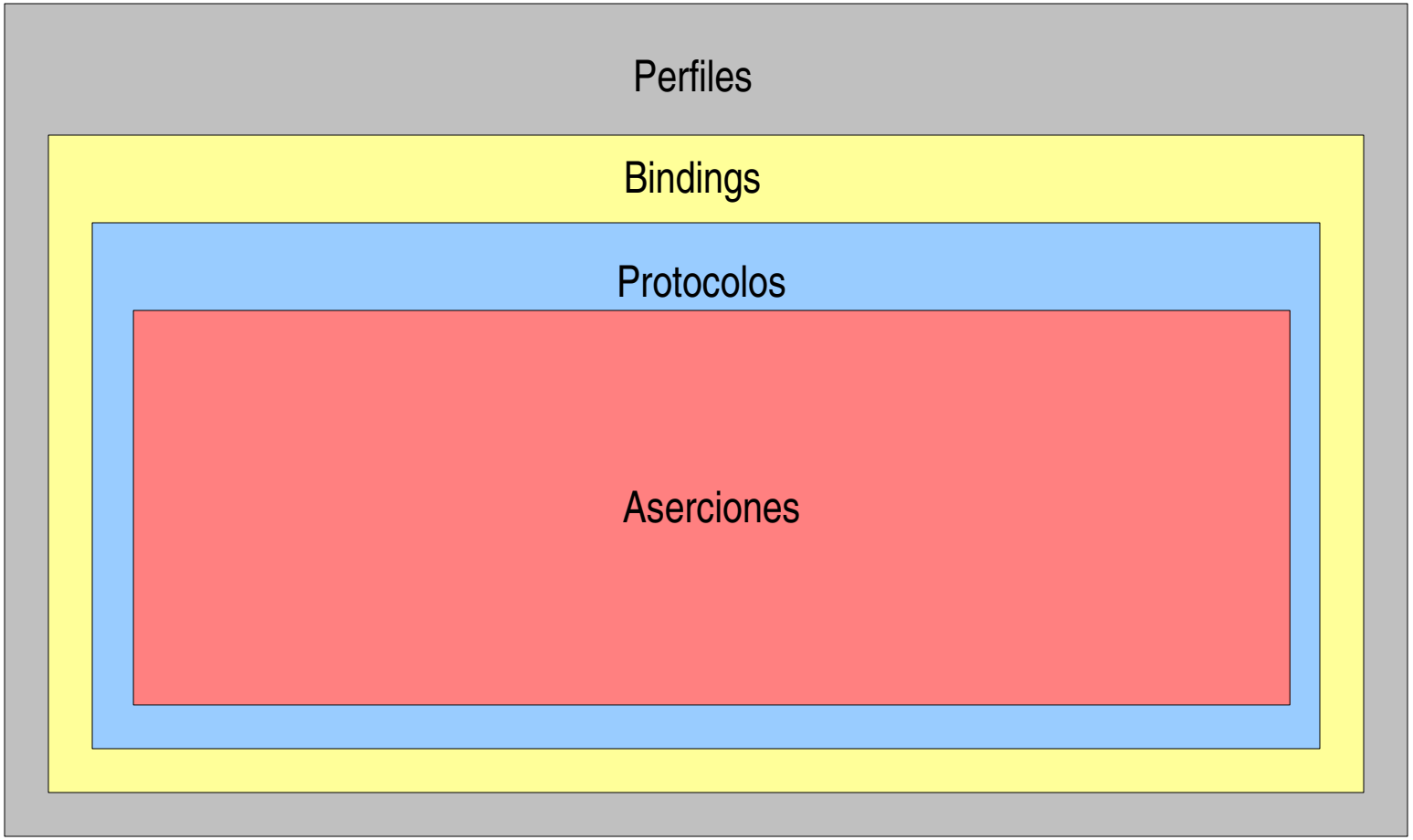
Bindings

- Define como debe mapearse SAML sobre protocolos de transporte.
- Los bindingd definidos son:
  - **SAML SOAP**: transporte de SAML sobre SOAP 1.1 y SOAP sobre HTTP
  - **HTTP Redirect**: envío a través de redirecciones HTTP (código 302)
  - **HTTP Artifact**: define como una referencia a una petición o respuesta SAML es transportada por HTTP. Existen 2 mecanismos: mediante formulario ó mediante cadena en una URL.
  - **HTTP POST**: como un mensaje SAML puede ser enviado codificado en base64 en un formulario HTML.
  - **PAOS** (reverse SOAP): especifica como un cliente HTTP puede responder a SOAP (especial para Wap Gateways).
  - **SAML URI**: define como una aserción de SAML puede obtenerse a partir de una resolución de URI.

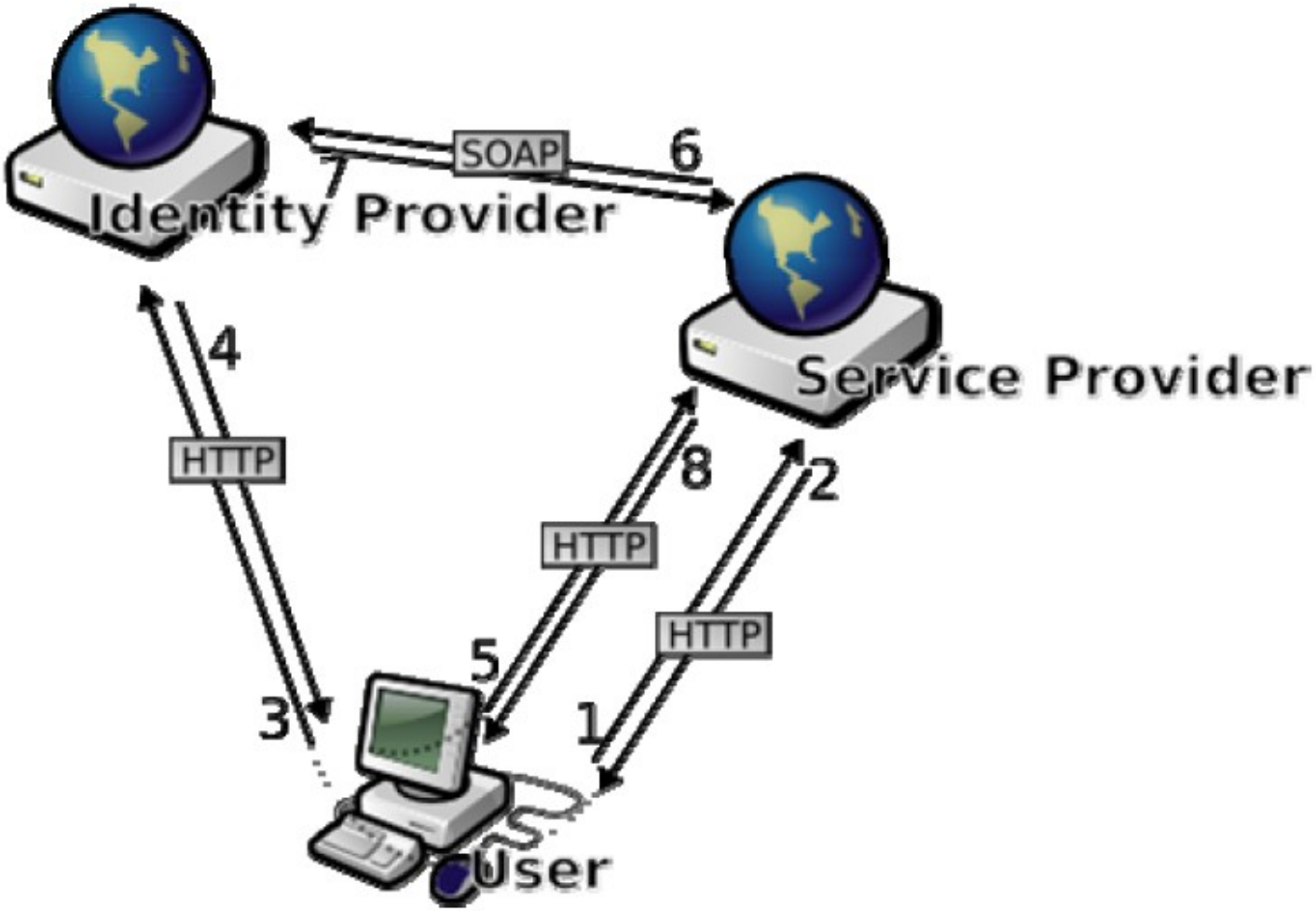
Perfiles (combinación de protocolos y bindings)

- **Navegador SSO**: define como un navegador puede tener soporte de SSO (usando mensajes de protocolo <AuthnRequest> combinado con bindings de redirecciones HTTP, HTTP POST y HTTP artifact.
- **Enhanced Client and Proxy** (ECP): define como usar mensajes de protocolo <AuthnRequest> con PAOS. Diseñado para dispositivos móviles.
- **Identity Provider Discovery**: define como un SP puede averiguar que IdP está usando el Principal.
- **Single Logout**:define como han de usarse los bindings para cerrar la sesión de los servidores federados.
- **Name Identifier Management**: como usar el protocolo de adm. de nombre identificador mediante los bindings.
- **Artifact Resolution**: uso de bindings siíncronos (como SOAP).
- **Assertion Query/Request**: uso de bindings siíncronos (como SOAP).
- **Name Identifier Mapping**: uso de bindings siíncronos (como SOAP).

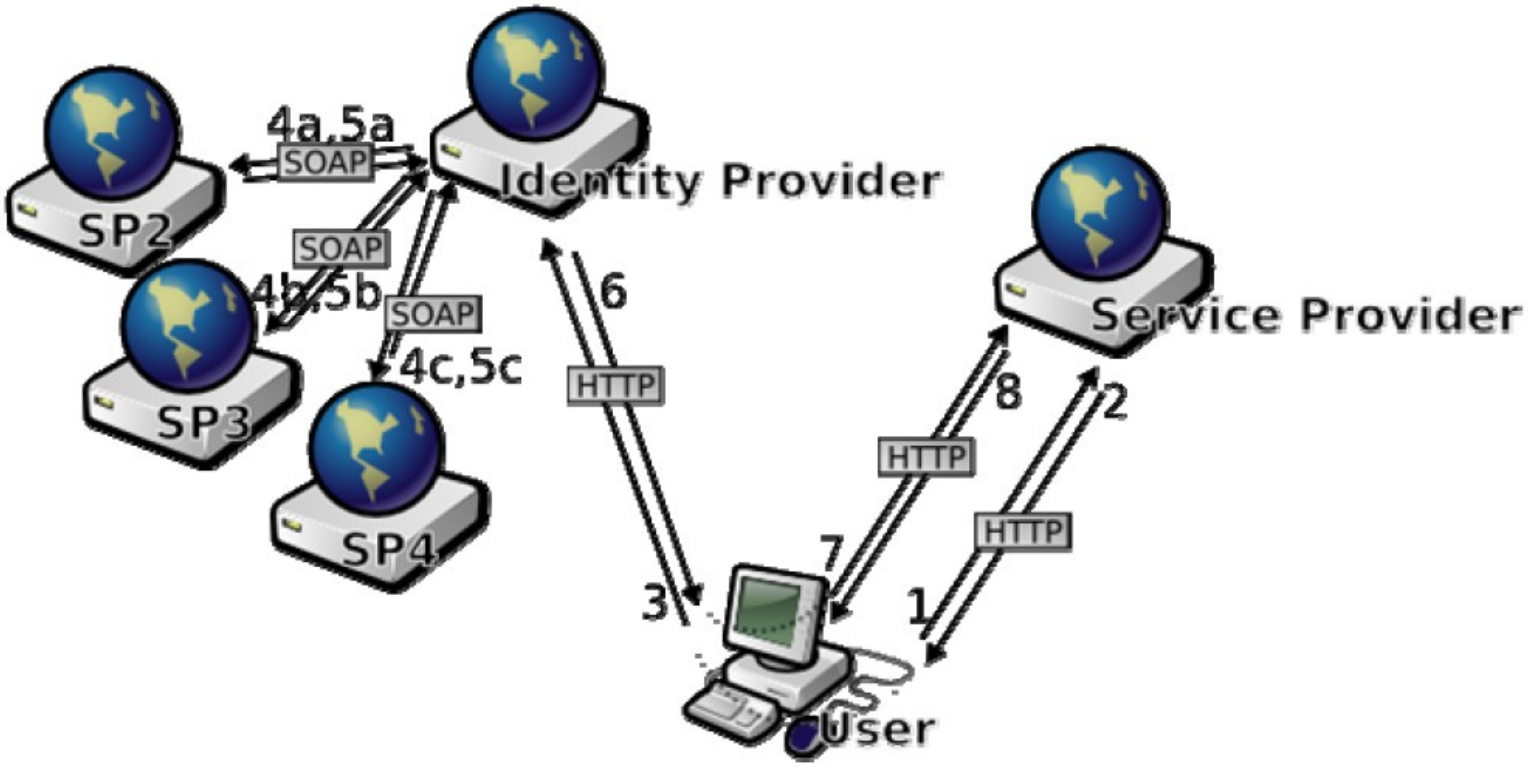
Encajando las piezas



Single SignOn



Single Logout



Probando SSO y Federación

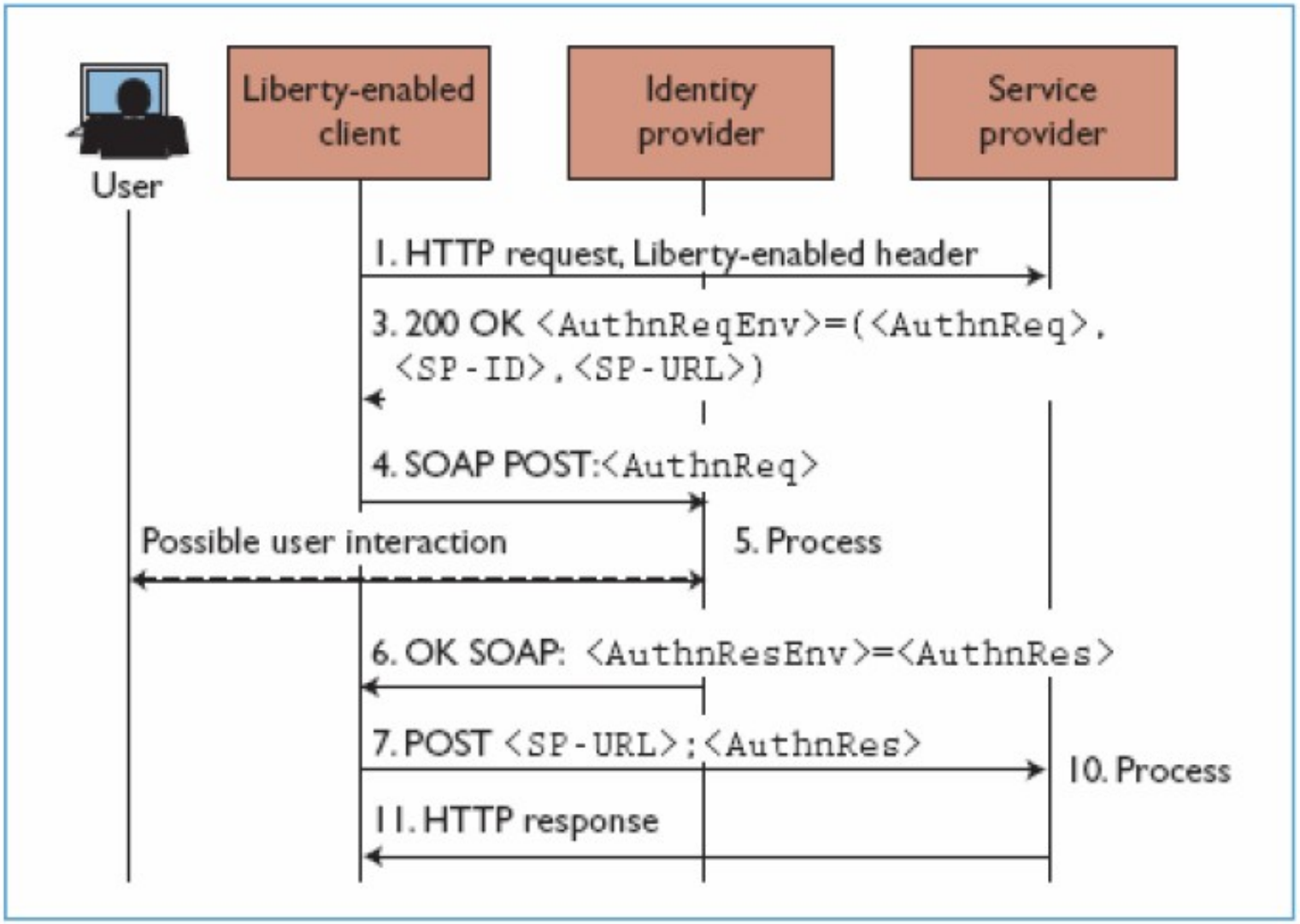
Hacking Liberty

- Actualmente no se conocen fallos importantes en el diseño de Liberty, sin embargo en la versión 1.0 existía un perfil llamado LECP (antecesor de ECP) que era vulnerable a un ataque de Man in the Middle. Esta vulnerabilidad se corrigió en la versión 1.1.
- LECP corresponde a las siglas Liberty Enabled Client and Proxy y permitía a los usuarios de navegadores simples (terminales móviles y navegadores ligeros) realizar Single SignOn.

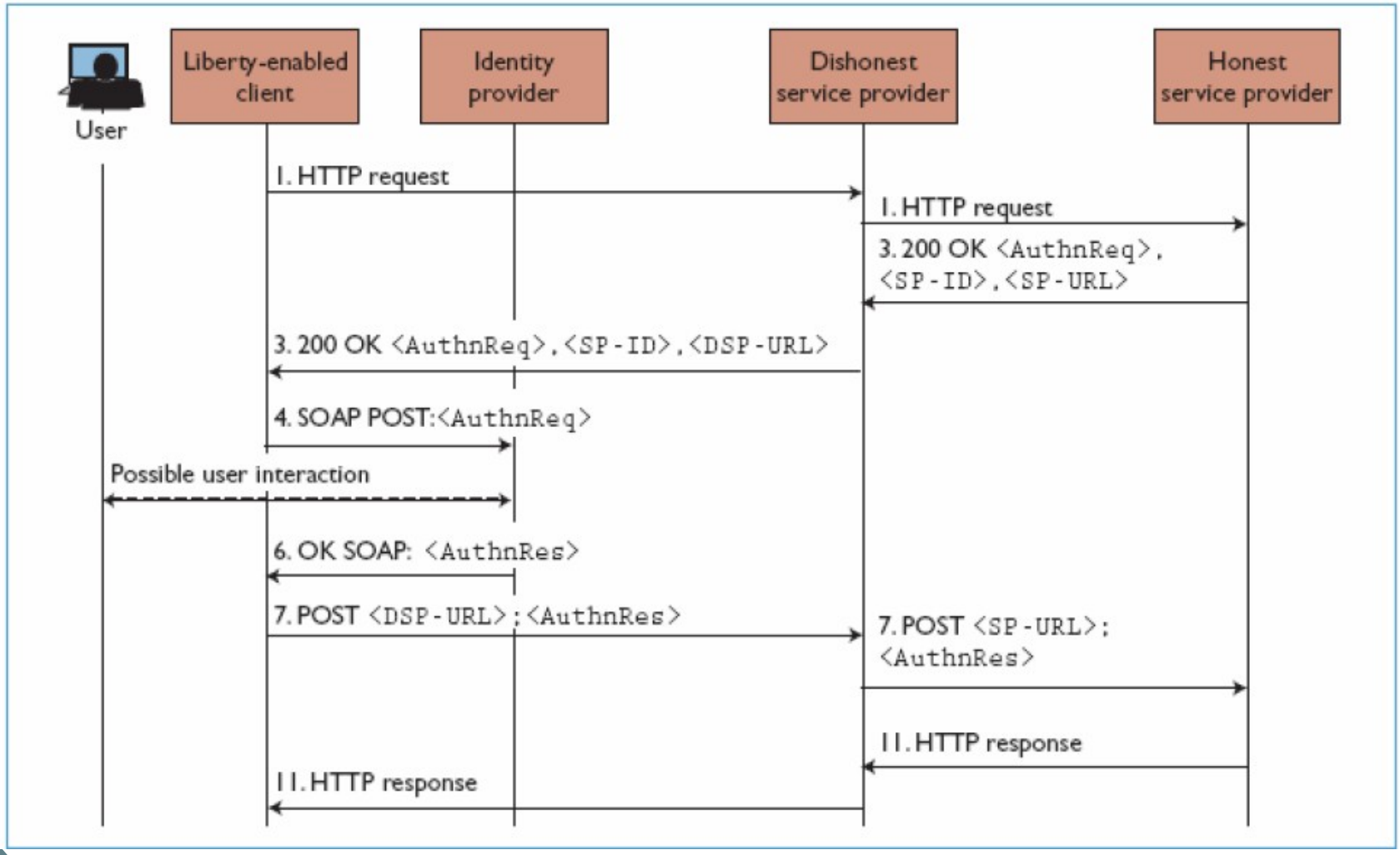
LECP

1.- Un cliente con un SP vía http indica que es LE (liberty enabled)
3.- Si el SP quiere autenticar al cliente y este entiede la cabecera LE, le envía la petición <AtuthReq> al cliente en un soap envelope <AuthnReqEnv> que incluye la identidad del SP <SP-ID> en la dirección <SP-URL> del proveedor de servicios que desea.
4.-El cliente desempaqueta el sobre (SOAP ENVELOPE) y envía un mensaje vía SOAP al IDP.
5.- El IDP comprueba la identidad del usuario y prepara una respuesta <AuthnRes> para el SP, que a su vez se la envía en un SOAP ENVELOPE <AuthnResEnv>.al cliente.
7.- El cliente obtiene la respuesta del sobre y envía el contenido a la dirección del SP <SP-URL>.
10.- El SP procesa la respuesta.
11.- Dependiendo de la respuesta, el SP continúa el trato con el cliente.

LECP cont.


User
Liberty-enabled client
Identity provider
Service provider
1. HTTP request, Liberty-enabled header
3. 200 OK <AuthnReqEnv>=(<AuthnReq>, <SP-ID>, <SP-URL>)
4. SOAP POST:<AuthnReq>
Possible user interaction
5. Process
6. OK SOAP: <AuthnResEnv>=<AuthnRes>
7. POST <SP-URL>:<AuthnRes>
10. Process
11. HTTP response

LECP Man In The Middle



FIN

http://www.zeroday.es/