Poniendo a SIR por las nubes

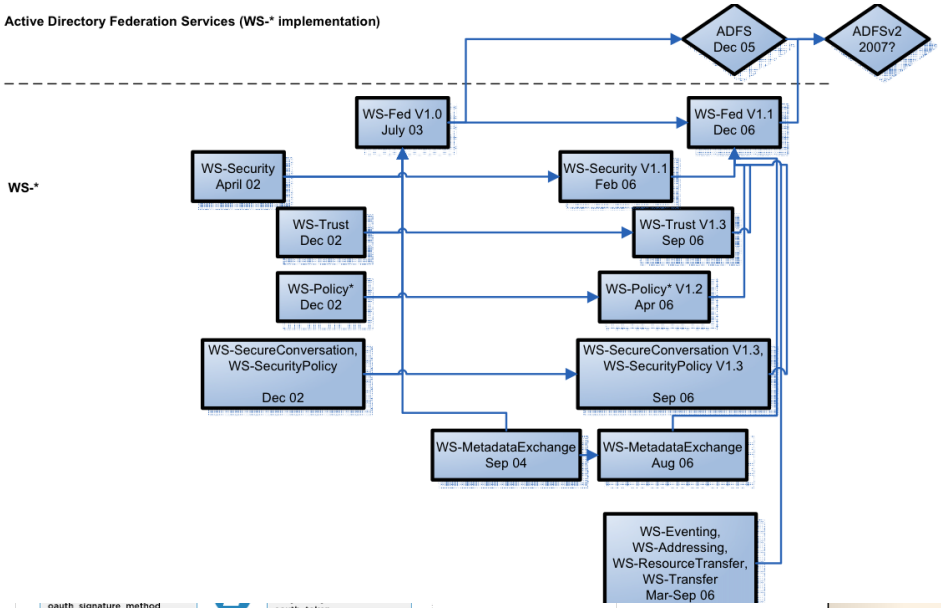
WS y externalización con federación de identidad

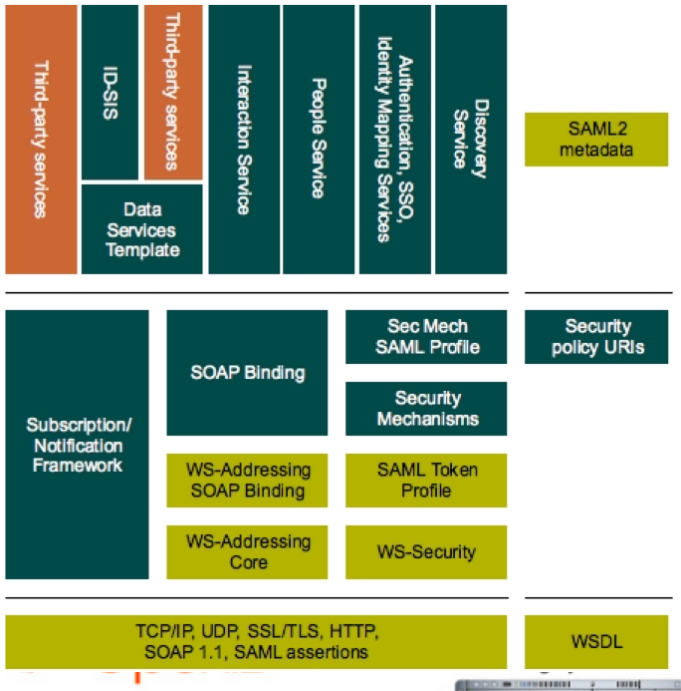
Diego R. Lopez
RedIRIS

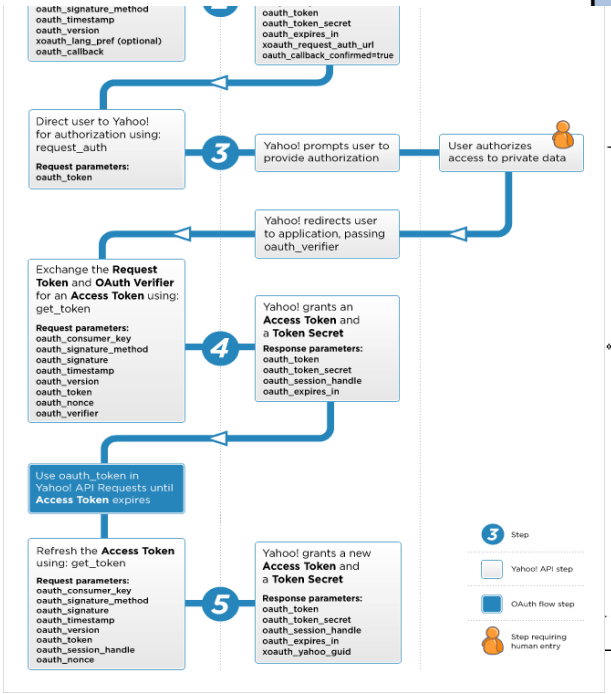
Nubes e identidad

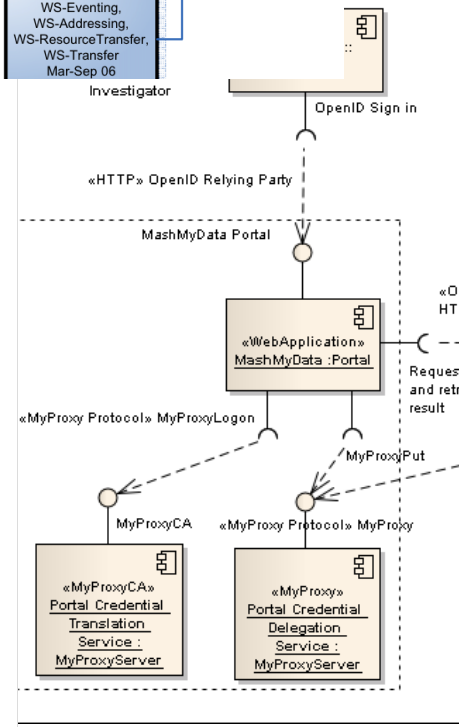
- Sin juegos de palabras ni admoniciones
- Un cambio radical en el modelo de seguridad
  - Seguridad bajo acuerdo en vez de bajo control
  - Múltiples áreas de responsabilidad (y jurisdicción)
  - ¿De verdad hay SLAs?
- La identidad digital cobra una mayor relevancia
  - Control de acceso
  - Contabilidad
  - Trazabilidad
  - Privacidad
  - Y la tentación de pasar al otro lado

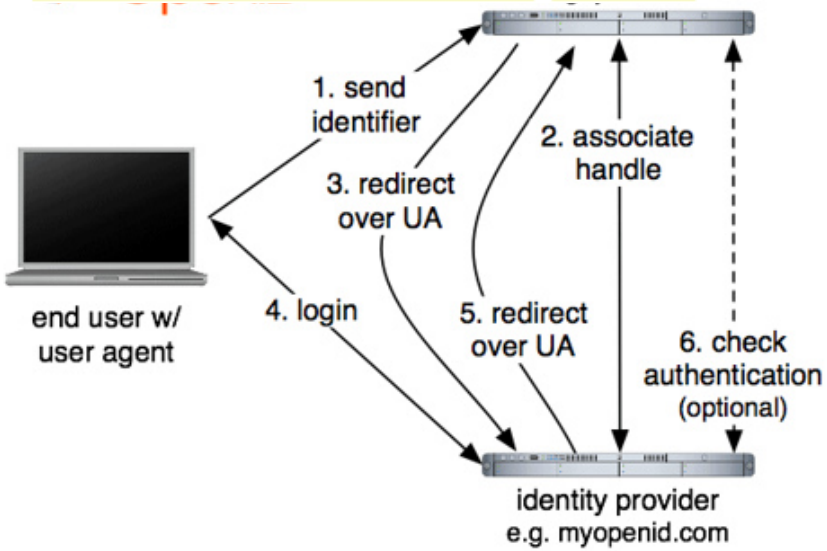
El paisaje


Active Directory Federation Services (WS-* implementation)
WS-*
ADFS Dec 05
ADFSv2 2007?
WS-Fed V1.0 July 03
WS-Fed V1.1 Dec 06
WS-Security April 02
WS-Security V1.1 Feb 06
WS-Trust Dec 02
WS-Trust V1.3 Sep 06
WS-Policy* Dec 02
WS-Policy* V1.2 Apr 06
WS-SecureConversation, WS-SecurityPolicy Dec 02
WS-SecureConversation V1.3, WS-SecurityPolicy V1.3 Sep 06
WS-MetadataExchange Sep 04
WS-MetadataExchange Aug 06
WS-Eventing, WS-Addressing, WS-ResourceTransfer, WS-Transfer Mar-Sep 06


Third-party services
ID-SIS
Third-party services
Data Services Template
Interaction Service
People Service
Authentication, SSO, Identity Mapping Services
Discovery Service
SAML2 metadata
Subscription/ Notification Framework
SOAP Binding
Sec Mech SAML Profile
Security policy URIs
Security Mechanisms
WS-Addressing SOAP Binding
SAML Token Profile
WS-Addressing Core
WS-Security
TCP/IP, UDP, SSL/TLS, HTTP, SOAP 1.1, SAML assertions
WSDL


Direct user to Yahoo! for authorization using: request_auth
Request parameters: oauth_token
Yahoo! prompts user to provide authorization
User authorizes access to private data
Yahoo! redirects user to application, passing oauth_verifier
Exchange the Request Token and OAuth Verifier for an Access Token using: get_token
Request parameters: oauth_consumer_key oauth_signature_method oauth_signature oauth_timestamp oauth_version oauth_token oauth_nonce oauth_verifier
Yahoo! grants an Access Token and a Token Secret
Response parameters: oauth_token oauth_token_secret oauth_session_handle oauth_expires_in
Use oauth_token in Yahoo! API Requests until Access Token expires
Refresh the Access Token using: get_token
Request parameters: oauth_consumer_key oauth_signature_method oauth_signature oauth_timestamp oauth_version oauth_token oauth_session_handle oauth_nonce
Yahoo! grants a new Access Token and a Token Secret
Response parameters: oauth_token oauth_token_secret oauth_session_handle oauth_expires_in xoauth_yahoo_guid
Step
Yahoo! API step
OAuth flow step
Step requiring human entry

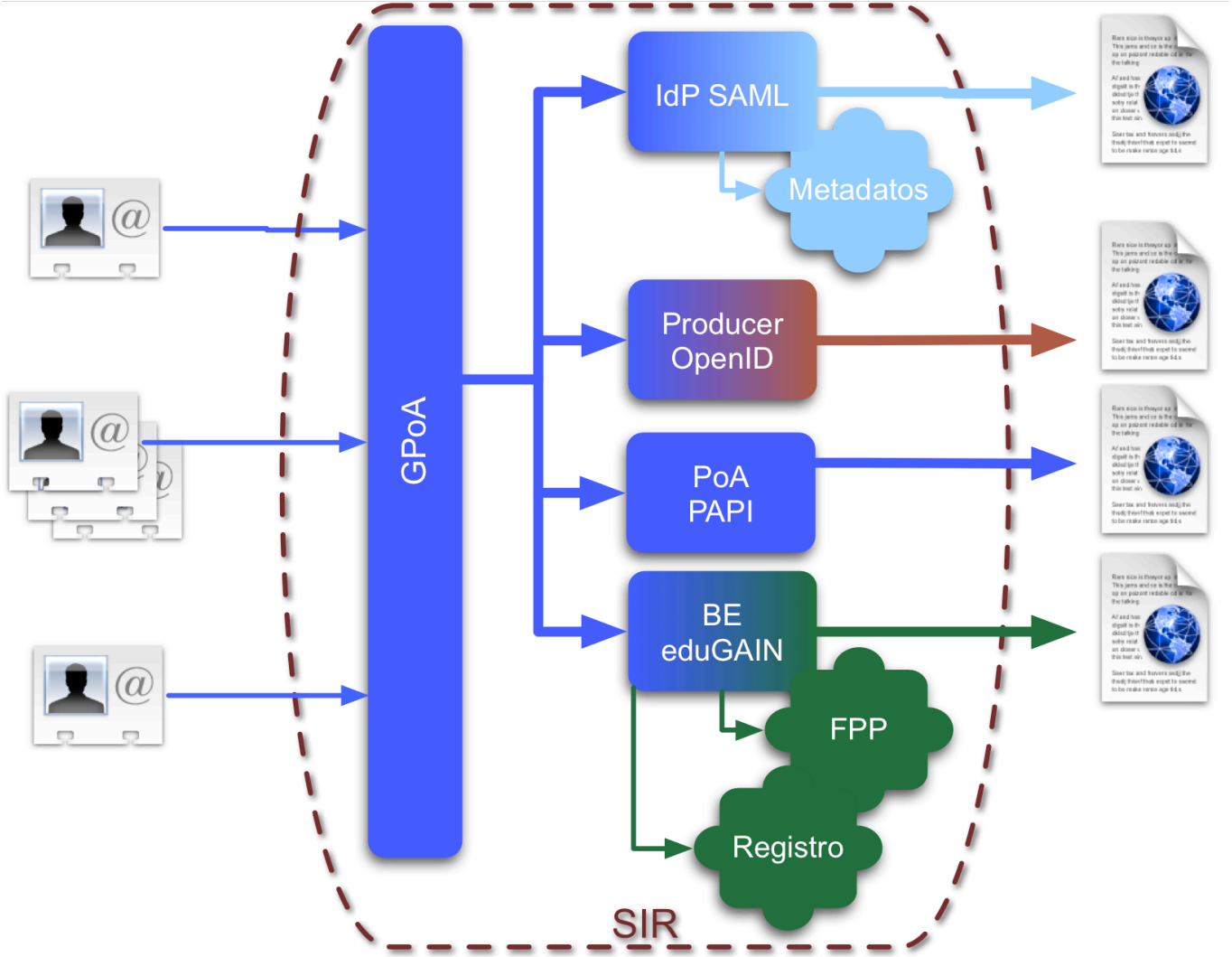

Investigator
OpenID Sign in
«HTTP» OpenID Relying Party
MashMyData Portal
«WebApplication» MashMyData :Portal
«MyProxy Protocol» MyProxyLogon
MyProxyPut
MyProxyCA
«MyProxy Protocol» MyProxy
«MyProxyCA» Portal Credential Translation Service : MyProxyServer
«MyProxy» Portal Credential Delegation Service : MyProxyServer


1. send identifier
2. associate handle
3. redirect over UA
4. login
5. redirect over UA
6. check authentication (optional)
end user w/ user agent
identity provider
e.g. myopenid.com

El modelo SIR

One Ring to bring them all and in the darkness bind them

In the Land of Mordor where the Shadows lie.



Conectores: Entrada

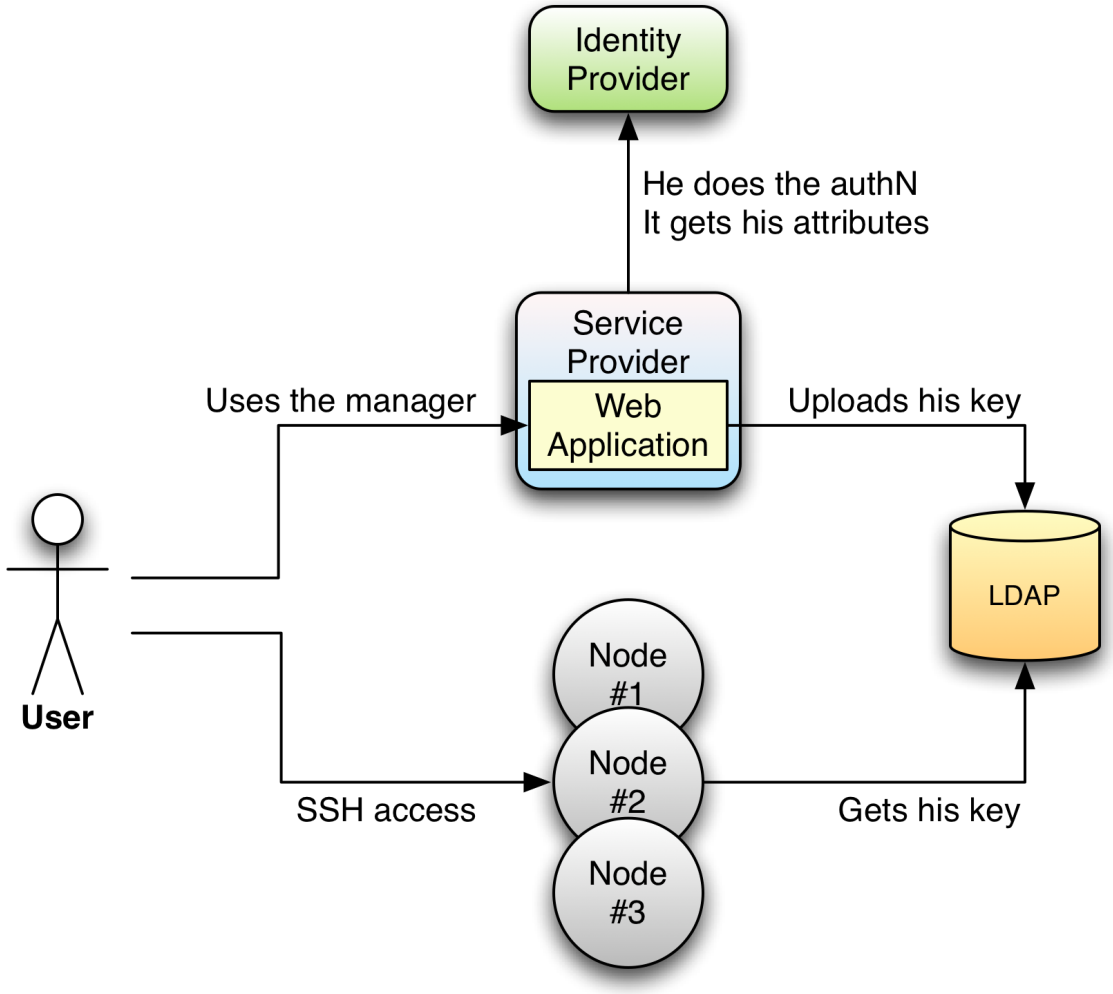
- Instituciones afiliadas a RedIRIS
  - Organizaciones virtuales relacionadas con ellas
- Deben disponer de un conector
  - Capaz de producir aserciones PAPI a partir de datos propios
  - PHP, Java (JSP y Filter), Apache, mod_perl Apache, ASP, Sun AM, AD, OSSO, WAM y algunos específicos
  - Fuentes de identificación y datos transparentes
- Recomendaciones en cuanto a atributos
- Aceptación de las condiciones de uso

Conectores: Salida

- Protocolo nativo
  - PAPI v1
- Estándares
  - SAML. 1 y 2
  - OpenID. Incluyendo SRE
  - OAuth. “Classic”
  - Y pronto: OAuth2, WS-Trust, SLCS
  - . . .
- Propietarios
  - TPS, MSDN-AA, Apple (`Referer`)
  - Pruebas con MS-Live y Facebook
  - . . .
- Y proxies

PAPIv1: FedSSH

- Las claves públicas se transmiten como atributos
  - O referencias a ellas
- LDAP como back-end
  - SQL o cualquier otro método
- Entornos basados en SSH
  - Esquemas equivalentes de seguridad: gestión de claves públicas



SAML: Google Apps

- SIR construye el identificador a partir de los atributos recibidos
  - Varias opciones
  - Privacidad
  - Conveniencia
- Gestión independiente
  - Procedimiento normalizado
  - Posibilidad de mayor integración

Google Apps para rediris.es - Edición para educación

Buscar cuentas | Buscar en el Centro de asistencia

Panel | Cuentas del usuario | Personalización del dominio | Herramientas avanzadas | Configuración del servicio

« Volver a Herramientas avanzadas

Editar inicio de sesión único (SSO)

Para configurar SSO, proporciónanos la siguiente información: Referencia de SSO

☑ Habilitar inicio de sesión único

URL de la página de acceso *

http://sir.rediris.es/rediris/saml2/idp/SSOService.php — URL para acceder a tu sistema y a Google Apps

URL de la página de fin de sesión *

http://sir.rediris.es/rediris/saml2/idp/SingleLogoutService.php — URL para redirigir usuarios cuando finalizan la sesión

Cambiar URL de contraseña *

http://www.rediris.es — URL para que los usuarios cambien su contraseña en tu sistema.

Certificado de verificación *

/Users/kan/Projects/RedIRI Browse... Subir

El archivo certificado debe contener la clave pública para que Google pueda verificar las solicitudes de acceso. Más información

◂ Inicio ◂ Servicios ◂ SIR ◂ Documentación técnica

Cómo configurar el acceso a Google Apps a través de SIR

SIR

Google Apps proporciona un método de acceso federado, denominado por ellos Single Sign On, que emplea el protocolo SAML2, soportado por SIR, y por tanto es posible usar las identidades digitales gestionadas a través de SIR para acceder a los servicios de Google Apps.

Para poder usar las identidades gestionadas por SIR en el acceso a Google Apps es necesario seguir los siguientes pasos:

1. El administrador del dominio se pondrá en contacto con el equipo de SIR para comunicarle su intención de conectar con GoogleApps a través de SIR. Es importante que en la comunicación detallen dos aspectos:
   1. El patrón que planean seguir para las cuentas de los usuarios en GoogleApps. Estas cuentas deben corresponderse con uno de los valores del atributo mail en las aserciones enviadas a través de SIR.
   2. Si la institución dispone de un mecanismo para hacer logout de su entorno de SSO Web y cuál es el URL para acceder al mismo. SIR proporciona un mecanismo de logout general a partir del enlace de fin de sesion en las aplicaciones de Google Apps, que puede ser extendido mediante el URL que mencionábamos para que se aplique también en el entorno SSO local.
2. El equipo de SIR responderá con un conjunto de datos que deben introducirse en el interface de administración de SSO para el dominio en Google Apps (**https://www.google.com/a/cpanel//SetupSSO**):
   - El URL del conector de SIR a Google Apps (campo **URL de la página de acceso**)
   - El URL del artefacto de logout SIR desde Google Apps (campo **URL de la página de fin de sesión**)
   - El certificado que debe asociarse al dominio para que Google Apps verifique los datos enviados por SIR (campo **Certificado de verificación**, opción de **Sustituir certificado**)

   NOTA: Dado que SIR es una infraestructura federada, el campo "Cambiar URL de contraseña" es una cuestión completamente interna a la institución.
3. El administrador del dominio comunicará al equipo de SIR la cumplimentación de los datos para que se active el conector. Para la activación es necesario que confirme si ha marcado o no la opción **Utilizar una determinada entidad emisora de dominios**. Si bien SIR puede funcionar en ambos casos, es necesario conocer su estado para configurar el conector. En cualquier caso, creemos que es recomendable activarla.
4. El equipo de SIR activará de manera temporal el conector para que los administradores del dominio realicen las pruebas pertinentes.
5. Una vez realizadas las pruebas, los administradores del dominio deben remitir al equipo de SIR el documento de Condiciones de Uso del SIR para SPs externos adecuadamente cumplimentado. Los datos en este caso son:

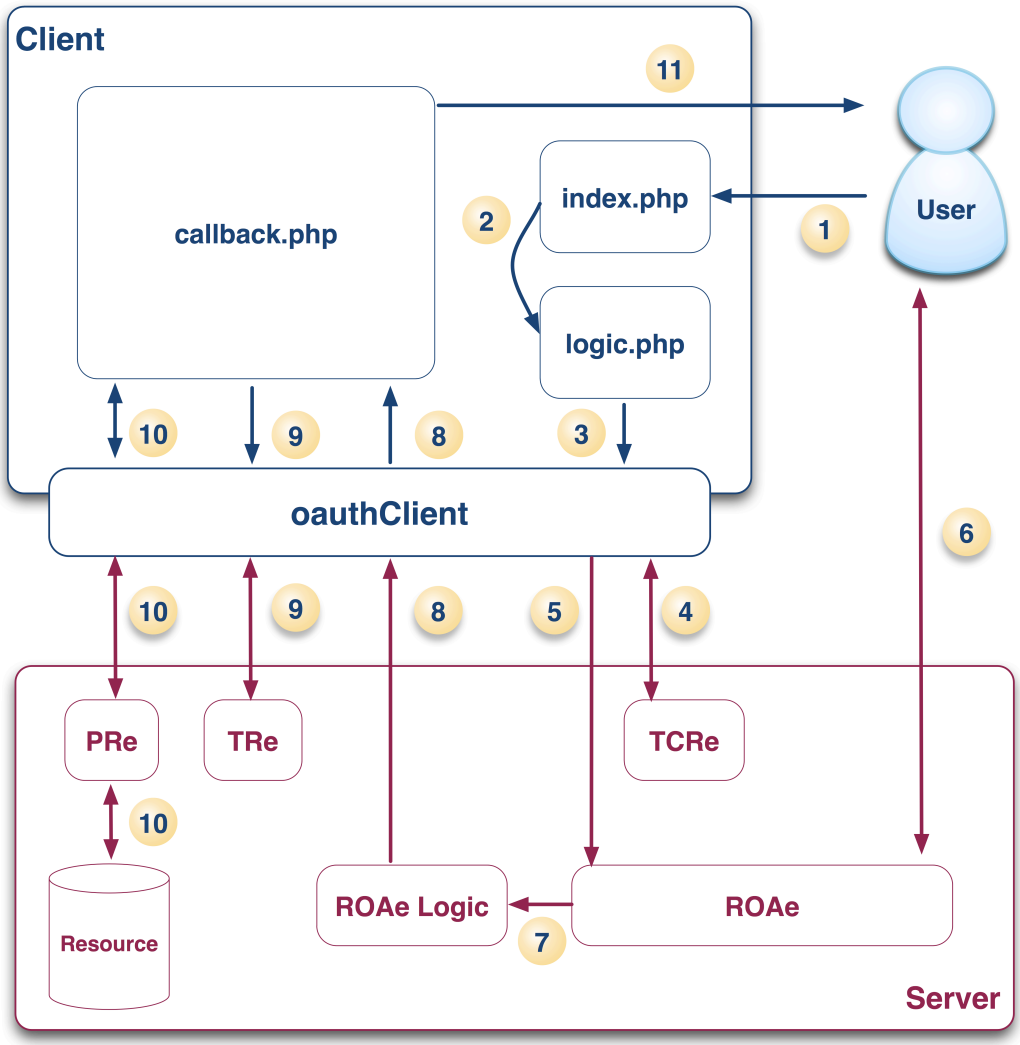
   ```
   URL: google.com
   Protocolo: SAML2
   Datos: mail
   ```

La conexión con Google Apps se considerará en pruebas hasta que el documento haya sido validado por el equipo de SIR.

OpenID: yo/jo/eu/ni

- Identificadores en cualquier lengua
  - Según el patrón yo.rediris.es/soy/…
  - Soporte de redirecciones OpenID2
  - OpenID derivado de diferentes atributos
  - Flexibilidad y privacidad
- SRE (*Simple Registration Extensions*)
  - Datos adicionales: mail, nombre, nickname/uid…
  - Implementado mediante filtros por SP y patrones de OpenID
- Aplicaciones web
- Servicios REST
- Un OpenID puede usarse como identificador único

OAuth "clásico"



1-3: El control pasa a la sección que implementa la lógica OAuth

4-5: Intercambio de credenciales cliente-servidor

6-7: Redirección al entorno SIR "normal"

8-9: Intercambio de tokens

10-11: Acceso al recurso

OAuth en SIR

- Hemos desarrollado
  - Un interface de registro
  - Una librería de servidor
  - Un cliente de referencia

Registro de Aplicaciones Clientes de OAuth

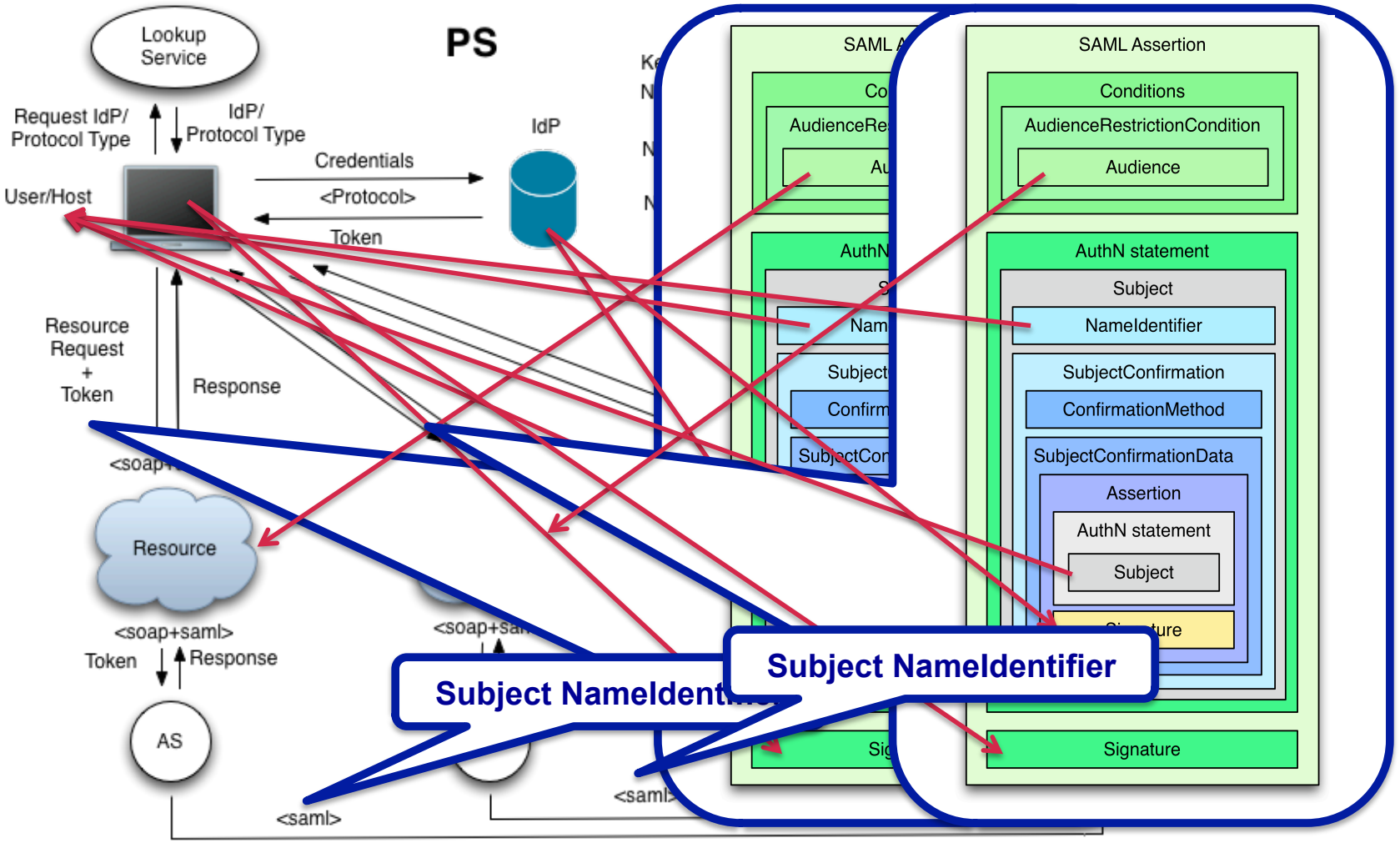
Backend

Peticiones:

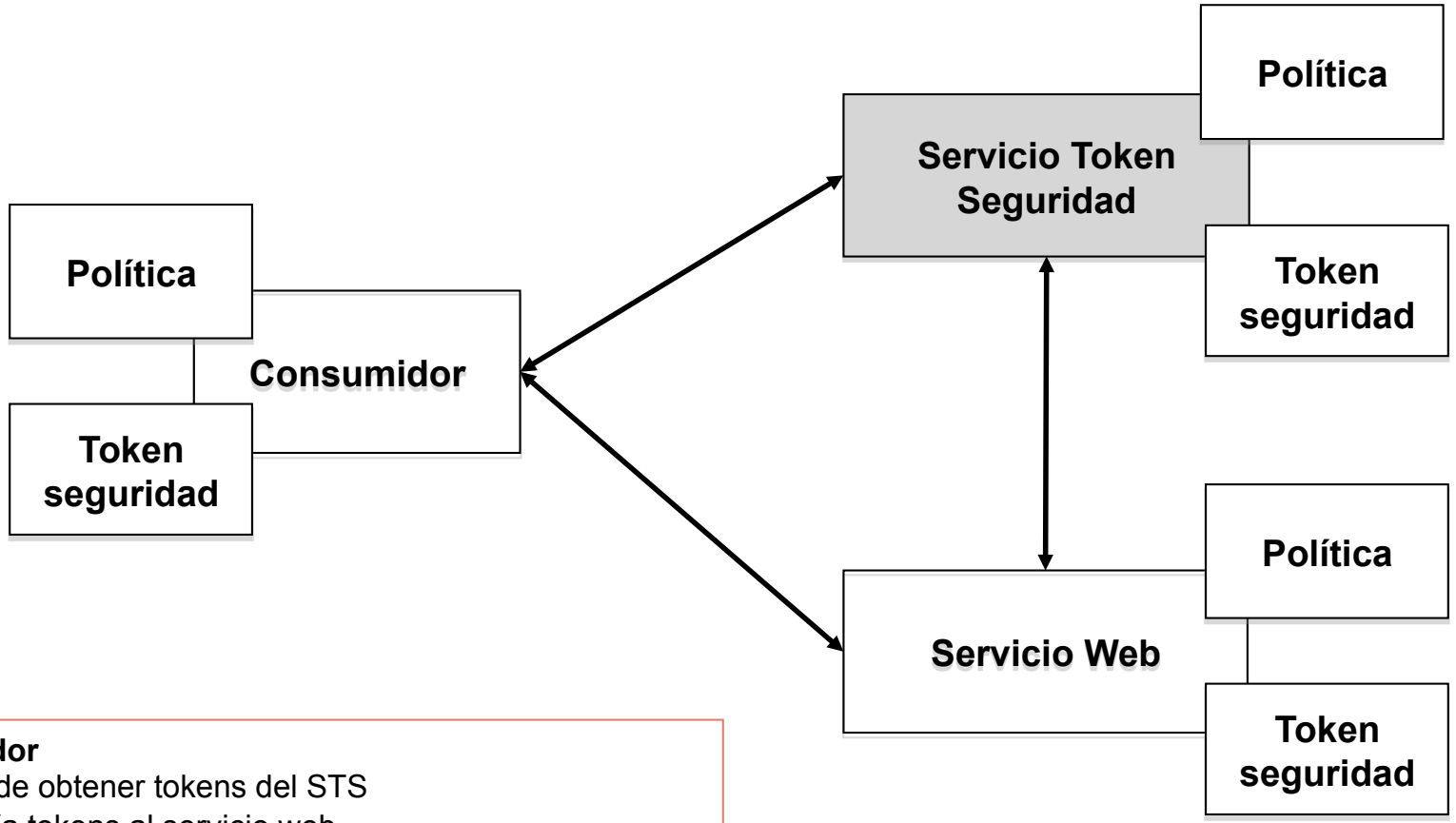
Usuario	Nombre App Cliente	Estado	Secreto		
elena.lozano@rediris.es	rediris.es:app1	Aceptado	5af5047e8b1320812693...	revocar	Borrar App
candido.rodriguez@rediris.es	rediris.es:one	Denegado	9ced40bb195a2e297e6f...	aceptar	Borrar App
elena.lozano@rediris.es	rediris.es:app2	Denegado	a297b1c3b1874d957234...	aceptar	Borrar App
elena.lozano@rediris.es	rediris.es:aplicacion1	Pendiente	82833016f1b877c51f8a...	aceptar denegar	Borrar App

Volver al Inicio

Tokens: PerfSONAR

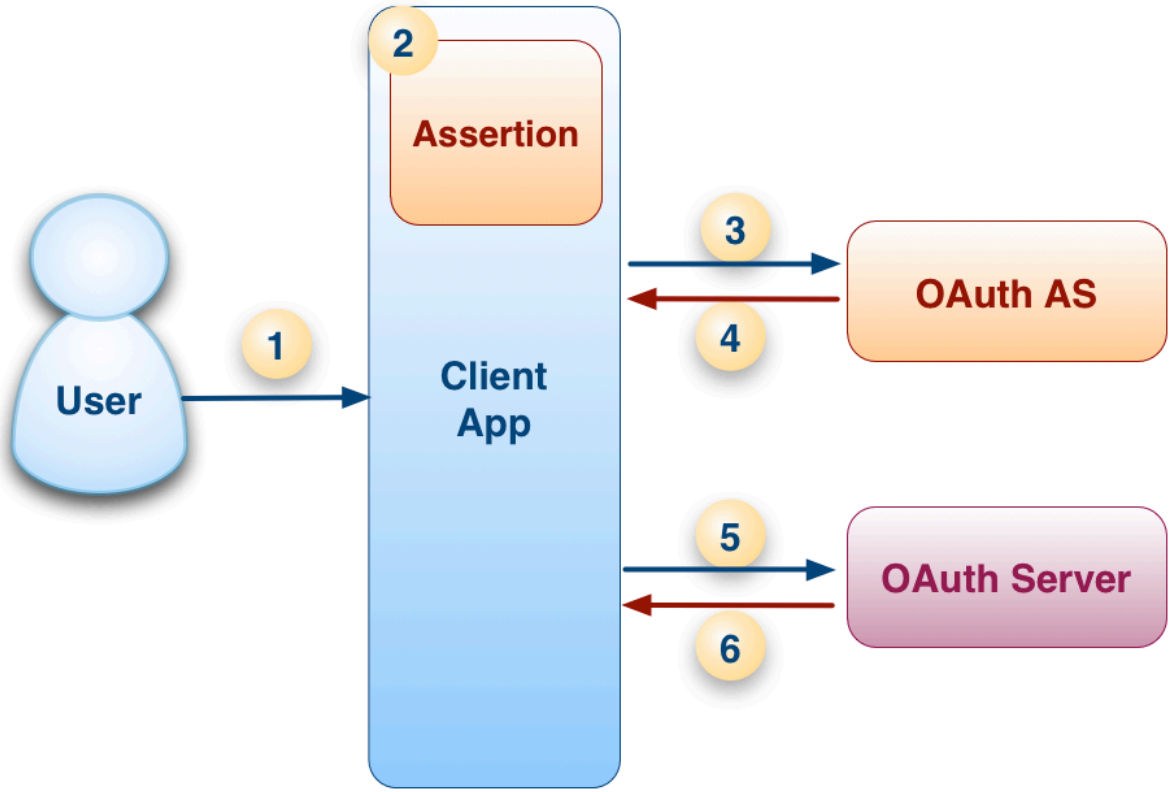


Tokens: STS



- **Consumidor**
  - -Puede obtener tokens del STS
  - -Envía tokens al servicio web
- **Servicio Token Seguridad (STS)**
  - -Emite, valida, renueva o cancela tokens de seguridad
- **Servicio Web**
  - - Recibe tokens del consumidor
  - - Puede utilizar el STS para validar el token, o validarlo él mismo

Tokens: OAuth2



1: El usuario accede al cliente

2: El cliente obtiene datos de SIR

3: El cliente los envía al AS

4: El AS genera un token

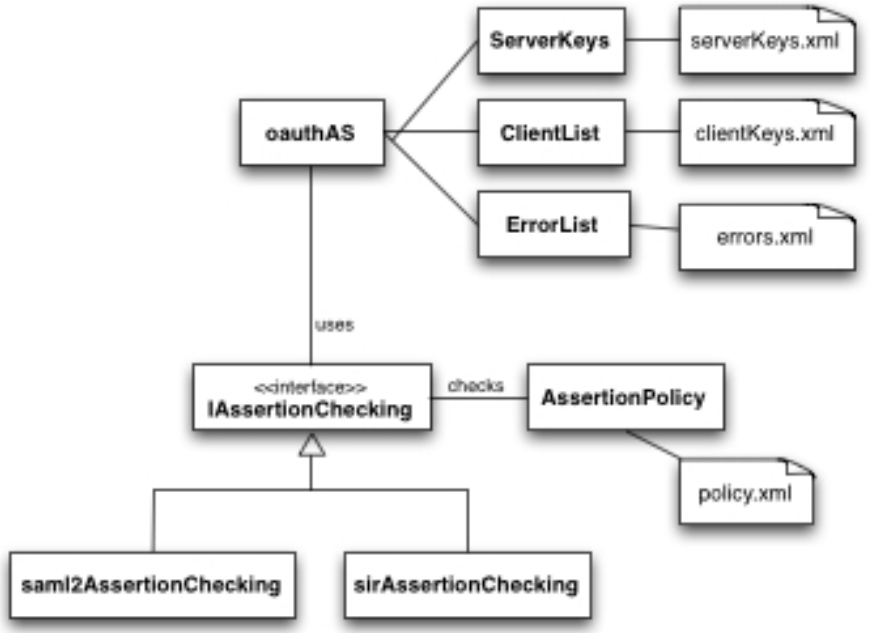
5-6: El cliente usa el token para acceder al servidor

http://www.rediris.es/oauth2/

La primera implementación libre del OAuth2 AP

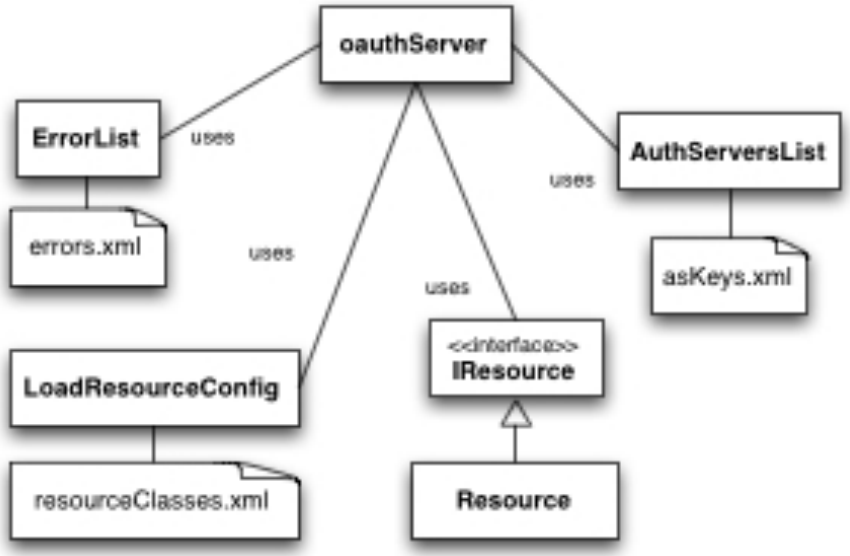
AS OAuth2

- Registro de servidores
  - Claves
  - Scopes
- Registro de clientes
  - Claves
- Políticas
  - Clientes
  - Atributos
  - Scopes
- Acepta aserciones SAML y PAPI
  - Interface extensible



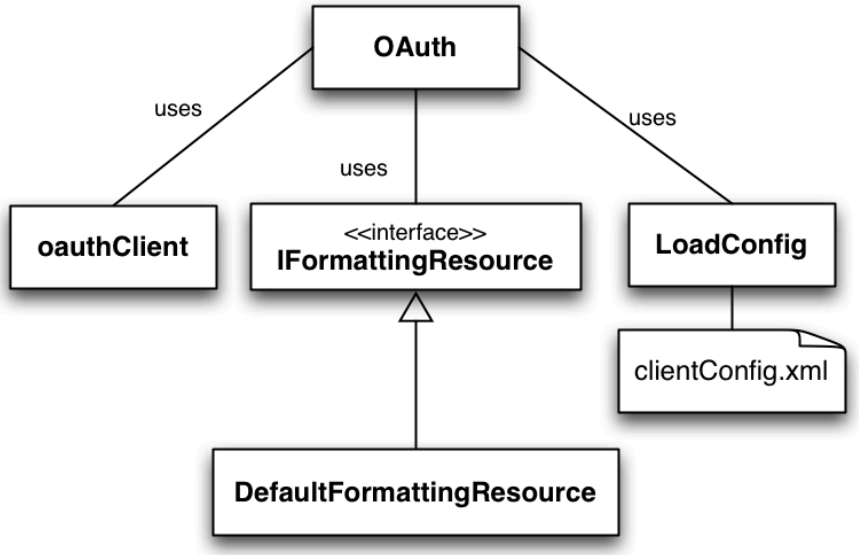
Servidor OAuth2

- ASes
  - Claves
- Recursos
  - Llamada a manejadores de contenidos

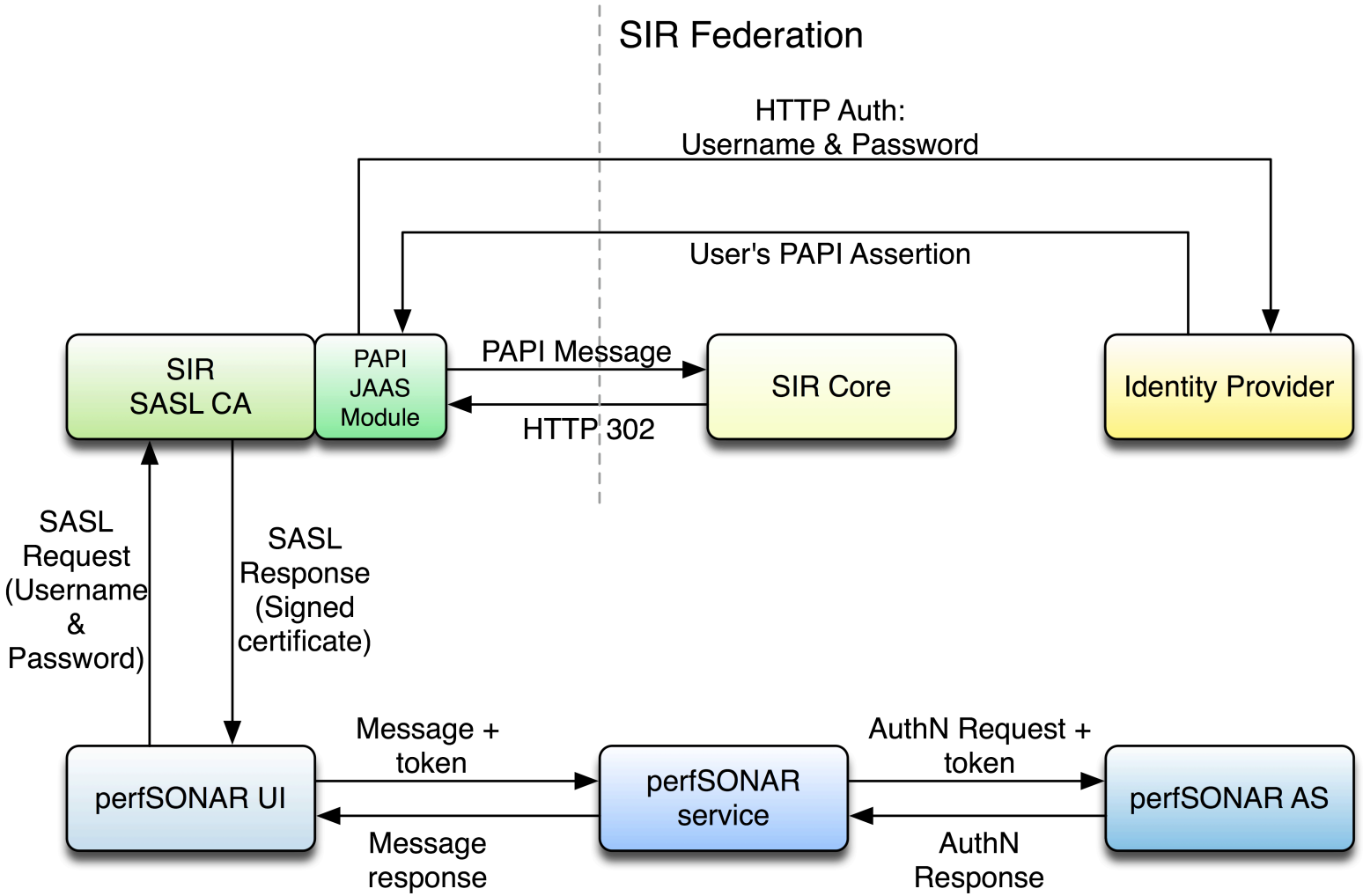


Cliente OAuth2

- Datos de la federación
  - Cómo conseguir y procesar los datos de identidad
- Datos OAuth2
  - Cómo acceder el AS y el servidor
- Datos del recurso
  - Puestos a disposición de la aplicación cliente



Certificados: SLCS



Protocolos propietarios

- Todos los que hemos implementado hasta ahora están basados en tecnologías bien conocidas
  - Estilo REST, como TPS
  - Estilo SOAP, como MS-Live
  - Redirecciones y cabeceras, como Facebook o AppleStore
- Implementar conectores SIR no ha sido una tarea compleja

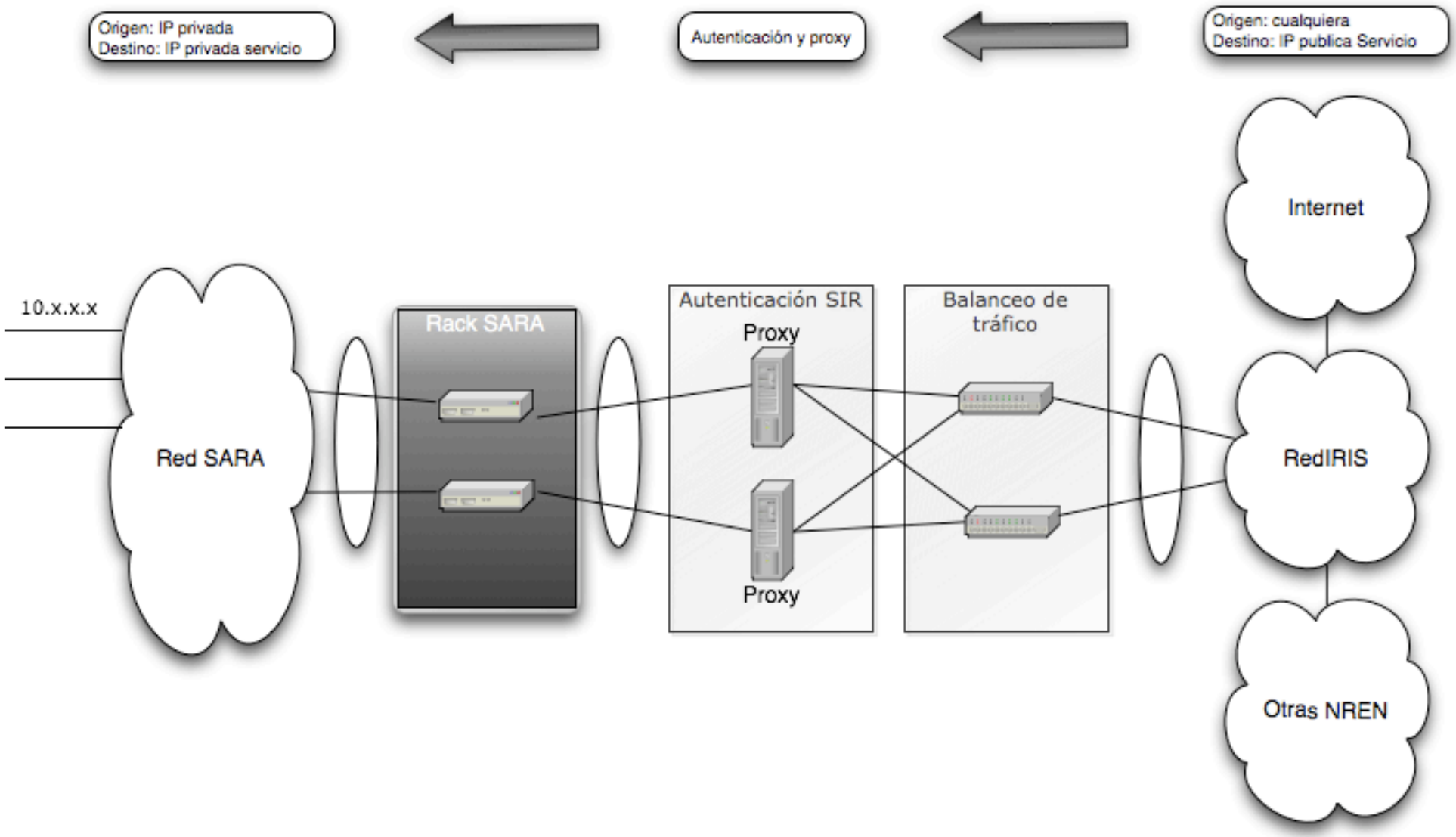
La navaja suiza

- Los proxies engloban varios conceptos que pueden proporcionar soluciones a problemas diversos
- Simplificar la gestión de accesos
  - Directo: soporte a virtualización de servicios
  - Inverso: firewall de servicios
- Accesos a servicios legacy
  - La aplicación más común
  - No sólo basados en IP
- Incrementar la privacidad
  - Una IP/password/certificado compartidos no son datos tan personales

PAPI::RewritingProxy

- Soporta diferentes métodos de acceso
  - Dirección IP
  - Autenticación HTTP (Basic y Digest)
  - Certificados de cliente
- Maneja atributos del usuario
  - Para gestionar el acceso
  - Para aplicar las reglas de reescritura
- Versión actual muy estable
  - Casi 10 años en uso
- Trabajamos en una versión mejorada
  - Usando bucket-brigades de Apache2

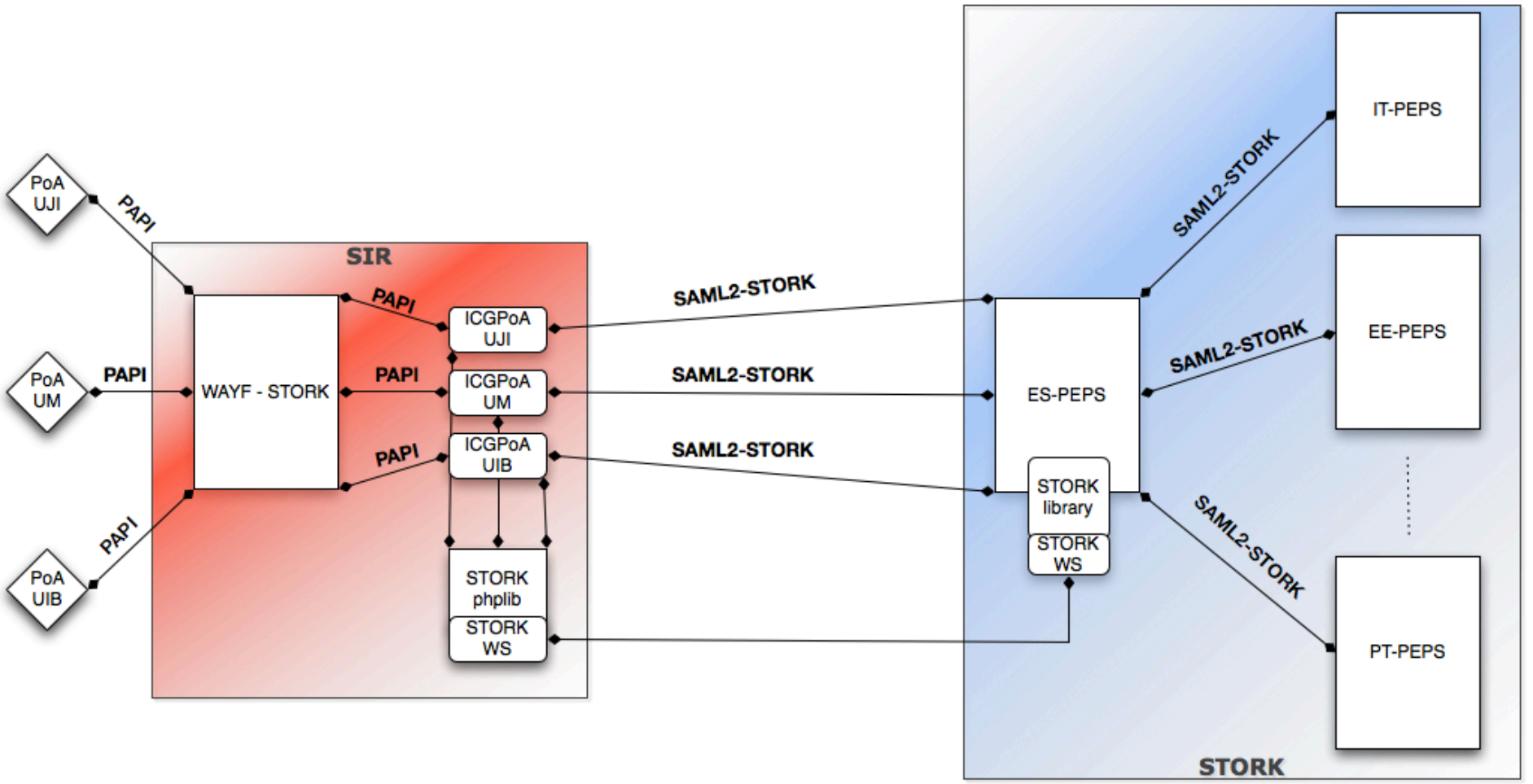
Proxies: SARA en SIR



El otro lado

- SIR es completamente neutral frente a sus fuentes de datos
- Los conectores de entrada son incluso más fáciles de integrar
- Una cuestión del LoA
  - Pruebas con Facebook, Twitter, Google, MS-Live...
  - STORK como fuente de datos de alta fiabilidad
  - Casos de uso complementarios
- Autoridades de atributos
  - La siguientre frontera en integración de fuentes de datos
  - El LoA es incluso más relevante aquí

STORK en SIR


PoA UJI
PoA UM
PoA UIB
PAPI
PAPI
PAPI
SIR
WAYF - STORK
PAPI
PAPI
PAPI
ICGPoA UJI
ICGPoA UM
ICGPoA UIB
STORK phplib
STORK WS
SAML2-STORK
SAML2-STORK
SAML2-STORK
ES-PEPS
STORK library
STORK WS
SAML2-STORK
SAML2-STORK
SAML2-STORK
IT-PEPS
EE-PEPS
PT-PEPS
STORK

Conclusiones

- El modelo SIR permite una integración sencilla de:
  - La identidad institucional en los mecanismos de aceso a todo tipo de servicios externalizados
  - Servicios diversos usando un esquema trazable, homogéneo y que preserva la privacidad
  - Fuentes de datos externas con diversos grados de fiabilidad
- Y ofrece garantías de
  - Evolución tecnológica
  - Integración de aplicaciones y servicios legacy

¿Cómo no vamos a poner a SIR por las nubes?