Manual de integración

SINGLE SIGN ON (SSO)

Versión v1.0

Fecha 6/11/19

ANII AGENCIA NACIONAL DE INVESTIGACIÓN E INNOVACIÓN

Big Cheese

ÍNDICE

Esta guía informativa está dirigida a desarrolladores de aplicaciones; y proporciona una descripción general de los roles de OAuth 2, tipos de autorización, casos de uso y flujos.

SOBRE SINGLE SIGN ON

El Single Sign On (SSO) es una solución que permite simplificar el uso diario de los sistemas y aplicaciones por parte de los usuarios, simplifica la administración de credenciales y permite incrementar los niveles de seguridad. El SSO permite al usuario acceder a las aplicaciones que estén configuradas, identificándose una única vez para acceder a las aplicaciones. Existen varios estándares para SSO en este manual de integración se cubre OIDC.

TERMINOLOGÍA

OIDC

Open Id Connect (OIDC) es una capa de identidad sobre el protocolo OAuth 2.0, el cual permite a los clientes verificar la identidad de un usuario basado en la autenticación realizada por un servidor de autorización, así como para obtener información de perfil del usuario utilizando un esquema REST. El estándar OIDC de SSO esta basado en los principios de OAuth 2.0.

OAUTH 2.0

Es una estructura (framework) de autorización que le permite a las aplicaciones obtener acceso limitado a cuentas de usuario en un servicio HTTP. Delega la autenticación del usuario al servicio que aloja la cuenta del mismo y autoriza a las aplicaciones de terceros el acceso a dicha cuenta de usuario.

ROLES

OAuth define cuatro roles:

- Usuario, Cliente, Servidor de recursos, Servidor de autorización

USUARIO

El propietario del recurso es el "usuario" que da la autorización a una aplicación, para acceder a su cuenta. El acceso de la aplicación a la cuenta del usuario se limita al "alcance" de la autorización otorgada (e.g. acceso de lectura o escritura).

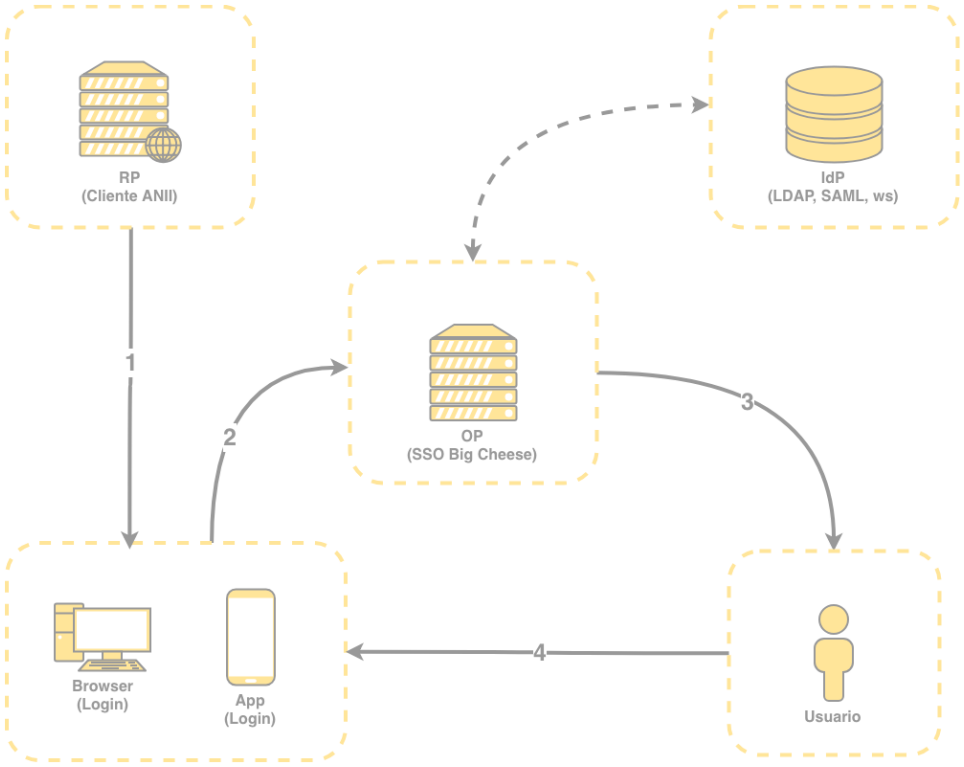
SERVIDOR DE RECURSOS

El servidor de recursos aloja las cuentas de usuario protegidas, y el servidor de autorizaciones verifica la identidad del usuario y luego genera tokens de acceso a la aplicación.
Desde el punto de vista del desarrollador de una aplicación, la API del servicio atiende tanto a los roles de recursos como a los de autorización. Nos referiremos a ambos roles combinados, como al rol de servicio o de API.

CLIENTE

El cliente es la aplicación que desea acceder a la cuenta del usuario. Antes de que pueda hacerlo, debe ser autorizado por el usuario, y dicha autorización debe ser validada por la API.

INFRAESTRUCTURA BASE



Esquemáticamente la infraestructura base cuenta con:

- RP: Relaying Party.
- OP: OpenID Connect Provider.
- IdP: Identity Provider.

A modo general:

1. El cliente RP redirecciona el user agent al formulario de autenticación.
2. Este formulario de ingreso interactúa con el servidor de OP.
3. En la interacción se le solicita al usuario otorgar el permiso mediante su cuenta SSO almacenada en el IdP.
4. Si el ingreso es correcto el OP redirecciona al recurso.

Para esta interacción el flujo base corresponde al estándar OAuth 2.0 donde:

a. Se solicita un permiso de autorización al servidor mediante un tipo de autorización OIDC.
b. Este retorna un access token y un refresh token para que el cliente pueda utilizar.
c. Utilizando el access token el cliente solicita el recurso que desea acceder.
d. El servidor devuelve el recurso protegido al cliente para que este lo pueda utilizar.
e. Si el cliente desea reutilizar el token para otro o para el mismo recurso.
f. El servidor responde que el token ya no es valido,
g. Utilizando el refresh token se solicita un nuevo access token.
h. El servidor le da un nuevo access token y un nuevo refresh token para una próxima oportunidad

El flujo real de este proceso variará dependiendo del tipo autorización (response type) que esté en uso, sin embargo, esta es la idea general. Examinaremos diferentes tipos de autorizaciones en una sección posterior.

RECURSOS DISPONIBLES

ÚltIma versión de este manual de integración:

- http://docs.sso.bigcheese.uy/manual.pdf

Colección Postman con ejemplos de uso:

- http://docs.sso.bigcheese.uy/postman.json

Documentación correspondiente a cada servicio mencionado en esta guía:

- https://api.sso.bigcheese.uy

Cliente OP para integración:

- https://sso.bigcheese.uy/

Cliente RP simplificado que ayuda a probar los flujos básicos de integración:

- https://test.sso.bigcheese.uy

Su código fuente se puede obtener de:

- https://github.com/n0v4c4n3/oidc-debugger

Cliente RP avanzado adicional:

- https://sso.bigcheese.uy/oxauth-rp/

Cliente de LDAP web (privado):

- https://ldap-sso.bigcheese.uy

SOLICITUD DE REGISTRO DE CLIENTE

Antes de utilizar el SSO de ANII, se debe registrar la aplicación con el servicio.
Esto se hace a través de un pedido de registro que se debe completar con algunos datos por parte del "desarrollador" o "API" para integrar el cliente con el servicio SSO, en el cual se debe proporcionar a ANII la siguiente información:

- Nombre de la aplicación.
- Sitio web de la aplicación.
- Redirect URI o Callback URL.
- Detalles adicionales sobre la aplicación

Redirect URI es donde el servicio redirecciona al usuario después de que se autorice (o deniegue) su solicitud y, por consiguiente, la parte de su aplicación que manejará códigos de autorización o tokens de acceso.
Identificador del cliente y secreto de cliente

Una vez esté registrada tu aplicación, el servicio emitirá "credenciales del cliente" en forma de un identificador de cliente y un secreto de cliente. El identificador (ID) de cliente es una cadena pública que utiliza la API de servicio para identificar la aplicación y para generar las URL de autorización que se presentan a los usuarios.
Una vez la aplicación solicita el acceso a la cuenta de un usuario, el secreto de cliente se utiliza para autenticar la identidad de la aplicación al API de servicio; y se deberá mantener la confidencialidad entre la aplicación y la API.

FLUJOS DE AUTORIZACIÓN

En el diagrama "base" presentado anteriormente, los primeros cuatro pasos abarcan la obtención de una autorización y el token de acceso.

El tipo de otorgamiento de la autorización depende del método utilizado por la aplicación para solicitar dicha autorización y de los tipos de autorización soportados por la API. Definimos cuatro tipos de autorización, cada uno de los cuales es útil para distintos casos de uso:

AUTHORIZATION FLOW:

- Flujo de código de autorización.
- Caso de uso server side apps.
- Utilizado por aplicaciones que cuentan con lado del servidor.

IMPLICIT FLOW

- Flujo implícito de autorización.
- Caso de uso single page apps.
- Utilizado por aplicaciones que se ejecutan en el dispositivo del usuario.

RESOURCE OWNER FLOW

- Flujo de credenciales de contraseña del propietario del recurso.
- Caso de uso apps nativas móviles.
- Utilizado por aplicaciones confiables, como aquellas pertenecientes al servicio.

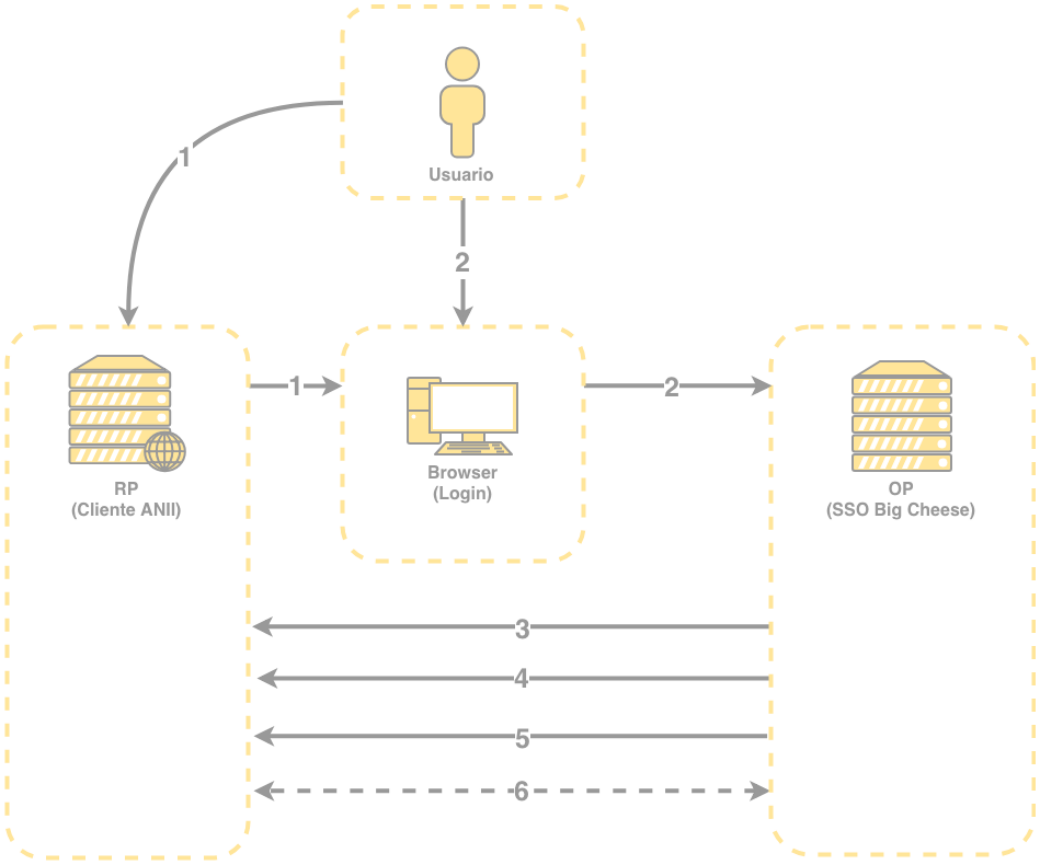
CLIENT CREDENTIALS FLOW

- Flujo de credenciales del cliente.
- Caso de uso recursos y servicios.
- Utilizadas por acceso API de aplicaciones.

En las siguiente secciones describiremos con mayor detalle los tipos de otorgamiento, sus casos de uso y flujos.

AUTHORIZATION FLOW

El tipo de otorgamiento más usado es el código de autorización, ya que ha sido optimizado para aplicaciones del lado del servidor, en donde el código fuente no está expuesto públicamente y se puede mantener la confidencialidad del secreto de cliente. Este es un flujo basado en la redirección, que significa que la aplicación debe ser capaz de interactuar con el agente de usuario (i.e. el navegador web del usuario) y recibir códigos de autorización API que se enrutan a través del agente de usuario.

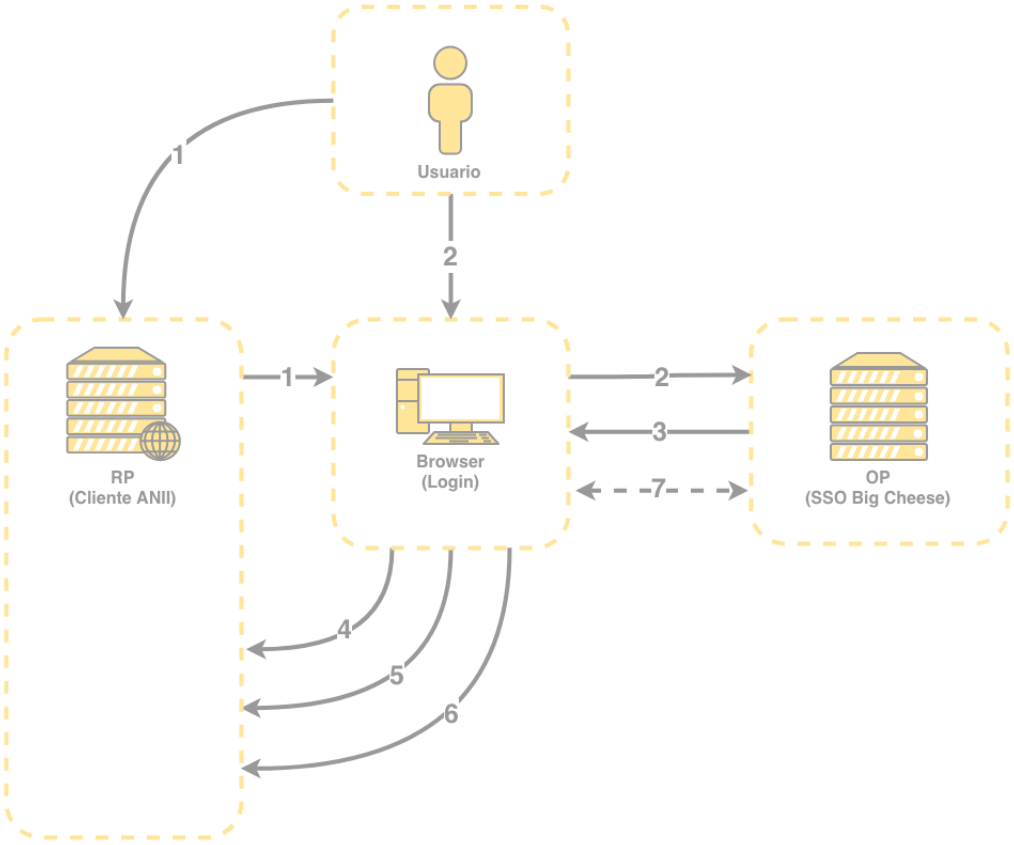


Descripción del flujo:

1. Enlace de código de autorización
2. El usuario autoriza a la aplicación
3. La aplicación recibe el código de autorización
4. La aplicación solicita token de acceso
5. La aplicación recibe el token de acceso
6. La aplicación renueva el token

IMPLICIT FLOW

El tipo de otorgamiento implícito se utiliza para aplicaciones móviles y aplicaciones web (i.e. aplicaciones que se ejecutan en un navegador web), donde no se garantiza la confidencialidad del secreto de cliente. El tipo de otorgamiento implícito también es un flujo basado en la redirección, pero el token de acceso se entrega al agente-usuario para reenviarlo a la aplicación, por lo que puede estar expuesto al usuario y a otras aplicaciones en el dispositivo del usuario. Además, este flujo no autentica la identidad de la aplicación y depende del redirect URI (que se registró con el servicio) para cumplir este propósito.

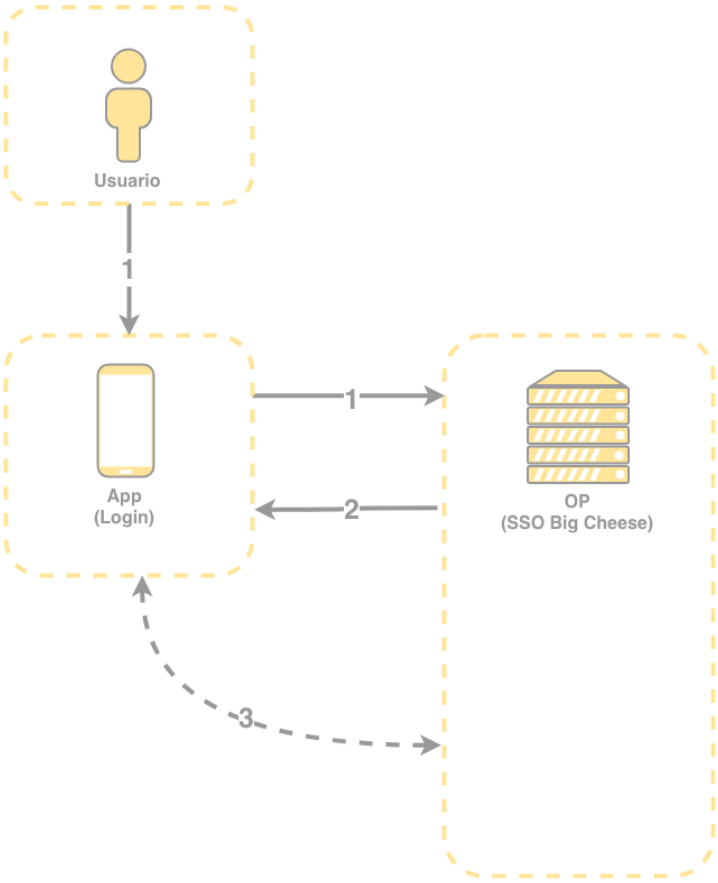


Descripción del flujo:

1. Enlace de autorización implícita.
2. El usuario autoriza a la aplicación.
3. El agente-usuario recibe el token de acceso con Redirect URI.
4. El agente-usuario sigue al Redirect URI.
5. La aplicación envía el script de extracción de tokens de acceso.
6. Token de acceso transferido a la aplicación.
7. La aplicación renueva el token.

RESOURCE OWNER FLOW

Con el tipo de otorgamiento de credenciales de contraseña del propietario del recurso, el usuario proporciona sus credenciales de servicio (nombre de usuario y contraseña) directamente a la aplicación, la cual utiliza dichas credenciales para obtener del servicio un token de acceso. Este tipo de autorización solo debe habilitarse en el servidor de autorización, si otros flujos no son viables. Además, solo debe utilizarse si la aplicación es confiable para el usuario (e.g. es propiedad del servicio o del sistema operativo de escritorio del usuario).
Flujo de credenciales de contraseña Después de que el usuario proporcione sus credenciales a la aplicación, ésta solicitará un token de acceso desde el servidor de autorizaciones. Si se comprueban las credenciales del usuario, el servidor de autorización devuelve un token de acceso a la aplicación.

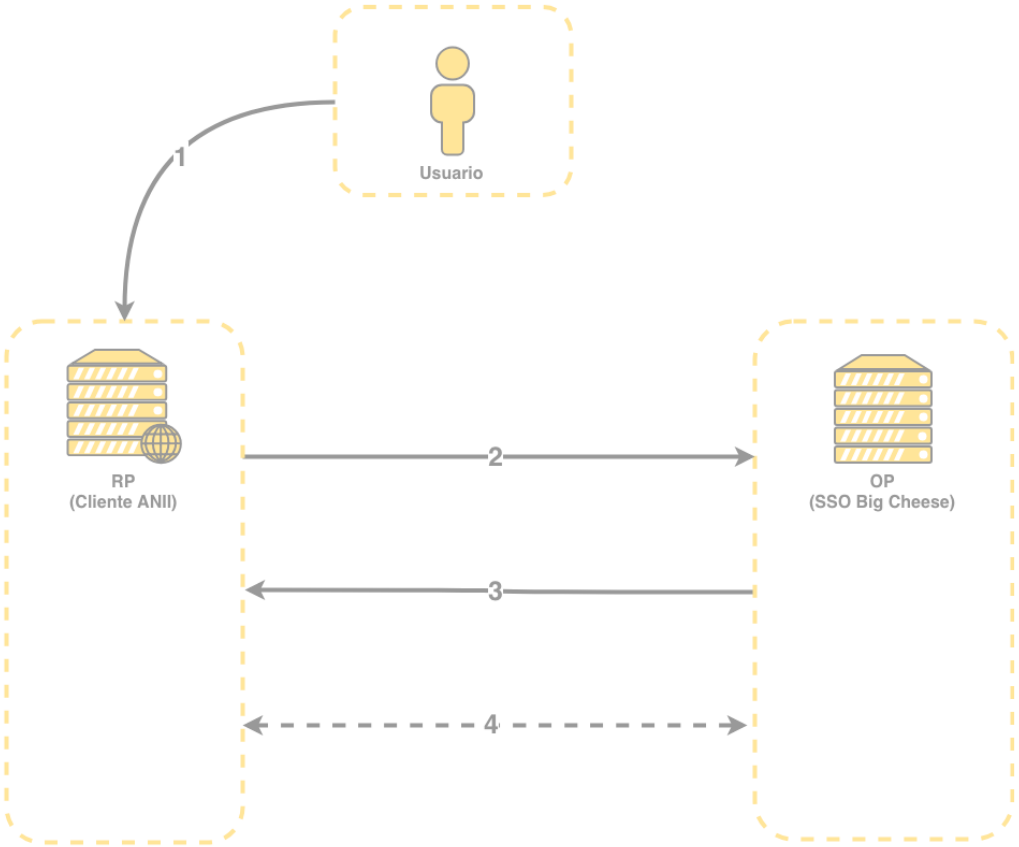


Descripción del flujo:

1. El usuario autoriza a la aplicación.
2. Token de acceso transferido a la aplicación.
3. La aplicación renueva el token.

CLIENT CREDENTIALS FLOW

El tipo de otorgamiento de credenciales del cliente proporciona a la aplicación una forma de acceder a su propia cuenta de servicio. Si una aplicación desea actualizar su descripción registrada o redirigir el URI, o acceder a otros datos almacenados en su cuenta de servicio a través de la API serían ejemplos de cuándo podría ser útil este tipo de otorgamiento.



Descripción del flujo:

1. El usuario o servicio solicita un recurso.
2. La aplicación envía los key y secret.
3. Token de acceso transferido a la aplicación.
4. La aplicación renueva el token.

ACTUALIZACIÓN DE TOKEN

Hacer una solicitud desde el API, utilizando un token de acceso que ha caducado, generará un error de token inválido "Invalid Token Error".

Si cuando se emitió el token de acceso original se incluyó un token de actualización, entonces éste puede ser usado para solicitar un token de acceso nuevo desde el servidor de autorización.

MANEJO DE STATE

Sobre como manejar el estado del usuario entre redirecciones de SSO.
Esto sirve para por ejemplo, retornar el usuario a una estado inicial hecho el llamado de callback al SSO.

Para contemplar esta funcionalidad el estándar define el state.

- https://openid.net/specs/openid-connect-core-1_0.html

state:RECOMMENDED. Opaque value used to maintain state between the request and the callback. Typically, Cross-Site Request Forgery (CSRF, XSRF) mitigation is done by cryptographically binding the value of this parameter with a browser cookie.

CUYOS USOS PUEDEN SER:

1. Almacenar un token de anti-falsificación
2. Almacenar un token de sesión para redirección

En pocas palabras el state es un estado, un valor opcional que se transmite a través de todo el flujo y se devuelve al cliente. Es común usar el estado para almacenar un token anti-falsificación que puede verificarse después de que se completa el flujo de inicio de sesión o almacenar la ubicación a la que se debe redirigir al usuario después de iniciar sesión.

Para nuestro flujo, es uno de los parámetros al generar la URL de redirección al SSO.

EJEMPLO

- https://sso.bigcheese.uy/oxauth/restv1/authorize?
  response_type=token&response_mode=query&client_id=@%2107BC.
  46B9.96E3.A008%210001%216F88.5A6A%210008%2146B3.0091.3CCD.3A42&redirect_uri=https://
  test.sso.bigcheese.uy/debug&state=8566c90b-c747-4b8b-9f3f-65b706ebe5c2&nonce= s6zqwux2w4s

Según el estándar el SSO al autenticarse el usuario responde retornado el mismo:

- [redirecturl]/code=b7b2c893-b82a-4c66-
  a0bb-5f73e848c32d&scope=openid+email&session_id=4fe7eb43-3ba9-400a-
  ad40-7678d2cd4b30&state=8566c90b-
  c747-4b8b-9f3f-65b706ebe5c2&session_state=aec61a4f7c0f96fd31c750d76f097198a40d0f3f34aa5e6fe61d
  6597c6c8435a.1f980310-60fb-4263-b37c-e9e49f510962

FLUJO DE EJEMPLO:

Este comportamiento se puede aprovechar de la siguiente manera:

1. El usuario ingresar al cliente
2. El cliente genera un identificador de sesión
3. El usuario realiza alguna acción que requiera SSO y devolver al estado previo
4. El cliente asocia la acción al identificador de sesión
5. Se pasa como parametro en el "state"
6. El usuario se autentica
7. El SSO redirecciona manteniendo el state
8. El cliente retoma ese "state"
9. Se busca la acción realizada por el usuario por medio del state guardado
10. Se envía al usuario a dicha acción
11. El usuario continua su historia

CONCLUSIÓN

Utilizando la documentación provista para cada servicio REST, este manual de integración y los datos de cliente es posible implementar un sistema de ingreso único que permita unificar en un punto central la gestión de los usuarios mejorando la seguridad de los recursos.

Con esto concluye la guía de integración de SSO ANII Big Cheese.