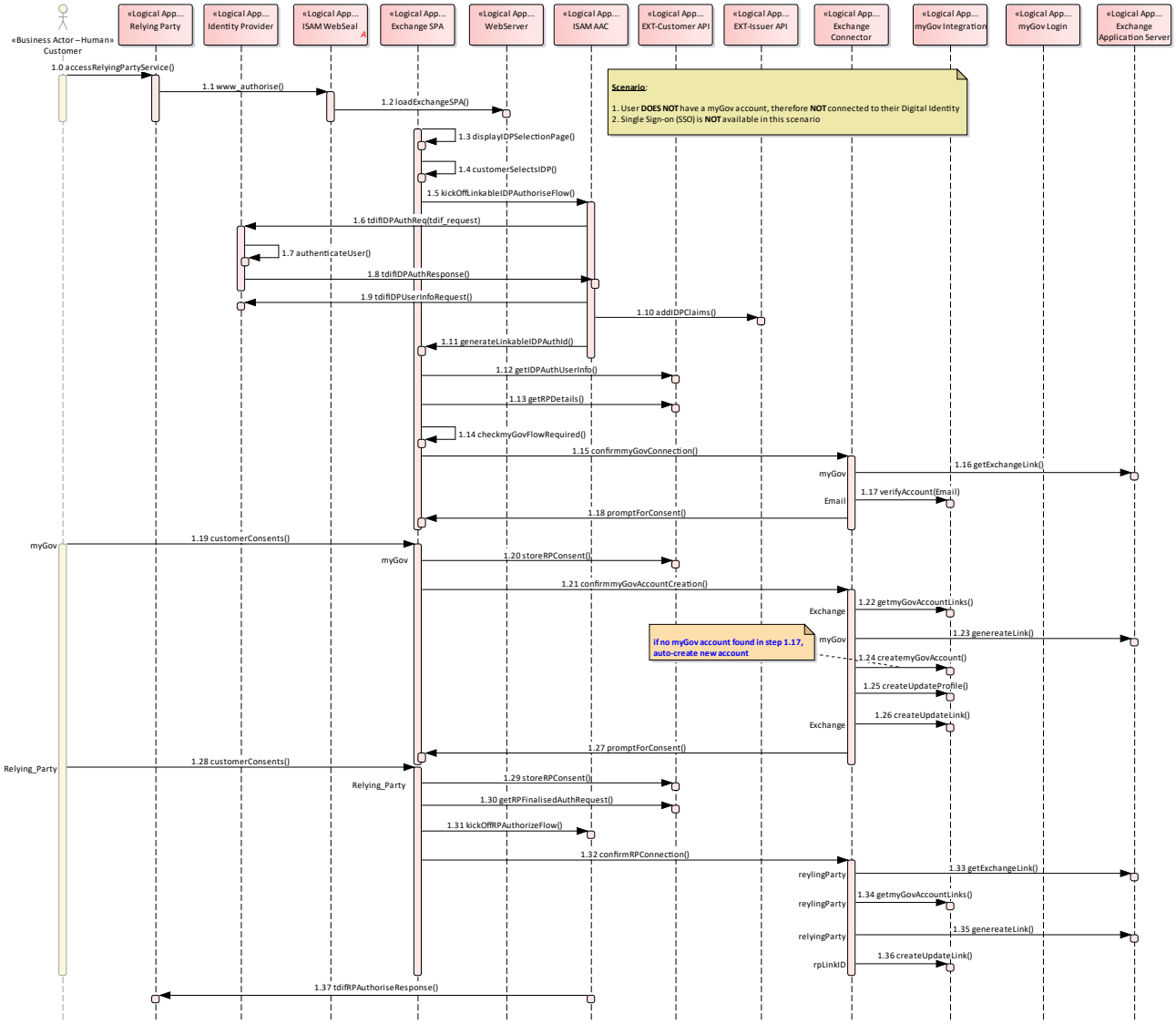
# S3 - No myGov - No Exchange - No SSO

Ram Challuri

14/11/2019 3:33:48 PM



### 1.0 - accessRelyingPartyService() : (Customer ==> Relying Party)
**Description:**
Customer Requesting to Access Relying Service.

### 1.1 - www_authorise() : (Relying Party ==> ISAM WebSeal)
**Description: OIDC**
Relying party issues an authorize request with the appropriate scope and claims

**Request:**
https://<<idhub>>/authorise?*response_type*=code&*client_id*=rp&*scope*=openid profile email phone tdif_business_authorisations&*redirect_uri*=<<hostname>>/rp/exchange&*nonce*=<<nonce>>&*state*=<<state>>&*acr_values*=urn:id.gov.au:tdif:acr:ip2:cl2&*claims*={"id_token":{"mygov_linked":{"essential":true}},"userinfo":{"mygov_linked":{"essential":true}}}

**Response:**
HTTP/1.1 200 OK

### 1.2 - loadExchangeSPA() : (ISAM WebSeal ==> WebServer)
**Description:**
Load Exchange SPA page

### 1.3 - displayIDPSelectionPage() : (Exchange SPA ==> Exchange SPA)
**Description: UI action**
Display the IDP Selection page to the user

### 1.4 - customerSelectsIDP() : (Exchange SPA ==> Exchange SPA)
**Description: User action**
Customer selects the preferred IDP from the list

### 1.5 - kickOffLinkableIDPAuthoriseFlow() : (Exchange SPA ==> ISAM AAC)
**Description: API**
Kickoff the IDP authorize flow

**Security:**
Bearer-Token: **ISAM_LEVEL1_JWT**

**Request:**
**POST**: /sso/sps/oidc/rp/IDENTITYHUB/kickoff/MYGOVIDP?acr_values=<<acr_values>>

**Response:**
HTTP/1.1 200 OK

**BODY:**
{"location":"/mga/sps/oauth/oauth20/authorize?scope=openid+email+link
&state=<<state>>
&client_id=rp_exchange_client
&nonce=<<nonce>>
&redirect_uri=/sso/sps/oidc/rp/IDENTITYHUB/redirect/MYGOVIDP
&acr_values=<<acr_values>>
&response_type=code"}

### 1.6 - tdifIDPAuthReq(tdif_request) : (ISAM AAC ==> Identity Provider)
**Description: API**
Send TDIF OIDC Authorize request to Selected IDP as a front channel redirect.

**Security:**
**ISAM_LEVEL1_JWT**

**Request:**
**GET:**
/proto-idp/oauth/authorize?claims=<<claims>>&state=<<state>>&client_id=PROTOIDP&nonce=<<nonce>>&redirect_uri=https://<<idhub>>/sso/sps/oidc/rp/IDENTITYHUB/redirect/PROTOIDP&acr_values=<<acr_values>>&response_type=code

**Response:**
HTTP/1.1 302 Found

### 1.7 - authenticateUser() : (Identity Provider ==> Identity Provider)
**Description:**
Authenticate the IDP User.

### 1.8 - tdifIDPAuthResponse() : (Identity Provider ==> ISAM AAC)
**Description:**
TDIF IDP Authorisation response

**Security:**
**ISAM_LEVEL1_JWT**

**Request:**
**GET**: /sso/sps/oidc/rp/IDENTITYHUB/redirect/PROTOIDP?code=<<code>>&state=<<state>>

**Response:**
HTTP/1.1 302
**Redirect from IDP**

### 1.9 - tdifIDPUserInfoRequest() : (ISAM AAC ==> Identity Provider)

### 1.10 - addIDPClaims() : (ISAM AAC ==> EXT-Issuer API)

**Description: API**
Store IDP claims in the database

**Security:**
**Header**: ex-ext-iss-api-key: <<**ISAM_EXCHANGE_JWT**>>

**Request:**
**PUT**: <<exchange>>/api-ext-issuer/v1/identityproviders/<<idp_client_id>>/authclaims

**BODY:**

```
{"claims":{"sub":"<<sub>>",
 "aud":"PROTOIDP",
 "acr":"urn:id.gov.au:tdif:acr:ip2:cl2",
 "auth_time":1564557530,
 "kid":"1",
 "iss":"https://<<idhub>>/proto-idp",
 "exp":<<exp>>,
 "iat":<<iat>>,
 "nonce":"<<nonce>>",
 "jti":"<<jti>>",
 "phone_number":"000",
 "family_name":"Mayweather",
 "email_verified":true,
 "phone_number_verified":true,
 "updateAt":1564557531302,
 "email":"test@email.com",
 "given_name":"Mike",
 "birthdate":"<<birthdate>>"}}
```


**Response:**
HTTP/1.1 200 OK

### 1.11 - generateLinkableIDPAuthId() : (ISAM AAC ==> Exchange SPA)

Generate **IDP_AUTH_ID** for the linkable flow

### 1.12 - getIDPAuthUserInfo() : (Exchange SPA ==> EXT-Customer API)

**Description**: API
Retrieve IDP Auth User Information

**Security:**
Bearer-Token: **ISAM_LEVEL3_JWT**

**Request:**
**GET**: <<idhub>>/api-ext-customer-ui/v1/idpauthentications/<<idp_auth_id>>/<<rp_auth_id>>/userinfo

**Response:**
HTTP/1.1 200 OK
**BODY:**

```
{"idpAuthId":"<<idp_auth_id>>","sessionId":"<<session_id>>","claims":[{"claims"}]}
```


### 1.13 - getRPDetails() : (Exchange SPA ==> EXT-Customer API)

**Description: API**
Retrieve details for a given relying party

**Security:**
Bearer-Token: **ISAM_LEVEL3_JWT**

**Request:**
**GET**: <<idhub>>/api-ext-customer-ui/v1/relyingparty/7

**Response:**
HTTP/1.1 200 OK

**BODY:**

```
{"id":"rp_biz_client",
 "name":"DSS GRS",
 "description":"Dept Social Services Grants Registration System",
 "properties":{"displayName":"Grants Registration Portal",
 "serviceName":"Grants Registration Portal"}}
```


### 1.14 - checkmyGovFlowRequired() : (Exchange SPA ==> Exchange SPA)

check if {"mygov_linked":{"essential":true}} is present in the Relying Party Authorisation request

### 1.15 - confirmmyGovConnection() : (Exchange SPA ==> Exchange Connector)

**Description: Internal API**

Verify and confirm if a link exists between myGov and Exchange for the given user

### 1.16 - getExchangeLink(string) : (Exchange Connector ==> Exchange Application Server)

**Description: API**
Verify if myGov is linked to Exchange

### 1.17 - verifyAccount() : (Exchange Connector ==> myGov Integration)

**Description: API**

Verify if an account exists in myGov for the given user identifier (email_address)

### 1.18 - promptForConsent() : (Exchange Connector ==> Exchange SPA)

Redirect to SPA to display the consent page to obtain user's consent for myGov auto-account creation and sharing details with myGov

### 1.19 - customerConsents() : (Customer ==> Exchange SPA)

**Description: User Action**
Customer provides the consent to share details to myGov

### 1.20 - storeRPConsent() : (Exchange SPA ==> EXT-Customer API)

**Description: API**
Store Relying Party Consent

**Security:**
Bearer-Token: **ISAM_LEVEL3_JWT**

**Request:**
**PUT:** <<idhub>>/api-ext-customer-ui/v1/rpauthentications/<<rp_auth_id>>/consent

**BODY:**
```
{"id":"<<rp_auth_id>>",
 "claimConsents":[],
 "abn":"83134235310",
 "triggerScope":<<triggerScope>>}
```

**Response:**
HTTP/1.1 201 Created

### 1.21 - confirmmyGovAccountCreation() : (Exchange SPA ==> Exchange Connector)

### 1.22 - getmyGovAccountLinks() : (Exchange Connector ==> myGov Integration)

**Description: API**
Verify if a link exists between myGov and the Relying Party (e.g. Exchange)

**Security:**
Bearer-Token: **EXCHANGE_MYGOV_JWT**

**Request:**
**GET:** /accounts/links
**PARAM:** relyingPartyId

**Response:**
HTTP/1.1 200 OK

**BODY:** RelyingPartyLink object
```
{
 "relyingPartyId": "ATO|EXCHGE|...",
 "relyingPartyName": "Australian Taxation Office|myGov Identity Hub|...",
 "relyingPartyLinkDetails": {
   "id": "<<id>>",
   "status": "permanent|transient",
   "created": "<<created>>",
   "lastModified": "<<lastModified>>"
 }
}
```

### 1.23 - genereateLink(relyingParty) : (Exchange Connector ==> Exchange Application Server)

Generate myGov link in Exchange

### 1.24 - createmyGovAccount() : (Exchange Connector ==> myGov Integration)

**Description: API**
Create a new myGov account for the User in the myGov system

**Security:**
Bearer-Token: **EXCHANGE_MYGOV_JWT**

**Request:**
**POST (Create):** /accounts/

**BODY: myGovAccount object**
```
{ "myGovAccount ": "<<myGovAccount>>"}
```

**Response:**
HTTP/1.1 201: Successfully created

### 1.25 - createUpdateProfile() : (Exchange Connector ==> myGov Integration)

**Description: API**
Create or Update User's myGov profile in myGov

**Security:**
Bearer-Token: **EXCHANGE_MYGOV_JWT**

**Request:**
**POST (Create):** /accounts/profile
**PUT (Update):** /accounts/profile

**BODY: myGov Profile**
```
{ "name": {
   "firstName": "string",
   "middleName": "string",
   "lastName": "string"
 },
 "dateOfBirth": "2019-11-05"
}
```

**Response:**
HTTP/1.1 204 Successfully Created or Updated

### 1.26 - createUpdateLink(rpLinkID) : (Exchange Connector ==> myGov Integration)
**Description: API**
Create a new myGov Relying Party Account Link / MBUN in the myGov system for a given relying party. The user account will be specified by the use of the "sub", "squ" and "sty" claims of the mgv-jwt JWT header

**Security:**
Bearer-Token: **EXCHANGE_MYGOV_JWT**

**Request:**
**POST (Create):** /accounts/links/

**BODY:** RelyingPartyLink object
```
{
 "relyingPartyId": "ATO|EXCHGE|...",
 "relyingPartyName": "Australian Taxation Office,
 "relyingPartyLinkDetails": {
   "id": "<<id>>",
   "status": "permanent|transient",
   "created": "<<created>>",
   "lastModified": "<<lastModified>>"
 }
}
```

**Response:**
HTTP/1.1 201 Relying party link successfully created

### 1.27 - promptForConsent() : (Exchange Connector ==> Exchange SPA)
Redirect to SPA to display the consent page

### 1.28 - customerConsents() : (Customer  ==> Exchange SPA)
**Description: User Action**
Customer provides the consent to share details to the relying party

### 1.29 - storeRPConsent() : (Exchange SPA ==> EXT-Customer API)
**Description: API**
Store Relying Party Consent

**Security:**
Bearer-Token: **ISAM_LEVEL3_JWT**

**Request:**
**PUT:** <<idhub>>/api-ext-customer-ui/v1/rpauthentications/<<rp_auth_id>>/consent

**BODY:**
```
{"id":"<<rp_auth_id>>",
 "claimConsents":[],
 "abn":"83134235310",
 "triggerScope":<<triggerScope>>}
```

**Response:**
HTTP/1.1 201 Created

### 1.30 - getRPFinalisedAuthRequest() : (Exchange SPA ==> EXT-Customer API)
**Description: API**
Retrieve Relying Party Finalised Auth Request

**Security: ISAM_LEVEL3_JWT**

**Request:**
**GET:** <<idhub>>/api-ext-customer-ui/v1/rpauthentications/<<rp_auth_id>>/finalised

**Response:**
HTTP/1.1 200 OK

**BODY:**
```
{"rpAuthId":"<<rp_auth_id>>",
 "fullURL":"<<fullURL>>",
 "sessionId":"<<session_id>>"}
```

### 1.31 - kickOffRPAuthorizeFlow() : (Exchange SPA ==> ISAM AAC)

**Description: API**
Invoke this API to kickoff the RP Authorise flow

**Security:**
Bearer-Token: **ISAM_LEVEL3_JWT**

**Request:**
**POST:**
<<idhub>>/sso/sps/oauth/oauth20/authorize?response_type=code&client_id=rp_biz_client&scope=<,scope>>&redirect_uri=<<redirect_uri>>&state=<,state>>&acr_values=<<acr_values>>&prompt=none

**Response:**
{"location":"https://<<idhub>>/proto-rp/exchange?state=<<state>>&code=<<code>>"}

### 1.32 - confirmRPConnection() : (Exchange SPA ==> Exchange Connector)

**Description: Internal API**

Verify and confirm if a link exists between the Relying Party and Exchange for the given user

### 1.33 - getExchangeLink(reylingParty) : (Exchange Connector ==> Exchange Application Server)

**Description: API**
Verify if the RP link exists in Exchange

### 1.34 - getmyGovAccountLinks() : (Exchange Connector ==> myGov Integration)

**Description: API**
Verify if a link exists between myGov and the Relying Party (e.g. Exchange)

**Security:**
Bearer-Token: **EXCHANGE_MYGOV_JWT**

**Request:**
**GET:** /accounts/links
**PARAM:** relyingPartyId

**Response:**
HTTP/1.1 200 OK

**BODY:** RelyingPartyLink object

```
{
 "relyingPartyId": "ATO|EXCHGE|...",
 "relyingPartyName": "Australian Taxation Office|myGov Identity Hub|...",
 "relyingPartyLinkDetails": {
   "id": "<<id>>",
   "status": "permanent|transient",
   "created": "<<created>>",
   "lastModified": "<<lastModified>>"
 }
}
```

### 1.35 - genereateLink(relyingParty) : (Exchange Connector ==> Exchange Application Server)

Generate relying party pairwise identifier in Exchange

### 1.36 - createUpdateLink(rpLinkID) : (Exchange Connector ==> myGov Integration)

**Description: API**
Create a new myGov Relying Party Account Link / MBUN in the myGov system for a given relying party. The user account will be specified by the use of the "sub", "squ" and "sty" claims of the mgv-jwt JWT header

**Security:**
Bearer-Token: **EXCHANGE_MYGOV_JWT**

**Request:**
**POST (Create):** /accounts/links/

**BODY:** RelyingPartyLink object

```
{
 "relyingPartyId": "ATO|EXCHGE|...",
 "relyingPartyName": "Australian Taxation Office,
 "relyingPartyLinkDetails": {
   "id": "<<id>>",
   "status": "permanent|transient",
   "created": "<<created>>",
   "lastModified": "<<lastModified>>"
 }
}
```

**Response:**
HTTP/1.1 201 Relying party link successfully created

### 1.37 - tdifRPAuthoriseResponse() : (ISAM AAC ==> Relying Party)

Send an authorized response back to the Relying Party along with the RP claims