數位服務個人化

服務提供者技術文件

V1.2

國家發展委員會

中華民國 108 年 5 月

版本修正紀錄

項次	版本	修正內容	時間	頁次
1	1.1	調整章節「陸、一、流程示意圖」。 調整章節「柒、二、MyData-API 請求及回覆規格說明」之內容。	108/05/11	P15 P16
2	1.2	增加章節「柒、MyData 整合方式說明」 調整章節捌為「捌、SP-API Endpoint 規格說明」 調整章節玖為「玖、MyData-API Endpoint 規格說明」	108/05/20	P17 P20 P23

目錄

壹、目的

本文件主要描述扮演「MyData 平臺之服務提供者」時應依循的作業流程、準則及相關注意事項。

貳、如何成為服務提供者

一、完成 MyData 服務提供者資格申請作業

機關單位如欲加入 MyData 成為「服務提供者」，需先完成資格申請。內容細節請參考本文件章節「肆、服務提供者資格申請作業」。

二、完成 MyData 線上註冊作業

註冊作業包括：

- 機關單位註冊：登錄機關單位基本資料。
- 服務註冊：登錄服務資訊。

內容細節請參考本文件章節「伍、服務提供者管理作業」

三、實作 SP-API 開發，提供予 MyData 平臺介接

服務提供者必需提供 SP-API Endpoint 並登錄於 MyData 管理後台。
MyData 平臺將利用此 SP-API Endpoint，於用戶同意授權後，將 permission_ticket 及 secret_key 傳送予服務提供者。

四、實作 MyData-API 之系統整合介接

MyData 平臺提供 MyData-API Endpoint 讓服務提供者可以透過此 API 取得該服務所需的用戶個人資料打包檔案。MyData 平臺的 MyData-API 的回應格式是 JWT，是將用戶個人資料打包檔案封裝於 JWT 中。
因此服務提供者需了解 MyData-API 及資料打包檔的相關規格，以完成 MyData-API 的整合介接工作。

參、名詞定義

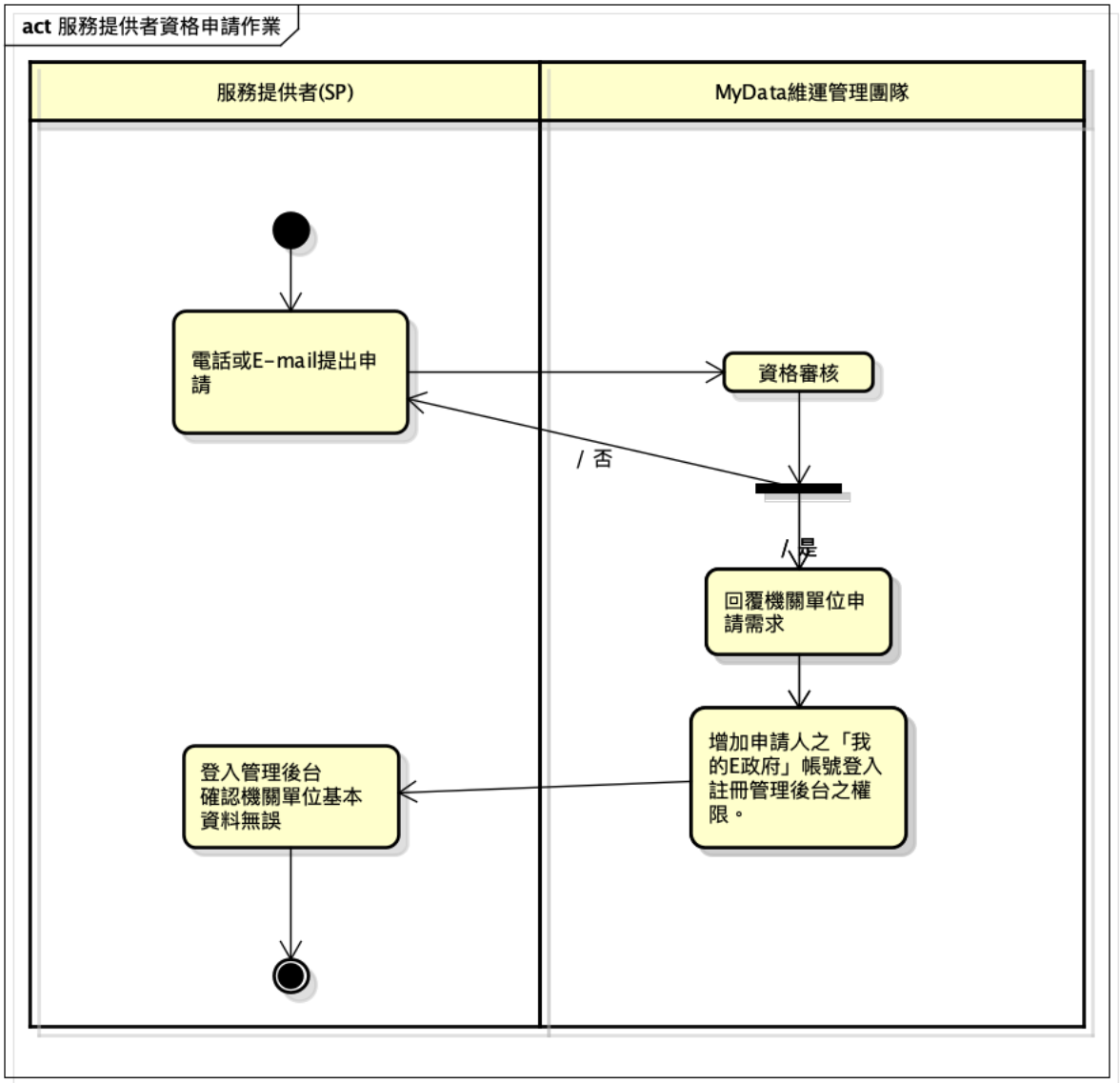
名稱	定義
OAS	共通性應用程式介面規範。

Data Provider, DP	資料提供者，存放或保管民眾個人資料之機關單位。
Service Provider, SP	服務提供者，提供民眾進行個人資料之加值服務機關單位。
Authorization Server, AS	授權管理者，執行身分驗證與授權管理機制，本規範之授權管理者為本會政府服務平臺(GSP)。
Resource Owner, RO	資料擁有者/使用者，泛指用戶或民眾。
GSP	電子化政府服務平臺
OAuth 2.0	系統授權流程規範，定義於 RFC 6749 The OAuth 2.0 Authorization Framework https://tools.ietf.org/html/rfc6749
OpenID Connect	OAuth 2.0 的補充規範，強調身分驗證流程 http://openid.net/connect/
eGov 帳號	我的 E 政府提供的會員帳號
access_token	AS 發的用戶同意授權

肆、服務提供者資格申請作業

機關單位欲成為 MyData 服務提供者角色，應先完成資格申請，步驟說明如下：

步驟項次	流程內容
1	機關單位以電話或 E-mail 聯繫管理團隊，提出 MyData 管理後台使用權限申請。 （聯絡資訊 Tel:02-86925588#5555, E-mail:mydata@ndc.gov.tw）
2	管理團隊回覆機關單位申請需求，增加機關單位申請人之「我的 E 政府」帳號登入管理後台之權限（不限於公務帳號）。
3	機關單位申請人以「我的 E 政府」帳號登入管理後台並確認機關單位基本資料無誤後，使用服務提供者管理功能項目。
4	機關單位成為服務提供者，使用相關功能註冊服務。



機關單位以電話（號碼）、電子郵件（信箱）聯繫 MyData 維運團隊申辦註冊管理後台機關帳號，並於申辦時提供「機關單位名稱」及「機關單位地址」、「申請人姓名」、「聯絡電話」、「電子郵件信箱」與申請人之「我的 E 政府註冊帳號」，由 MyData 維運人員協助完成帳號註冊作業。完成機關單位註冊後，MyData 維運人員將透過註冊時機關單位提供之「聯絡電話」及「電子郵件信箱」通知機關單位聯絡人。

伍、服務提供者管理作業

一、基本資料編輯

機關單位登入管理平臺後，點選「機關單位管理」功能項目，可自行編輯機關單位基本資料，包含「聯絡人姓名」、「聯絡電話」、「聯絡 E-mail」、「E 政府帳號」（管理後台登入使用）。於此功能頁面中，可瀏覽目前機關單位已建立之資料集與加值服務項目。

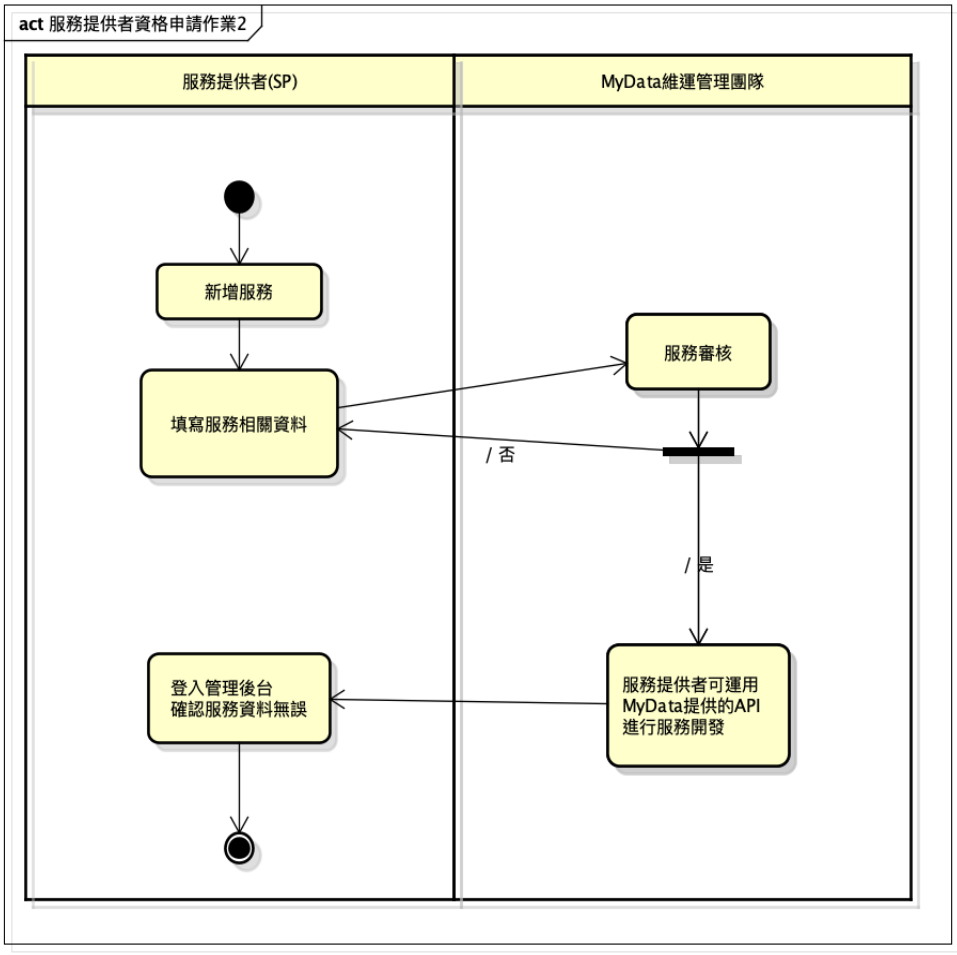
機關單位登入管理平臺後，點選「服務提供者管理」功能項目，若尚未同意「服務提供者合作契約」內容，需先同意契約內容成為服務提供者身分後，即可開通服務提供者相關功能使用權限（服務新增、編輯與管理功能）。

二、服務註冊

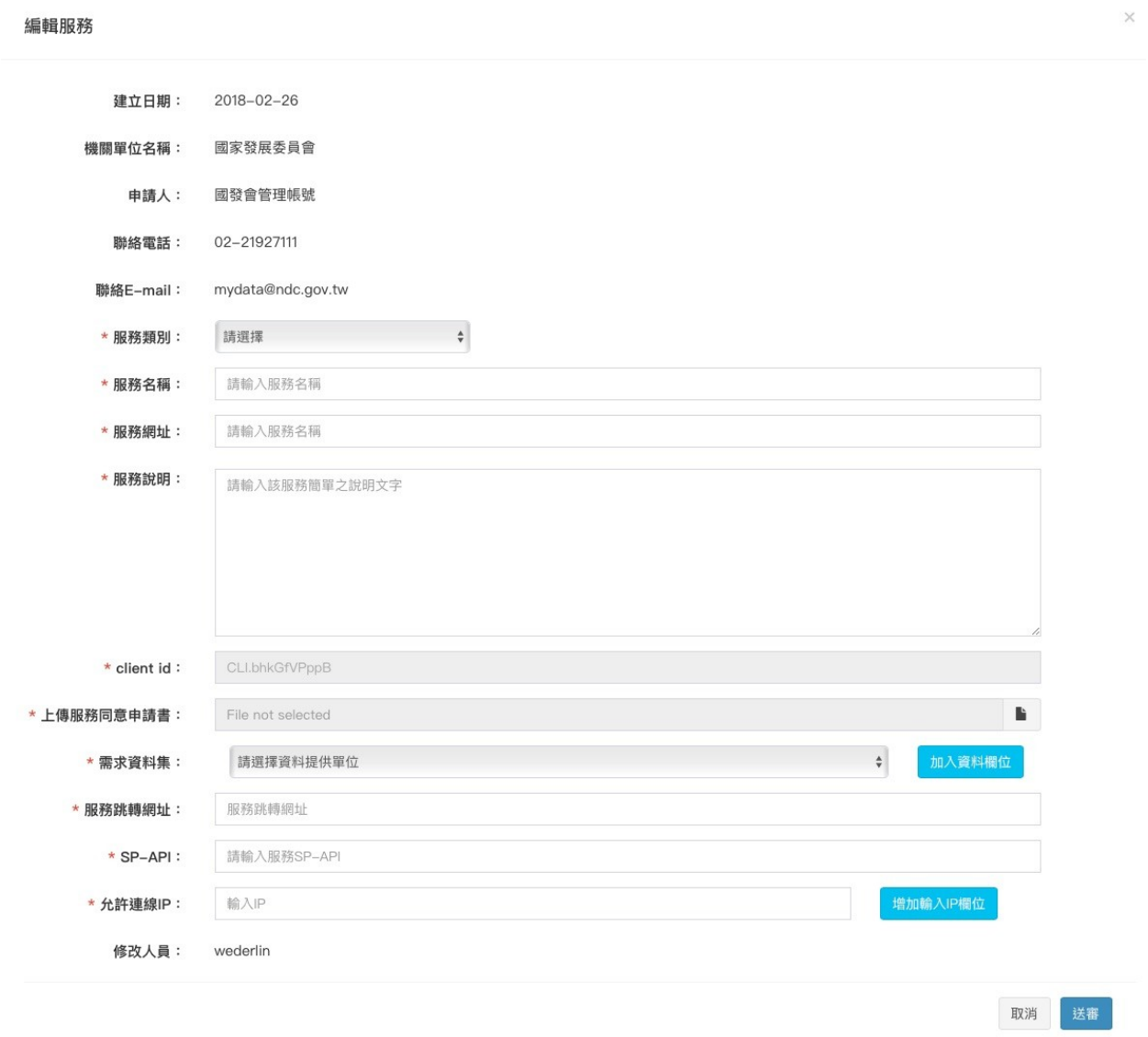
步驟項次	流程內容
1	服務提供者使用 MyData 管理後台「新增服務」功能。

2	完成新增服務頁面中相關欄位內容，並提交審核。
3	經維運管理團隊確認相關內容符合 MyData 規範後，通知服務單位（申請人）審核通過。若審核未通過，將回覆服務單位（申請人）修正建議，服務單位依修正建議調整後可重新提交審核申請。
4	依 MyData 之技術規範完成 SP-API 開發實作及 MyData-API 整合介接工作。

新增服務	
基本資料欄位	申請日期、單位名稱、申請人姓名、聯絡電話、聯絡電子郵件信箱、服務類別、服務名稱、服務說明、client_id，上傳服務同意申請書，需求資料集、服務跳轉網址、SP-API 網址，允許連線 IP。
選填欄位	
功能動作	選擇資料集、上傳服務同意申請書、取消、送審。



欄位介面示意：



欄位序號	欄位名稱	說明
1	client_id	註冊新服務時，MyData 系統會產生用以識別服務的唯一的識別值 client_id。當 SP 網站重導向至 MyData 整合網址時，須帶入 client_id 於 path parameter 中。

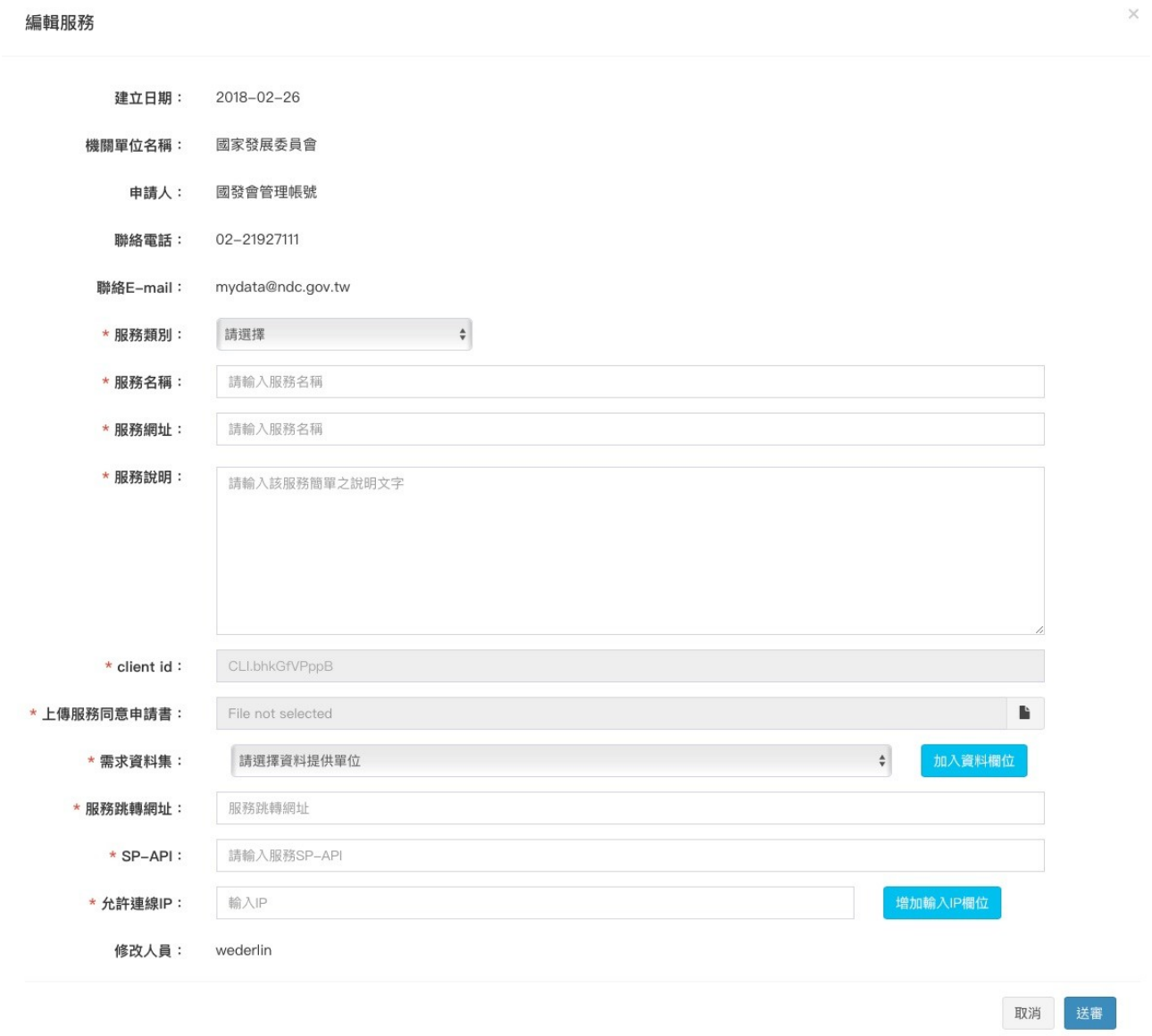
2	允許連線 IP	允許連線 IP 的設定用於 SP 呼叫 MyData-API 時，MyData 系統用以過濾請求來源。允許連線 IP 可以設定多筆。

三、服務管理

顯示已註冊、申請中之服務項目清單，並提供關鍵字查詢與新增、修改、刪除功能。已送審或上架之服務無法修改服務內容。刪除功能僅限服務狀態為停用者，服務狀態為啟用者應先完成停用申請流程。

修改服務	
基本資料欄位	申請日期、單位名稱、申請人姓名、聯絡電話、聯絡電子郵件信箱、服務類別、服務名稱、服務說明、client_id，上傳服務同意申請書，需求資料集、服務跳轉網址、SP-API 網址，允許連線 IP。
選填欄位	
功能動作	選擇資料集、上傳服務同意申請書、取消、送審。

欄位介面示意：



四、可運用的資料集

服務提供者檢視 MyData 已註冊資料提供者與資料集清單時，可使用「查詢列表 / 所有資料集列表」功能項目，將以清單顯示資料提供者與相對應資料集名稱，並提供依資料提供者篩選資料及 API 識別值、資料集名稱關鍵字搜尋功能。

欄位序號	欄位名稱	說明

1	resource_id	系統自動建立之識別碼
2	資料集名稱	資料提供者註冊之資料集名稱
3	SCOPE	OAuth2 資源 SCOPE 值
4	需要的身分驗證安全等級	資料集要求的授權身分驗證等級
5	資料提供機關單位名稱	資料提供者名稱

欄位介面示意：

五、異常管理

顯示 SP 服務狀態現況，內容如下：

服務狀態列表	
基本資料欄位	項次、申請日期、機關單位名稱、服務類別、服務名稱、狀態。
功能動作	查看、前往服務頁。

欄位介面示意：

異常管理

陸、MyData 整合協作流程說明

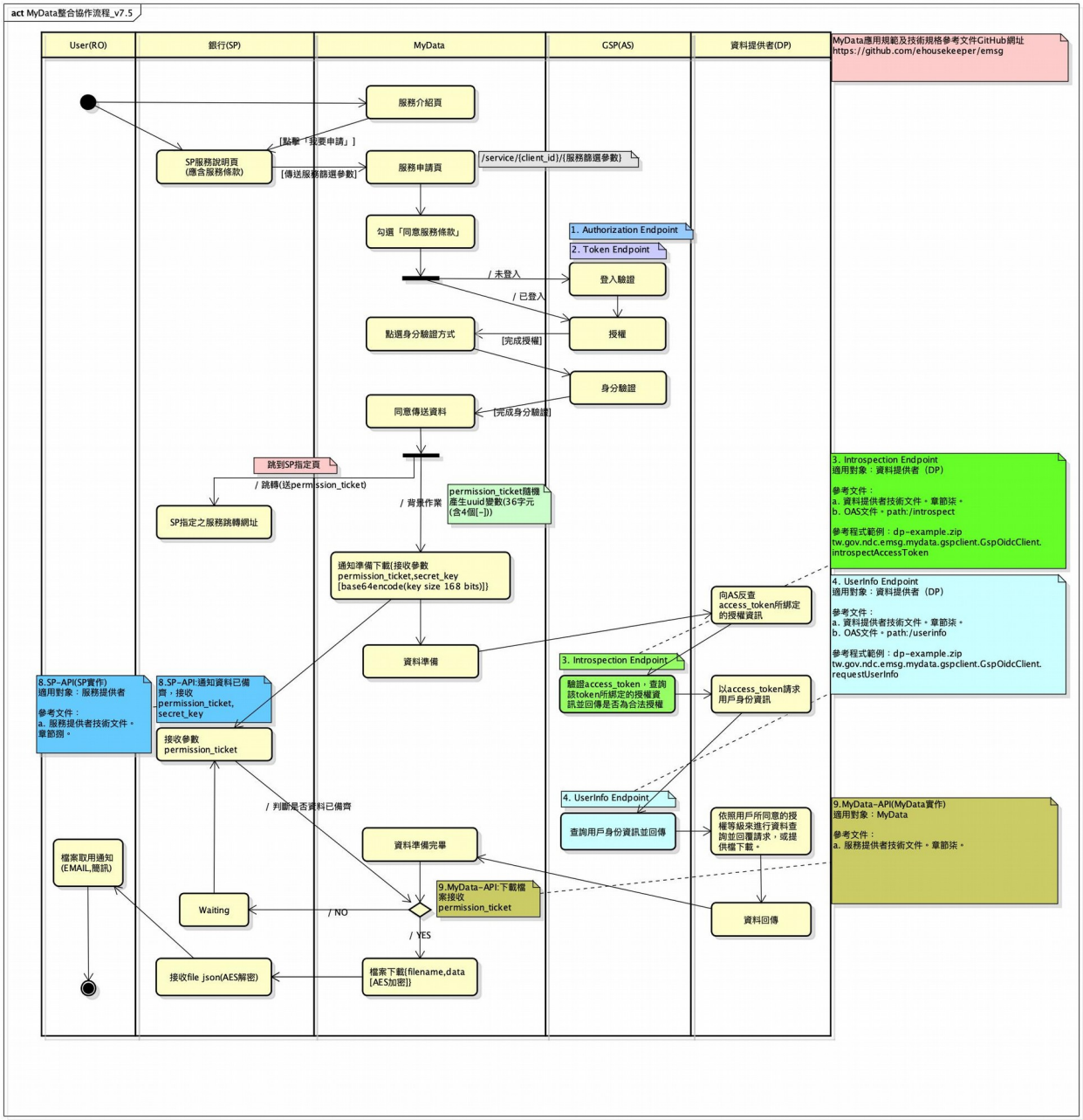
MyData 平臺的身分驗證及授權機制，採整合介接 GSP 所提供的 OAuth2 身分驗證及授權主機，民眾用戶使用我的 E 政府帳號進行登入驗證，且需同意授權 MyData 平臺向資料提供者取得用戶自己的個人資料。MyData 平臺則會將代表該用戶已同意授權的憑據 access_token 傳遞給資料提供者，讓資料提供者可依此向 GSP OAuth2 授權主機發出請求，以檢核用戶是否已同意授權及取得用戶資訊。

GSP OAuth2 授權主機之身分驗證及授權機制符合 OpenID Connect (OIDC) 規範。OIDC 是基於 RFC 6749 The OAuth 2.0 Authorization Framework 標準之上的一種 OAuth 2.0 協議。它使客戶端可以根據授權服務器執行的身分驗證來驗證最終用戶的身分，及以可互操作和 REST 的方式獲取有關最終用戶的基本配置文件信息。

本文件主要對象為提供資料提供者參考，為避免混淆，僅著重描述服務提供者何時請求呼叫以下 API：

- SP-API
- MyData-API

一、流程示意圖


act MyData整合協作流程_v7.5
User(RO)
銀行(SP)
MyData
GSP(AS)
資料提供者(DP)
MyData應用規範及技術規格參考文件GitHub網址
https://github.com/ehousekeeper/emsg
服務介紹頁
[點擊「我要申請」]
SP服務說明頁
(隱含服務條款)
[傳送服務篩選參數]
服務申請頁
/service/{client_id}/{服務篩選參數}
勾選「同意服務條款」
1. Authorization Endpoint
2. Token Endpoint
/ 未登入
登入驗證
/ 已登入
點選身分驗證方式
[完成授權]
授權
身分驗證
同意傳送資料
[完成身分驗證]
跳到SP指定頁
/ 跳轉(送permission_ticket)
SP指定之服務跳轉網址
/ 背景作業
permission_ticket隨機
產生uuid變數(36字元
(含4個[-]))
通知準備下載(接收參數
permission_ticket,secret_key
[base64encode(key size 168 bits)])
資料準備
3. Introspection Endpoint
3. Introspection Endpoint
適用對象：資料提供者（DP）
參考文件：
a. 資料提供者技術文件，章節柒，
b. OAS文件，path:/introspect
參考程式範例：dp-example.zip
tw.gov.ndc.emsg.mydata.gspclient.GspOidcClient.
introspectAccessToken
4. UserInfo Endpoint
適用對象：資料提供者（DP）
參考文件：
a. 資料提供者技術文件，章節柒，
b. OAS文件，path:/userinfo
參考程式範例：dp-example.zip
tw.gov.ndc.emsg.mydata.gspclient.GspOidcClient.
requestUserInfo
向AS反查
access_token所綁定
的授權資訊
驗證access_token，查詢
該token所綁定的授權資
訊並回傳是否為合法授權
以access_token請求
用戶身份資訊
8.SP-API(SP實作)
適用對象：服務提供者
參考文件：
a. 服務提供者技術文件，
章節捌。
8.SP-API:通知資料已備
齊，接收
permission_ticket,
secret_key
接收參數
permission_ticket
/ 判斷是否資料已備齊
4. UserInfo Endpoint
查詢用戶身份資訊並回傳
依照用戶所同意的授
權等級來進行資料查
詢並回覆請求，或提
供檔下載。
9.MyData-API(MyData實作)
適用對象：MyData
參考文件：
a. 服務提供者技術文件，章節柒。
資料準備完畢
檔案取用通知
(EMAIL,簡訊)
9.MyData-API:下載檔
案接收
permission_ticket
Waiting
/ NO
/ YES
資料回傳
檔案下載(filename,data
[AES加密])
接收file json(AES解密)

二、應用範圍

（一）規範資料集下載通知格式

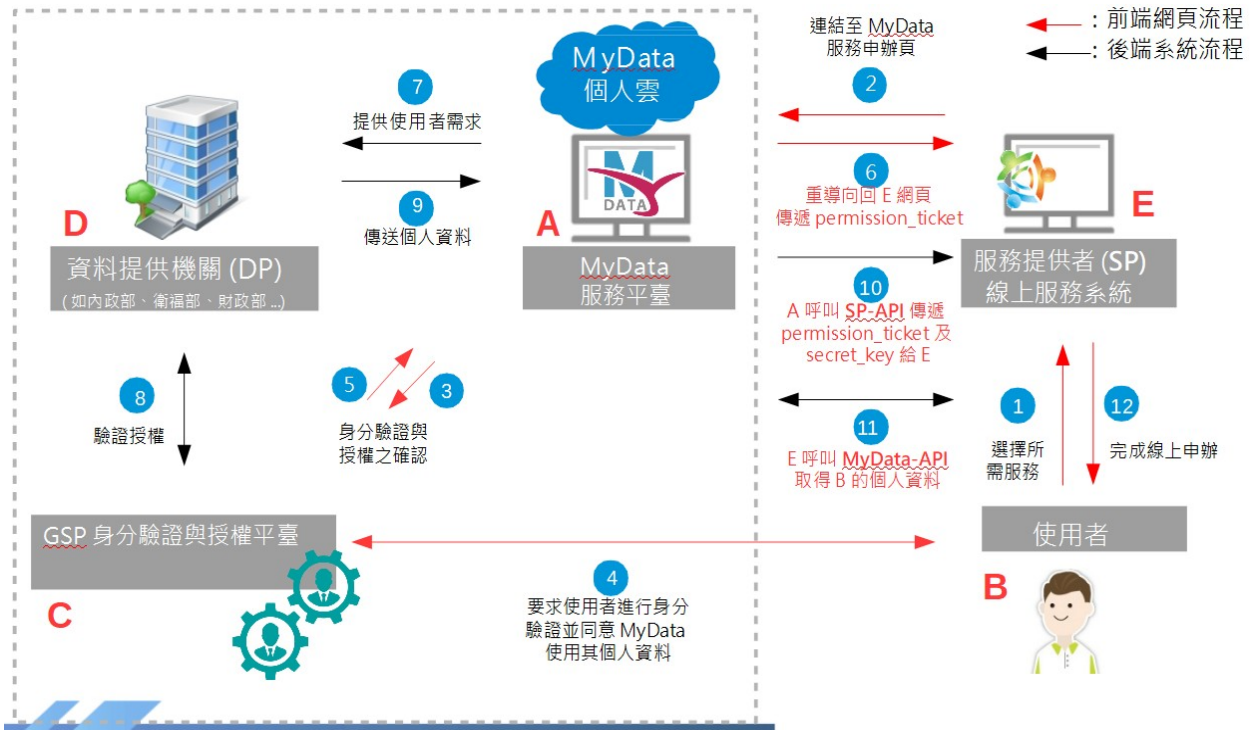
服務提供者應實作資料集下載通知 API(SP-API)，此 API 為 MyData 通知有使用者允許下載的資料集打包壓縮 zip 檔，給予 permission_ticket 和 secret_key，以 permission_ticket 下載資料，並以 secret_key 解密資料檔案及驗證 JWT 簽章，內容細節請參考本文件章節「捌、SP-API Endpoint 規格說明」。

（二）資料集下載

服務提供者取得用戶同意授權的資料集檔案。內容細節請參考本文件章節「玖、MyData-API Endpoint 規格說明」。

柒、MyData 整合方式說明

一、服務情境示意圖



步驟 1, 2, 3, 4, 5, 6, 11 為前端網頁流程。其它則為後端系統流程。

二、MyData 整合網址及參數說明

上述服務情境示意圖中步驟 2，由 SP 網站導向 MyData 整合網址時以 Path Parameter 帶入所需參數，示意如下：

```
GET /service/{client_id}/{resource_id_base64encoded_string}?
returnUrl={sp_return_url}
```

參數說明如下：

參數	說明
client_id	SP 於 MyData 管理後台新增服務後所得的 client 識別值。
resource_id_base64encoded_string	Base64Encode 編碼後的 DP 資料集識別值。若需多個 DP 資料集可以:符號分隔各別的資料集識別值。 示意如下： Base64Encode({resource_id1}:{resource_id2})
sp_return_url	SP 載明，令 MyData 處理完身分認證及同意授權後，重導向回到 SP 網站的網址。 同時 SP 也須將這個返回網址登錄於 MyData 管理後台中。MyData 會依據管理後台中的返回網址設定來判斷此參數值是否合法。 MyData 的判斷原則為只判斷 url path 是否相同，但不會判斷 request parameter 是否完全相同。因此 SP 可視實際需要附加其它的 request parameter，MyData 將不會移除任何 SP 原本附加的 request parameter，以

	方便 SP 系統後續處理。 此參數值必須以 UrlEncode 編碼處理過。

當 SP 令瀏覽器導向至上述 MyData 整合網址後，MyData 以 client_id 識別 SP 為誰，以 resource_id_base64encoded_string 識別 SP 欲請求的 DP 資料集有那些。

MyData 系統會檢核 SP 所請求的 DP 資料集，是否符合 SP 申請服務時所載明的 DP 資料集項目。

三、正常返回 SP 網址之處理方式說明

```
GET /{sp_return_url}?permission_ticket={permission_ticket}
or
GET /{sp_return_url}?
sp_param=abc&permission_ticket={permission_ticket}
```

當 MyData 處理完成 SP 請求後，會核發一個只於該次交易有效的唯一識別值（permission_ticket），格式為 version4 UUID，長度共 36 字元的字符串。

參數說明如下：

參數	說明
permission_ticket	代表該次用戶同意授權的交易識別碼。 格式為 version 4 UUID 字符串。 單次有效且唯一不重覆。 有效時間最長 24 小時。
sp_return_url	SP 的返回網址。 sp_param=abc 用於示意表示 SP 原本附加的參數，MyData 將原值返回。

四、異常返回 SP 網址之處理方式說明

當 MyData 無法處理或拒絕處理來自 SP 的請求，或發現參數檢核失敗時，MyData 會將異常狀態碼，以 code 參數附加於 sp_return_url 網址上重導向回 SP 網站，以利 SP 後續處理作業。

網址示意如下：

```
GET /{sp_return_url}?code={code}
or
GET /{sp_return_url}?sp_param=abc&code={code}
```

參數說明如下：

參數	說明
code	異常狀態碼。 400：無法順利解析 SP 帶入的 path parameter。 401：無效的 client_id 或 resource_id。 403：sp_return_url 不符合 MyData 管理後台中所登錄的設定。 404：MyData 判斷 SP 所請求的 resoruce_id 不在該 SP 服務申請的 DP 資料集項目中。 501：SP 請求的 DP 資料集之系統已停止服務。 504：SP 請求的 DP 資料集之系統異常，目前無法順利連接 DP 系統以取得資料檔案。
sp_return_url	SP 的返回網址。 sp_param=abc 用於示意表示 SP 原本附加的參數，MyData 將原值返回。

五、無法返回 SP 網址之處理方式說明

若因網路問題或其它不可控因素，導致 MyData 無法令瀏覽器順利返回 SP 網址時，SP 應視該交易為無效交易。

捌、SP-API Endpoint 規格說明

一、系統環境與條件

API endpoint 以 RESTful Service 方式提供介面，且皆基於 TLS v1.1 以上提供加密傳輸管道。

二、SP-API 請求及回覆規格說明(由服務提供者實作)

服務提供者必須實作 API，以接收 MyData 平臺傳送來的，使用者於平臺授權資料的加密鍵值。以利後續服務端抓取資料及將資料解密使用。

（一）MyData 發出請求 - 告知 SP 準備來捉取資料檔

```
POST /mydata-sp/notification HTTP/1.1
Host: xxx.xxx.xxx.xx
Content-Type: application/json

{
  permission_ticket: {uuid_v4_string},
  secret_key: {256bit_random_string}
}
```

欄位說明：

欄位	說明
permission_ticket	MyData 核發，代表該次用戶同意授權的交易識別碼。 格式為 version 4 UUID 字符串。 單次有效且唯一不重覆。 有效時間最長 24 小時。

secret_key	只於該次交易有效的金鑰。隨機產生的英數字含大小寫的字符串，長度為 256bit, 32bytes. 產製 JWT 簽章時使用的金鑰。請參考章節玖、三、MyData-API 的回傳格式與加簽機制說明。 加密 MyData 資料打包檔時使用的金鑰。請參考章節玖、四、MyData-API 資料解密方法說明。

（二）MyData 發出請求 - 告知 SP 無法給予資料檔

```
POST /mydata-sp/notification HTTP/1.1
Host: xxx.xxx.xxx.xx
Content-Type: application/json

{
  permission_ticket: {uuid_v4_string},
  unable_to_deliver: [
    {resource_id1},{resource_id2}
  ]
}
```

欄位說明：

欄位	說明
permission_ticket	MyData 核發，代表該次用戶同意授權的交易識別碼。 格式為 version 4 UUID 字符串。 單次有效且唯一不重覆。 有效時間最長 24 小時。
unable_to_deliver	MyData 已確認無法取得的 DP 資料集。 例：若該次交易 SP 請求的 DP 資料集共有 3 個，但只有其中 1 個已確定無法傳遞時，此欄位值只會載明已確定無法傳遞的 DP 資料集識別值共 1 個，但仍會以陣列的方式表述。

當 SP 請求的 DP 資料集中有任何一個資料集無法順利傳遞時，MyData 即視為該筆交易為失敗。

當以下情況發生時，MyData 將無順利傳遞 DP 資料集予 SP：

1. MyData 向 DP 發出請求成功後，等候逾時仍無法取得資料檔案。
2. MyData 向 DP 發出請求失敗。

（三）SP 回覆請求成功

```
HTTP/1.1 200 OK
Content-Type: application/json
```

（四）SP 回覆請求失敗

```
HTTP/1.1 403 Forbidden
Content-Type: application/json
```

SP 以 HTTP 狀態碼來表示回覆請求失敗的狀況。

HTTP 狀態碼	說明
403	拒絕存取。

玖、MyData-API Endpoint 規格說明

一、系統環境與條件

API endpoint 以 RESTful Service 方式提供介面，且皆基於 TLS v1.1 以上提供加密傳輸管道。

二、MyData-API 請求及回覆規格說明

（一）SP 發出請求

```
GET /service/data HTTP/1.1
Host: mydata.nat.gov.tw
Content-Type: application/json
permission_ticket: {permission_ticket}
```

參數說明：

參數	說明
permission_ticket	代表該次用戶同意授權的交易識別碼。 格式為 version 4 UUID 字符串。 單次有效且唯一不重覆。 有效時間最長 24 小時。

（二）MyData 回覆請求成功 - 即時回應

```
HTTP/1.1 200 OK
Content-Type: application/jwt
```

請求回覆內容格式為 JWT，請參考章節「玖、三、MyData-API 的回傳格式與加簽機制說明」。

（三）MyData 回覆請求成功 - 等候處理

```
HTTP/1.1 429 Too Many Requests
Content-Type: application/jwt
Retry-After: {delay_seconds}
```

若 MyData-API 不能即時回應請求，則以 HTTP 429 回應。

參數說明：

參數	說明
delay_seconds	下次發動請求前需等待的秒數。

（四）MyData 回覆請求失敗

```
HTTP/1.1 403 Forbidden
Content-Type: application/jwt
```

MyData 以 HTTP 狀態碼來表示回覆請求失敗的狀況。

HTTP 狀態碼	說明
401	未完成身分驗證或身分驗證失敗。
403	拒絕存取。若請求來源 IP 不合法也會以此狀態回應。
504	無法傳送 DP 個人資料檔案。

三、MyData-API 的回傳格式與加簽機制說明

MyData-API 的回傳格式為 JWT (JSON Web Token)，服務提供者可以使用 MyData 平臺的 secret_key 來針對該 JWT 進行驗簽章的動作，以確認該 JWT 的內容沒有被竄改。

（一）JWT 格式說明

JWT 格式為「header.payload.signature」，三段資料以「.」隔開，範例如下：

```
eyJ0eXAiOiJKV1QiLA0KICJhbGciOiJIUzI1NiJ9
.
ewogICJjb2RlIjogIjAiLAogICJmaWxlbmFtZSI6ICJhYmMuemlwIiwK
ICAiZGF0YSI6ICJhcHBsaWNhdGlvbi96aXA7ZGF0YTpYc2RmYXNDU0ZEU
0FERkFTVmN4diIKfQ==
.
dBjftJeZ4CVP-mB92K27uhbUJU1p1r_wW1gFWFOEjXk
```

編碼前的 header 示意：

```
{
  "alg": "HS256"
  "typ": "JWT"
}
```

編碼前的 payload 示意：

```
{
  "filename": "abc.zip",
  "data": "application/zip;data:XsdfasCSFDSADFASVcxv"
}
```

驗證 signature 的方法說明：

1. 自 JWT 中擷取 header 的字符串，並以 Base64Decoder 解碼。
2. 自 alg 值 HS256，得知 MyData 所使用的演算法是 HMAC-SHA256。目前 MyData 固定使用 HS256，不會動態改變 alg 值。
3. 自 JWT 中擷取 header.payload 的字符串。
4. 以 secret_key，對 header.payload 進行 HMAC-SHA256 演算。
5. 演算後所得值如符合 JWT signature 值，即代表 JWT 未被竄改。

四、MyData-API 資料解密方法說明

完成上述 MyData-API 的回應內容 JWT 的簽章驗證無誤後。服務提供者接下來可將 JWT payload 的部份進行 Base64Decoder 解碼後，可得到一個 JSON 格式的資料內容。

（一）JSON 中的各欄位說明：

欄位	說明
filename	代表打包檔的檔案名稱，目前一律是壓縮 zip 檔，檔案名稱為 {client_id}.zip，client_id 為變數代表該服務項目的識別值。
data	代表 MyData 資料打包檔以 Base64Encode 編碼後的內容。其中 application/zip;data: 是前置碼，與資料內容無關，只是在說明 Base64Decoder 解碼及解密後的檔案格式為何。

（二）解密方法說明：

服務提供者將上述 data 欄位值進行 Base64Decoder 解碼處理後，須再以 MyData 提供的 secret_key 進行

AES（AES/ECB/PKCS5Padding）解密，解密後才將 binary 儲存為 filename 中所述的檔案名稱，即完成解密步驟。

五、MyData-API 的資料打包檔規格說明

由於服務提供者一次請求的用戶個人資料可能來自於多個不同的資料提供者，為了簡化 MyData 平臺與服務提供者之間傳遞資料的機制，MyData 平臺希望將來自於多個不同的資料提供者的資料檔案，打包成為一個壓縮 zip 檔案後再傳遞給服務提供者，因此 MyData 平臺規範了資料提供者檔案的打包規則，也同時規範了資料提供者檔案的打包規則，藉此達成讓服務提供者有一致性的檔案處理規則及做法。

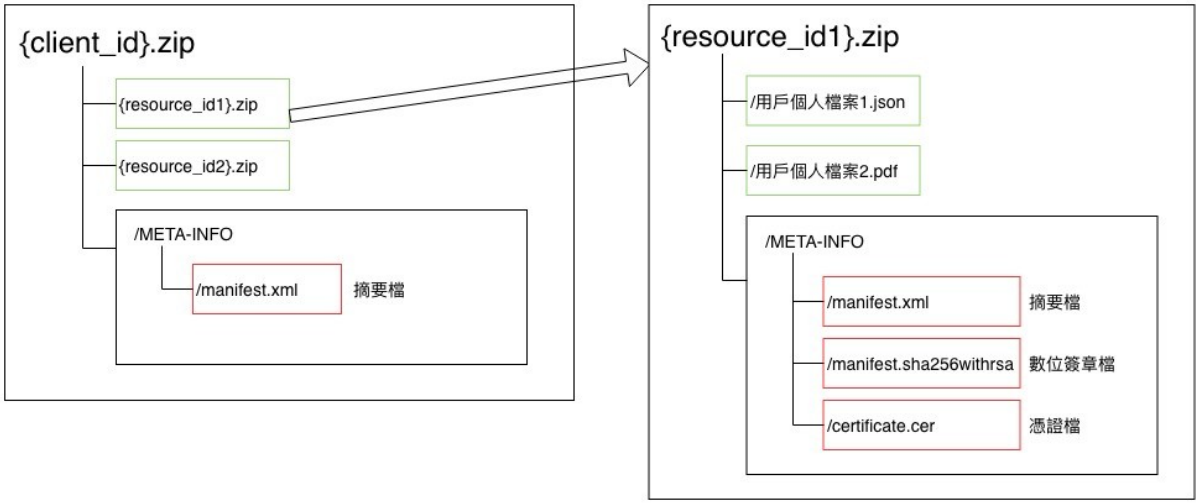
（一）MyData 資料打包檔規格要點如下：

1. MyData 資料打包檔格式為壓縮 zip 檔。
2. zip 檔的檔名為 {client_id}.zip。client_id 是變數，代表服務的識別代碼。
3. zip 檔中包含來自各資料提供者的資料打包檔，檔案格式為壓縮 zip 檔，檔名為 {resource_id}.zip，resource_id 是變數，代表資料集的識別代碼。
4. zip 檔中包含 META-INFO 的子目錄。
5. META-INFO 目錄中，包含 manifest.xml 摘要檔。

（二）manifest.xml 摘要檔格式說明如下：

```
<?xml version="1.0" encoding="UTF-8">
<files>
  <file>
    <filename>{resource_id1}.zip</filename>
    <resource_id>{resource_id}</resource_id>
    <resource_name>資料集中文名稱 1</resource_name>
  </file>
  <file>
    <filename>{resource_id2}.zip</filename>
    <resource_id>{resource_id}</resource_id>
    <resource_name>資料集中文名稱 2</resource_name>
  </file>
</files>
</xml>
```

（三）MyData 資料打包檔目錄結構示意如下



六、資料提供者的 DP 資料打包檔規格說明

為了使 MyData 的應用更加廣泛，同一份資料可能提供多種格式，包括機器可讀的格式，如：json, csv, xml, excel 等，以及易於人讀的格式，如：pdf, word, excel 等。

此外也包括了保證資料未經竄改的數位簽章檔，及為了方便服務提供者進行驗簽的憑證檔。

為了簡化 MyData 平臺與資料提供者(DP)之間傳遞資料的機制，MyData 平臺希望資料提供者可以將同一份資料的多個檔案打包成為一個壓縮 zip 檔案後再傳遞給 MyData 平臺，因此 MyData 平臺規範了用戶個人資料檔案的打包規則，同時也使服務提供者在處理來自不同資料提供者的檔案時，可以有一致性的處理規則及做法。

（一）DP 資料打包檔案規格要點如下：

1. DP 資料打包檔格式為壓縮 zip 檔。

2. zip 檔的檔名非限定但建議為 {resource_id}.zip。resource_id 是變數，代表資料集的識別代碼。

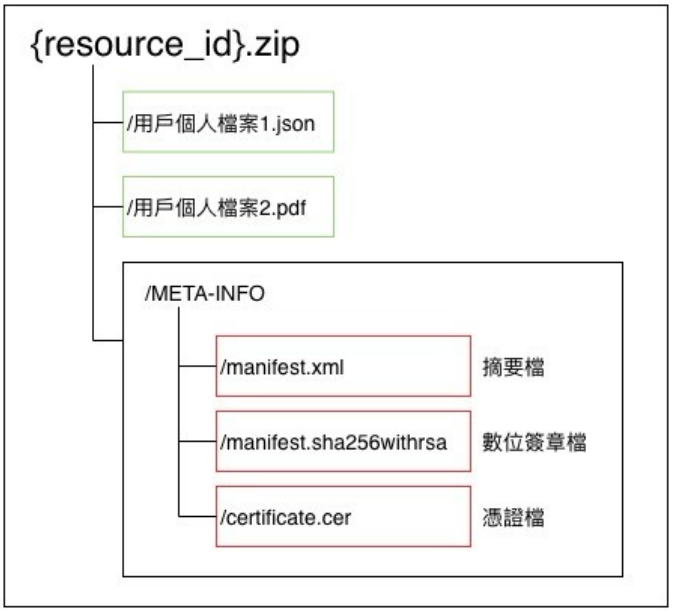
3. zip 檔中可能包含多個個人資料檔案，依資料提供者規範需包含的檔案用途、數量、檔名、副檔名等。

4. 如提供數位簽章，則包含於 META-INFO 子目錄。

5. 承上，META-INFO 子目錄中包含檔案：

   - manifest.xml 摘要檔
   - manifest.sha256withrsa 數位簽章檔
   - certificate.cer 憑證檔

（二）DP 資料打包檔案目錄結構示意如下：



（三）META-INFO 目錄及內含檔案說明：

META-INFO 目錄下放置摘要檔、數位簽章檔及憑證檔。若 DP 沒有產製數位簽章，則不會產生 META-INFO 子目錄。

manifest.xml：

針對各別的資料檔案，以 SHA256 演算法，演算出的數位指紋（摘要值）後載明於 manifest.xml 檔案中。

內容格式示意如下：

```
<?xml version="1.0" encoding="UTF-8">
<files>
  <file>
    <filename>用戶個人資料檔案1.json</filename>
    <digest>{digest value}</digest>
  </file>
  <file>
    <filename>用戶個人資料檔案2.pdf</filename>
    <digest>{digest value}</digest>
  </file>
</files>
</xml>
```

manifest.sha256withrsa：

以 SHA256 演算出 manifest.xml 的數位指紋後，以 DP 的 RSA 私鑰進行加密演算後所得的二進位內容，以副檔名 sha256withrsa 來示意所使用的演算機制為 SHA256withRSA。

certificate.cer：

憑證檔。PEM 格式的憑證資訊。PEM 格式的檔案是 ASCII (base64) 檔案，內容包含前置及後置文字，如下示意：

```
-----BEGIN CERTIFICATE-----
MIID/
zCCAuegAwIBAgIJAMhtYm3fde9AMA0GCSqGSIb3DQEBCwUAMIGVMQswCQ
YD
-----END CERTIFICATE-----
```

七、驗證 DP 資料檔案沒有被竄改的方法說明

（一）驗證憑證檔的有效性

服務提供者可向簽發憑證的 CA 驗證憑證有效性。原則上會建議 DP 向 GCA 政府憑證管理中心來申請數位簽章用的憑證。

GCA 支援兩種驗證憑證有效性的方法，包括：CRL 及 OCSP。

（二）憑證檔中取出 DP 公鑰

DP 夾帶的憑證檔為 PEM 格式，SP 須從憑證檔中取出 DP 公鑰，用於後續驗證數位簽章檔案 manifest.sha256withrsa。

（三）驗證 manifest.xml 沒有被竄改

manifest.xml 檔案中載明了各別資料檔案的數位指紋（摘要值）。因此先驗證 manifest.xml 沒有被竄改，即代表 manifest.xml 檔中所載明的摘要值沒有被竄改。

manifest.sha256withrsa 是針對 manifest.xml 所做出來的數位簽章檔，SP 須以 DP 的公鑰對 manifest.sha256withrsa 進行解密後，可得到正確的 manifest.xml 的摘要值。

SP 對 manifest.xml 進行 SHA256 演算後，比較前後兩者摘要值是否相符，若相符則代表 manifest.xml 沒有被竄改。

（四）驗證各別的資料檔案沒有被竄改

SP 讀取 manifest.xml 內容後，得到各別資料檔案的正確的摘要值，再針對各別資料檔進行 SHA256 演算後，比較前後兩者摘要值是否相符，若相符則代表該資料檔案沒有被竄改。