VeriDas

/Phygital

/Integration with external authorization APIs

/API Authentication

The third-party should implement an authentication endpoint based on OAuth 2.0 Client Credentials Grant protocol.

- The third-party provides Veridas its **client_id** and **client_secret** to be able to complete the authentication process.
- An authentication url needs to be defined.
- The result of the authentication is to obtain an access token, which will be used as the authentication header in endpoint requests

Authentication Request

Entity	Method	Content-Type	Header	Request Content
Veridas	POST	application/x-www-form-urlencoded	{ "Authentication": "Basic base64({client_id}:{client_secret})" }	{ "grant_type": "client_credentials", "scope": "openid" }

Authentication Response

Entity	Content-Type	Response Content
Third-party	application/json	{ "token_type": token_type (usually Bearer), "access_token": auth_token, }

/Access

In the case of the physical access process, the call from Veridas to the third-party with the information of the user contains the information that the system is handling at the moment of the authorization of the physical access. Besides, the user_id of the user is always sent; therefore, it can be always correlated by the third party with the information of the registration (*optional*).

Important: the resource endpoint will be the same both for user registration (*optional*) and access. The third-party clients must use the field **action** that comes in the request if they want to make different actions but the endpoint must be the same. Also the response must be the same and mandatory.

Resource Request

Entity	Method	Endpoint	Content-Type	Request Content

Veridas	POST	/api/enforcements	application/json	{
				"timestamp": isoformat_timestamp,
				"subject": user_id,
				"object": access_point,
				"action": "**enter**",
				"claims": {
				"acs_id": user_id
				}
				}

The **acs_id** is used by terminals to identify the person who is being authenticated.

Resource Response

Entity	Content-Type	Response Content

Third-party	application/json	{
		"result": "granted"/"denied",
		"reason": just_if_denied,
		}

/User registration [*OPTIONAL*]

During the registration phase, Veridas can make a call to the third-party API with the user's data in order to allow the third-party to register the person doing the onboarding in their system as well. The third-party must return either **granted** or **denied**. If the result is granted, we'll continue with the onboarding and the user will be registered in Veridas. If the result is denied, the onboarding will fail and the user won't be registered. This can be optional as some integrations just require the access authentication use case.

The third-party **must store** all information needed about the user as in access **just the acs_id** will go in the request.

Resource Request

Entity	Method	Endpoint	Content-Type	Request Content

Veridas	POST	/api/enforcements	application /json	{
				"timestamp": isoformat_timestamp,
				"subject": user_id,
				"object": saloon_location,
				"action": "**onboard**",
				"claims": {
				"name": name,
				"surname": surname,
				"birth_date": birth_date,
				"document_type": id_card, passport...,
				"id_number": id_number,
				“expiry_date”: document_expiry_date
				“document_number”: doc_number.
				"extra_fields": {
				"email": email,
				“phone”: phone,
				}
				}
				}

* The claims value of the request content is just an example. Data can vary depending on onboarding required info.

Resource Response

Entity	Content-Type	Response Content

Third-party	application/json	{
		"result": "granted"/"denied",
		"reason": just_if_denied,
		}

/User cancellation [*OPTIONAL*]

In order to be synchronized in both Veridas and Third-party systems, we recommend making use of this use case if a third-party integrator uses the user registration. This can be a bidirectional request. However, we will always make a deletion request to the third-party API, regardless of whether the deletion is triggered by Veridas (Web) or by the third-party.

Request triggered by Veridas

The endpoint will be defined by the third-party. **Must** include the **Veridas user_id** in the resource endpoint.

Entity	Method	Content-Type	Endpoint Example
Veridas	DELETE	application/json	DELETE /third-party-api/users/**user_id**

Request triggered by Third-Party

Must call the endpoint with the **user_id**

Entity	Method	Content-Type	Endpoint
Third-party	DELETE	application/json	DELETE /veridas-api/users/**user_id**