# VeriSaas das-Peak API V1

## *Release 2019Q4.1*

| Release | Date | Description | Author | Reviewer | Approver |
|---|---|---|---|---|---|
| 2.2.1 | 17/01/2020 | VeriSaas das-Peak 2019 Q4.1 | SPC | MSY | MSY |

# 0. What's new?

This new version of das-Peak incorporates two new endpoints to the API for speaker identification functionalities. Briefly, this new version of das-Peak introduces the following changes:

- Identification endpoint to compare a wav file with a voice credentials list, returning the highest score credential and the rest of scores.
- Identification endpoint to compare a voice credential with a voice credentials list, returning the highest score credential and the rest of scores.

# 1. das-Peak microservice

Voice biometrics is a state-of-the-art technology that allows a person to be validated by his/her voice. VERIDAS solution captures the unique physical features of the vocal apparatus and features such as frequency, speed and accents and compiles them together into a virtually **unique voice biometric vector** per person.

**The voice biometric vector is a mathematical descriptor obtained from the characteristics of the voice in an audio recording. This mathematical conversion from voice into a biometric vector is irreversible. Therefore, it is not possible to recover a person's voice signal from the calculated biometric vector.**

VERIDAS has developed an own speaker verification engine (das-Peak) as a cloud-based solution that can be consumed via APIs.

VERIDAS' voice biometrics technology is based on the use of neural networks and has been evaluated at position number 13 out of 204 participants (**being the 4th best company**), worldwide, in the **NIST (National Institute of Standards and Testing, USA) SRE 2018. Veridas achieved 0.56% of SRE on NIST dataset.**

das-Peak calculates the similarity between two audio recordings (in terms of the speakers present in them) using biometric algorithms. das-Peak engine allows to identify users voice **without** the need of using a password or predefined phrase (passive recognition) as it is based on **text-independent technology**. This means that the biometric comparison is related to the voice

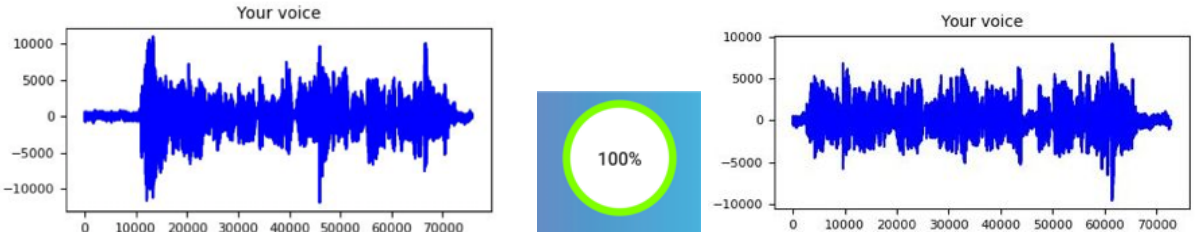
characteristics and not to the content of the sentence. However, the system is flexible to use pre-defined phrases in order to fulfill customer requirements or additional controls.

Within the voice biometrics field, two scenarios are typically handled:

- **Verification**: The process of checking the identity of a person by comparing two audios.
- **Identification**: The process of searching a person or a set of persons within a database of identities and its audio input data.

So far, das-Peak holds solution for the verification and identification problem.

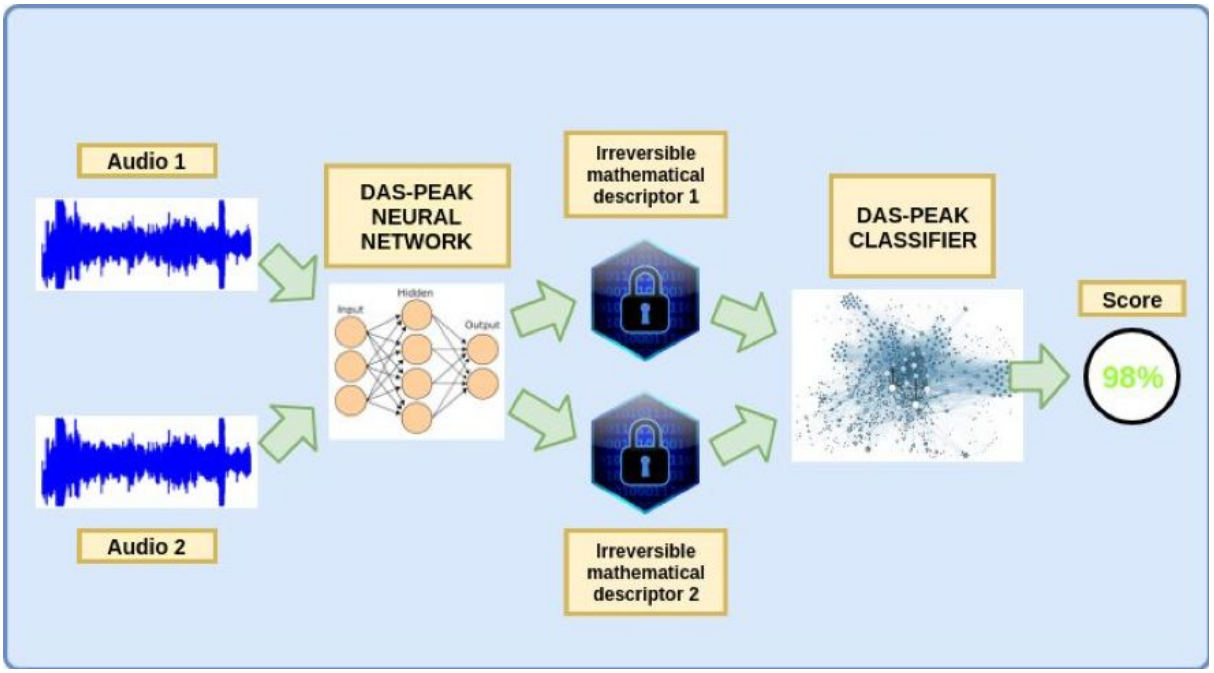
Given two audio recordings, the system returns a score based on the similarity of both of them, not regarding to speech recognition but to the speakers present in them.




das-Peak is offered as an API REST format. The process to obtain the value of similarity between two audios is described below.

1. Two audio recordings are sent to the API.
2. The audio recordings are pre-processed. This process detects voice in the audio recordings (removing parts of silence) and analyzes the noise of the signals.
3. The audio recordings are converted into irreversible mathematical descriptors (voice biometric vectors).
4. Both mathematical vectors are compared and a matching score between 0 and 1 is provided. This matching score represents the probability that the audios belong to the same person. The higher the score, the greater the certainty to be the same person.
5. You can use this matching score to validate the identity of a customer. Recommendation to define a threshold within required confidence level using the FPR (False Positive Rate) and FNR (False Negative Rate) expected ratios (see section 1.1.1.2)

das-Peak also provides the biometric vector generated from audio. With this information, it is possible to carry out the verification between a biometric registration vector and a new audio, instead of between two audios.




VERIDAS does not store any personal data in the cloud. All the user information (i.e., both the audio recordings as well as the processed data) is **immediately deleted**.

## 3. Main features

The main features of the das-Peak voice biometrics engine are:

- **Text-independent**: Allows to compare phrases with different content. That is, the user does not have to remember any phrase or have to read the same phrase to be authenticated.
- **Certified technology**: VERIDAS achieved 0.56% of SRE on NIST 2018 dataset, being the 4th best company worldwide.
- **Language-independent**: Allows to compare voices in different languages. The biometric voice engine has been specially trained for the following languages: German, English, Spanish, French, Italian, Chinese, Taiwan, Dutch, Estonian, Persian, Turkish, Welsh, Kabyle, Catalan and Euskera.
- **Minimum duration:** Allows verifying audios with a minimum voice duration of 5 seconds.
- **Verification time:** 0.14 seconds for comparing a biometric vector and an audio.
- **Minimum size biometric vector:** The biometric vector size is 1.1 Kbytes.
- **Voice activity detection**: Compute the total quantity of voice in the input audio to accept a verification request.
- **Noise detection**: Compute the total quantity of noise against voice in the input audio to accept a verification request.

To ensure optimal performance, recommendation for audios to be captured under specific conditions is needed. VERIDAS offers different proprietary SDKs for audio recording, and they are available for different platforms (iOS, Android). These SDKs ensure that capture process is performed following the best conditions. Most relevant conditions are:

- **Audio format:** WAV.
- **Number of channels**: Mono.
- **Bits per Sample**: 16.
- **Sampling frequency**: At least 8 kHz.

das-Peak can process any audio that meets the above conditions, not just audios recorded with the SDKs.

# 4. System quality report

## 4.1. Verification performance

VERIDAS has evaluated its voice biometrics model (das-Peak 2019Q4.1 version) with an internal database with different duration enrollment audios and different duration audios test (26.544 comparisons) in different acoustic conditions obtaining the values of False Positive Rates (FPR) and False Negative Rates (FNR) with different threshold values. The FPR is the probability to accept a non legit person and the FNR is the probability to reject a legit person. With this values it is possible to choose the desired working point of the voice biometric system.

| | Enrollment=5s Verification=5s | | Enrollment=10s Verification=5s | | Enrollment=10s Verification=10s | |
|---|---|---|---|---|---|---|
| **Similarity Threshold** | **FPR(%)** | **FNR(%)** | **FPR(%)** | **FNR(%)** | **FPR(%)** | **FNR(%)** |
| 0.5 | 1 | 2.01 | 1 | 0.7 | 1 | 0.1 |
| 0.55 | 0.7 | 2.4 | 0.7 | 0.9 | 0.7 | 0.15 |
| 0.6 | 0.5 | 3.02 | 0.5 | 1.3 | 0.5 | 0.26 |
| 0.65 | 0.3 | 3.8 | 0.3 | 1.6 | 0.39 | 0.29 |
| 0.7 | 0.2 | 4.8 | 0.27 | 1.9 | 0.27 | 0.59 |
| 0.75 | 0.17 | 6.3 | 0.17 | 2.7 | 0.18 | 0.69 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 0.8 | 0.1 | 8.7 | 0.1 | 3.6 | 0.1 | 1.1 |
| 0.85 | 0.05 | 12.2 | 0.05 | 5.4 | 0.06 | 1.7 |
| 0.9 | 0.02 | 18.5 | 0.02 | 8.8 | 0.02 | 2.7 |
| 0.95 | 0.01 | 32.9 | 0.01 | 17.1 | 0.01 | 5.7 |

This calibration shows different security work points depending on the similarity threshold and the audios voice duration that are compared.

For example, If the use case is 5 seconds to register and 5 seconds to verify, with a threshold equal to 0.6, it is obtained FPR=0.5% and FNR=3.02%. In this case, 96.98% of the comparisons of a person voice and its corresponding voice registration will be considered as authentic, and only 0.5% of the cases comparing voices to different persons will be incorrectly classified as authentic.

Possibility to further calibrate system with real use cases (if call recordings provided) during integration process.

### 4.2. Identification performance

VERIDAS has evaluated its voice biometrics model (das-Peak 2019Q4.1 version) with a internal database with 250 speakers and 5896 audios for the identification 1:N use case (N=2 to 10). In this database, speakers have been recorded in different sessions with different acoustic conditions (street, pub, train station, room, church,...). A number of 1000 random identification tasks have been performed for each N (from N=2 to 10). For example, in a 1 to 10 identification process (N=10) the probability to identify the right individual is **97.8 %**. The identification accuracy results for N=2 to 10 can be observed in the following table:

| **Nº Speakers** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|
| **Accuracy (%)** | 99.6 | 99.1 | 98.9 | 98.7 | 98.4 | 98.2 | 98.0 | 97.9 | 97.8 |

# 5. API Considerations

The following are some general considerations about this API that must be taken into account before consuming the service.

## 5.1. Authentication

This service sits behind a gateway responsible for authenticating end users and routing requests. The authentication method is API key based.

## 5.2. Requests

- The multipart/form-data content type must be used on every request.
- The wav files sent to das-Peak must have a format extension (.wav).
- The API is HTTP-based and uses SSL everywhere with valid certificates. For security reasons, customers should never trust das-Peak endpoints exposing invalid certificates.
- Endpoints attempt to conform to the design principles of Representational State Transfer (REST).
- The service includes an /alive endpoint that returns the 200 HTTP status code if the service is up and running. This can be used to check the service's health.

All responses will be encoded using JSON, regardless of the accepted content-type specified by the client. Responses will return a suitable HTTP status code indicating if the request was successful (200 or 204 if nothing else is returned) or not (any other code). Responses will also include a code field in the JSON body that can provide more information about the concrete error on each case.

In general, successful responses will have the following format:

HTTP Status: 200 OK

```
{
      "data": {
            DATA
      }
}
```

or

In case of error:

| Field | Description |
|---|---|
| exception | exception that raised the error |
| error | A message indicating what went wrong |

Example:

```
{
	“exception”: “InputException”,
	“error”: “The wav is not mono.”
}
```

### 5.3 Versioning

The API version will be included in the URL, after the base url and before the endpoint:

`https://<base_url>/<service>/v{number:integer}/<endpoint>`

Non-backwards compatible changes will cause a version increment. As of now, the API only supports the **v1** version.

## 6. API Definition

This service is a REST API where the following endpoints are exposed:

**Public Base URL** (v1):

https://<base_url>/daspeak/v1/

**Resources**:

| Method | Public URL | Description |
|---|---|---|
| **GET** | /alive | Checks if the service is up |
| **POST** | /credential/wav | Generate a biometric credential from a given audio input, using the latest available model |
| **POST** | /similarity/credential2wav | Computes the voice similarity between an audio input and a previously generated voice biometric credential, and returns how similar are both voices |
| **POST** | /similarity/wav2wav | Computes the voice similarity between two audio inputs, and returns how similar are both voices |

| | | |
|---|---|---|
| **POST** | /identification/wav2credentials | Computes the voice similarity between an input wav and a list of voice credentials, and returns the guess result and a list of all results. A result contain credential ID and its score. |
| **POST** | /identification/credential2credentials | Computes the voice similarity between a reference voice credential and a list of voice credentials, and returns the guess result and a list of all results. A result contain credential ID and its score. |

### 6.1. Check if the service is alive

The service receives a GET request with no params, and returns a 200 status code indicating that the server is up.

**GET** /alive

#### Response: 200

Empty response.

#### Response: 500

Server error response.

Content-Type: application/json

| exception | error message |
|---|---|
| ServerError | Unexpected server fatal error |

### 6.2. Generate a voice biometric credential

This endpoint is used to generate a voice biometric credential from a given audio input, using the latest available model. The biometric credential size is 1.1 Kbytes.

**POST** `/credential/wav`

### Request Body

Request for voice biometric credential generation.

| Name | Req. | Type | Description |
|---|---|---|---|
| audio | yes | WAV file | Audio with the target speaker voice. As multipart/form-data, it should be a file, as application/json, the file content encoded in base64. |

### Response: 200

Returns one voice biometric credential for the given audio file.

#### Content-Type: application/json

| Name | Req. | Type | Description |
|---|---|---|---|
| credential | yes | string | A biometric credential string |
| version | yes | string | An object with two items: hash of the model used to generate the biometric credential, and the operation mode used to configure the model. |

### Response: 400

Request format error.

#### Content-Type: application/json

| exception | error message |
|---|---|
| InputException | The wav is not mono |
| InputException | The wav sample rate is not 8000 |
| InputException | The wav bits per sample are not PCM_16 |
| InputException | The wav duration is longer than 30s |
| InvalidAudio | Noise level exceeded |
| InvalidAudio | The duration of the voice is not enough: 2.5 s less than 3.0 s |

### Response: 500

Server error response.

Content-Type: application/json

| exception | error message |
|---|---|
| Exception | Error opening <_io.BytesIO object at 0x7f6baab4dbf8>: File contains data in an unknown format. |

## 6.3. Compute similarity between biometric credential and audio

Computes the voice similarity between an audio input and a previously generated voice biometric credential, and returns how similar are both voice.

**POST** `/similarity/credential2wav`

Request Body

Request for voice verification with a voice biometric credentials and wav audio input.

| Name | Req. | Type | Description |
|---|---|---|---|
| credential_reference | yes | string | A reference voice biometric credential string generated with 6.2 API endpoint |
| audio_to_evaluate | yes | WAV file | Audio to evaluate with the speaker voice. As multipart/form-data, it should be a file, as application/json, the file content encoded in base64. |

Response: 200

Returns the confidence (or similarity) between the audio to evaluate and the biometric voice credential, being more similar as much close this number is to one. The number is in range [0,1].

Content-Type: application/json

| Name | Req. | Type | Description |
|---|---|---|---|
| confidence | yes | number | A probability number in range [0,1] |
| version | yes | string | API version |

Response: 400

Request format error.

Content-Type: application/json

| exception | error message |
|---|---|
| InputException | The wav is not mono |
| InputException | The wav sample rate is not 8000 |
| InputException | The wav bits per sample are not PCM_16 |
| InputException | The wav duration is longer than 30s |
| InvalidAudio | Noise level exceeded |
| InvalidAudio | The duration of the voice is not enough: 2.5 s less than 3.0 s |

Response: 500

Server error response.

Content-Type: application/json

| exception | error message |
|---|---|
| Exception | Decryption error |
| Exception | Incorrect padding |
| Exception | Error opening <_io.BytesIO object at 0x7f6baab4dbf8>: File contains data in an unknown format. |

## 6.4. Compute similarity between two audios

Computes the voice similarity between two audio inputs, and returns how similar are both voices.

**POST** `/similarity/wav2wav`

### Request Body

Request for voice verification with two audios.

| Name | Req. | Type | Description |
|---|---|---|---|
| audio_reference | yes | WAV file | Audio with the reference speaker voice. As multipart/form-data, it should be a file, as application/json, the file content encoded in base64. |
| audio_to_evaluate | yes | WAV file | Audio to evaluate with speaker voice. As multipart/form-data, it should be a file, as application/json, the file content encoded in base64. |

### Response: 200

Returns the confidence (or similarity) between both speaker voices, being more similar as much close this number is to one. The number is in range [0,1].

### Content-Type: application/json

| Name | Req. | Type | Description |
|---|---|---|---|
| confidence | yes | number | A probability number in range [0,1] |
| version | yes | string | API version |

### Response: 400

Request format error.

### Content-Type: application/json

| exception | error message |
|---|---|
| InputException | The wav is not mono |
| InputException | The wav sample rate is not 8000 |
| InputException | The wav bits per sample are not PCM_16 |
| InputException | The wav duration is longer than 30s |

| | |
|---|---|
| InvalidAudio | Noise level exceeded |
| InvalidAudio | The duration of the voice is not enough: 2.5 s less than 3.0 s |

Response: 500

Server error response.

Content-Type: application/json

| exception | message |
|---|---|
| Exception | Error opening <_io.BytesIO object at 0x7f6baab4dbf8>: File contains data in an unknown format. |

### 6.5. Compute voice identification between a wav input and a list of credentials

Computes the voice similarity between an input wav and a list of voice credentials, and returns the identification results and the scores.

**POST** /identification/wav2credentials

Request Body

Request for voice identification with one audio input and a voice credentials list.

| Name | Req. | Type | Description |
|---|---|---|---|
| audio_reference | yes | WAV file | Audio with the reference speaker voice. As multipart/form-data, it should be a file, as application/json, the file content encoded in base64. |
| credentials_list | yes | Dictionary | List of voice credentials with its corresponding identification tags. Each element on this list shall contain user ID and its credential using a dict with fields “id” and “credential” |

Response: 200

Returns the identification tag  with the score and a dictionary with all the tags and scores.

Content-Type: application/json

| Name | Req. | Type | Description |
|---|---|---|---|
| result | yes | Dictionary | Tuple with the identification tag and its score. |
| scores | yes | Dictionary | Results with the identification tags and scores. |

Response: 400

Request format error.

Content-Type: application/json

| exception | error message |
|---|---|
| InputException | The wav is not mono |
| InputException | The wav sample rate is not 8000 |
| InputException | The wav bits per sample are not PCM_16 |
| InputException | The wav duration is longer than 30s |
| InvalidAudio | Exceeded Noise Level |
| InvalidAudio | The duration of the voice is not enough: 2.5 s less than 3.0 s |
| InputException | Length of credentials list exceeds the maximum allowed. |

Response: 500

Server error response.

Content-Type: application/json

| exception | message |
|---|---|
| Exception | Error opening <_io.BytesIO object at 0x7f6baab4dbf8>: File contains data in an unknown format. |

## 6.6. Compute voice identification between a reference credential and a list of credentials

Computes the voice similarity between a reference voice credential and a list of voice credentials, and returns the identification results and the scores.

**POST** /identification/credential2credentials

### Request Body

Request for voice identification with one reference voice credential and a voice credentials list.

| Name | Req. | Type | Description |
|---|---|---|---|
| credential_reference | yes | Dictionary | Tuple with the voice credential and its identification tag. |
| credentials_list | yes | Dictionary | List of voice credentials with its corresponding identification tags. Each element on this list shall contain user ID and its credential using a dict with fields “id” and “credential” |

### Response: 200

Returns the identification tag with the score and a dictionary with all the tags and scores.

Content-Type: application/json

| Name | Req. | Type | Description |
|---|---|---|---|
| result | yes | Dictionary | Tuple with the identification tag and its score. |
| scores | yes | Dictionary | List with the identification tags and scores. |

### Response: 400

Request format error.

Content-Type: application/json

| exception | error message |
|---|---|
| InputException | Length of credentials list exceeds the maximum allowed. |

Response: 500

Server error response.

Content-Type: application/json

| exception | message |
|---|---|
| Exception | Error opening <_io.BytesIO object at 0x7f6baab4dbf8>: File contains data in an unknown format. |

# 7. License

The following clauses set the terms, rights, restrictions and obligations on using this Software, created and owned by VERIDAS DIGITAL AUTHENTICATION SOLUTIONS, S.L. ("VERIDAS" or the "Licensor"), without prejudice to the provisions laid down in the contracts subscribed by the Licensor and your entity (the Licensee), which shall prevail over this file.

## 7.1 LICENSE GRANT

VERIDAS hereby grants to the Licensee a non-exclusive, non-assignable and non-transferable, non-commercial, indivisible, without the rights to create derivative works license for the term specified in the Offer to use the offered software (the "Software") for the specific purpose specified between the Parties and/or in the Offer, subject to the terms and conditions contained herein and other legal restrictions set forth in third party software used while running the Software.

The Software has different components that can be used for several applications. However, the license is granted over the Software licensed components as a whole, and no separated use is permitted other than the specific purposes agreed by the Licensor and the Licensee.

The Software is comprised of proprietary code. However, the Software may include certain third-party components with separate legal notices or governed by other agreements, as may be described in the Software. Even if such components are governed by other agreements, the disclaimers and the limitations on and exclusions of damages below also apply. On specific products, if necessary, VERIDAS may install a computer application, in some cases connected with an external server, that allows VERIDAS to verify that the system is updated and payments are correctly made.

## 7.2 USE OF THE SOFTWARE

2.1. The Licensee cannot use the Software for other purposes than as specified in the Offer.

2.2. The Licensee may permit its employees to use the Software for the purposes agreed by the parties and/or described in the Offer, provided that the Licensee takes all necessary steps and imposes the necessary conditions to ensure that all employees using the Software do not commercialize or disclose the contents of it to any third party, or use it other than in accordance with the terms herein.

2.3. The Licensee will not distribute, sell, license or sub-license, lease, trade or expose for sale the Software to a third party.

2.4. No copies of the Software are to be made other than as expressly approved by VERIDAS.

2.5. No changes to the Software or its content may be made by Licensee.

2.6. The Licensee will provide technological and security measures to ensure that the Software, which the Licensee is responsible for, is physically and electronically secure from unauthorized use or access.

2.7. The Licensee shall ensure that the Software retains all VERIDAS copyright notices and other proprietary legends and all trademarks or services marks of VERIDAS, as specified in clause 3 below.

2.8. The Licensee shall not, under any circumstances, use reverse engineering practices on the Software.

2.9. The Licensee is responsible for extending the obligations herein to Clients, to the extent they may apply.

## 7.3 INTELLECTUAL PROPERTY RIGHTS

3.1. Intellectual Property Rights means all rights in and to any copyright, trademark, trading name, design, patent, know-how, trade secrets and all other rights resulting from intellectual activity in the industrial, scientific, literary or artistic field and any application or right to apply for registration of any of these rights and any right to protect or enforce any of these rights.

3.2. All Intellectual Property Rights over and in respect of the Software are owned by VERIDAS. The Licensee does not acquire any rights of ownership in the Software, and it must use the Intellectual Property Rights exclusively as required for reasonable and customary use within the purposes of the License.

3.3. Any modification made on the Software in order to adapt it for the provision of the service to the Licensee and/or the Client, shall be include in the Intellectual Property Rights as defined in clause 3.2 above.

## 7.4 LIMITATION OF LIABILITY

4.1. To the extent permitted under the law, the Software is provided under an "AS IS" basis. The Licensee acknowledges and agrees that neither VERIDAS nor its board members, employees or agents, will be liable for any lost or damage arising out of or resulting from VERIDAS' provision of the Software under this License, or any use of the Software by the Licensee, the Client or their employees. The Licensee hereby releases VERIDAS to the fullest extent from any such liability, loss, damage or claim, both its own or of the Clients. Regarding the processing of personal data with the Software, the Licensee acknowledges and agrees that the Licensor is no liable for the data collected by the use of the Software within the scope of the Licensee's activities.
This limitation shall apply to any issue related to the software, services, contents (including code) found at third-party websites or third-party programs.

4.2. The Licensee must indemnify, defend and hold harmless the Licensor, its board members, employees and agents from and against any and all claims (including third party claims), demands, actions, suits, expenses (including attorney's fees) and damages (including indirect or consequential loss) resulting in any way from: Licensee's and Licensee's employee's use or reliance on the Software; any breach of the terms of the License by the Licensee or its employees; any other act of Licensee that can be considered negligent.

4.3. Notwithstanding the previous general clause, VERIDAS expressly excludes any liability resulting from:

(a) willful, fraudulent, deliberately unlawful acts, penalized as a criminal offence, or which are voluntarily against the law, carried out by or against the Licensee or a Client, or their employees;
(b) any fact or circumstance, real or suspected, the Licensee or the Clients know about or could have reasonably foreseen, and that may affect, in any way, to the correct functionality of the Software;
(c) mechanical fails, electric fails (including interruptions, outages, overvoltages or power cuts) and fails on telecommunication or satellite transmission systems, given that those fails are not due to an act or omission of VERIDAS or to an error of the delivered Software;
(d) damage or loss of the Licensee's or Client's data stored or hold in VERIDAS' systems, as well as the costs resulting from such circumstance; or

(e) any act or claim alleging, derived from or based on funds, money or value transfers or any other negotiable instrument for or from a bank or financial institution.

4.4. Licensee and Clients acknowledge that the Software has an associated error rate which makes not possible, considering the state of the art, to guarantee a complete level of reliability. In this sense, VERIDAS provides information about the levels and error rates of every Software version. On the other hand, the outputs of the system are not binary but they offer a probabilistic result that has to be configured by the Licensee or the Client.

The level of reliability is subject to voice capture conditions during the process, and therefore the Licensor shall not be responsible of any use the Licensee may make of the Software in different environments than those recommended by the Licensor.

As a result, it is the responsibility of the Licensee and/or the Client to carry out any necessary internal control evaluation, to implement due diligence measures in accordance with the regulation they have to comply with (including without limitation: regulation on the prevention of money laundering, citizen security, border controls, etc.) and to evaluate the convenience of the Software as an instrument for complying with current legislation.

VERIDAS provides some tools that may help the Licensee and/or the Client to implement prevention mechanisms, but it is not responsible for the study of the convenience of its implementation, the specific configuration of the Software, or the use of the Software (expressly including the validation results obtained with the Software).

4.5. Licensee is responsible for communicating and transferring the previous limitations of liability to the Clients.