das-Peak v2.3.1
VeriSaaS Release 2020Q3

Release	Date	Description	Author	Reviewer	Approver
2.3.1	08/09/2020	das-Peak 2020Q3	SPC	MSY	MSY

0. What’s new? 3

1. das-Peak microservice 3

2. Main features 5

3. System quality report 7

3.1. Verification performance (Telephone Channel) 7

3.2. Verification performance (Lossless Audio) 8

3.3. Identification performance 9

3.4. Voice Authenticity (Anti-spoofing) 9

4. Voice Verification Use Case 10

4.1 Enrollment 10

4.2 Verification 11

5. API Considerations 12

5.1. Authentication 12

5.2. Requests 12

5.3 Versioning 13

6. API Definition 14

6.1. Check if the service is alive 15

6.2. Get list of available models 15

6.3. Obtain metadata from a model 16

6.4. Obtain calibration modes of a model 17

6.5.Obtain metadata from a voice biometric credential 19

6.6. Generate a voice biometric credential 20

6.7. Compute similarity between voice biometric credential and audio input 22

6.8. Compute similarity between two audio inputs 25

6.9. Compute voice identification between a wav input and a list of voice credentials 27

6.10. Generate a voice biometric credential with a specific model 30

7. License 32

7.1 LICENSE GRANT 33

7.2 USE OF THE SOFTWARE 33

7.3 INTELLECTUAL PROPERTY RIGHTS 34

7.4 LIMITATION OF LIABILITY 34

0. What’s new?

This new version of das-Peak incorporates these new functionalities:

- Different multi-calibration modes: this new functionality allows customers to select between different calibration modes to get the best accuracy of the verification/identification results depending on the use case (telephone audio, lossless audio or non-calibration).
- New endpoint to obtain metadata information of a model given its hash. With this endpoint, it is possible to get some metadata about a specific model.
- New endpoint to obtain the types of calibration that a model supports. With this endpoint, it is possible to know the calibration allowed in a specific model to be used for a specific use case.
- New endpoint to obtain the hash model that was used to create a voice credential.
- Support for a contextual data header: this new header is logged in every query allowing to build statistics and billing segregated by customer, use cases, sub-clients and so on.

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.**

VERIDAS’ voice biometrics group has participated in the **short-duration Speaker Verification Challenge (SdSV) 2020 getting the 3º award** (2º single model), demonstrating best results in the state of the art in Voice Biometrics for **short utterances conditions**. [https://sdsvc.github.io/]

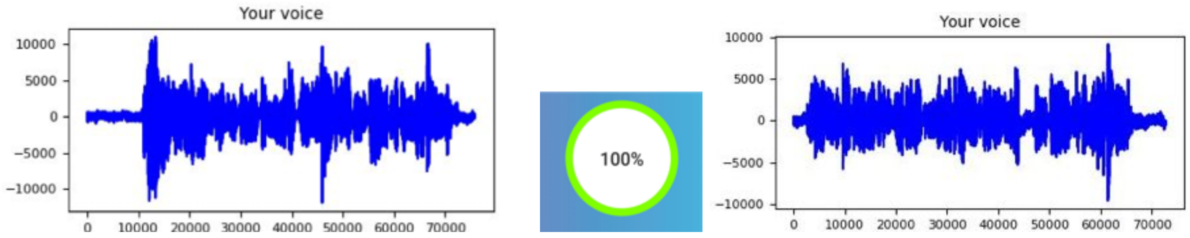
das-Peak calculates the similarity between two audio recordings (in terms of the speakers present in them) using biometric algorithms. das-Peak engine allows to authenticate users voice **without** the need of using a password or predefined phrase (passive recognition) as it is based on **text-independent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

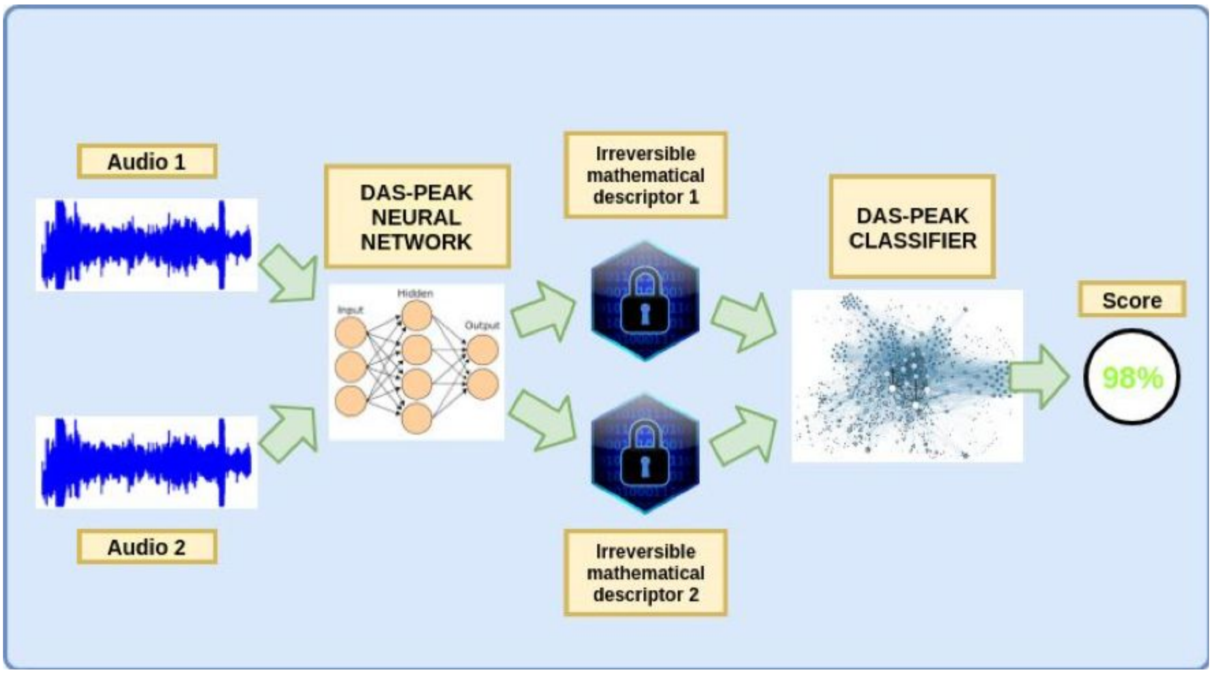


das-Peak is offered as an API REST format. The process to obtain the value of similarity between two audios is described below.

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.

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**.

2. 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. In the Short-duration Speaker Verification challenge 2020 (SdSV), Veridas achieved a 0.0769 %, being the 2th best company in single system.
- **Language-independent**: Allows to compare voices in different languages. The biometric voice engine has been specially trained for the following languages: English, Spanish, German, 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 1.1Kbytes.
- **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 in the input audio to accept a verification request.
- **Voice Authenticity Detection:** given an audio input it is possible to detect if the voice recorded in the audio is authentic or is a replay attack spoof performed by reproducing a voice through a smartphone speakers.
- **Different Calibration modes:** Include different calibration to obtain better verification/identification results depending on the use case (Telephone channel, Lossless audio, without calibration). If the use case is the telephone audio, the customer must use the parameter calibration="telephone-channel". If the customer uses SDKs for audio capture, the parameter calibration="lossless-audio" should be added.

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**: 8 kHz or 16 kHz.
- **Do not use lossy audio encoding:** audio encodings such as mp3 can degrade the performance of the voice biometry engine. Converting mp3 to wav is not recommended because the lossy encoding is retained.
- **Do not use audios with multiple voices to register/verify**: If there are multiple voices in the audio, the voice credential created will contain multiple voice information and this will negatively affect future verification processes.
- **Use “lossless-audio” calibration:** To obtain correct verification/identification results working with SDKs, it is necessary to use this calibration.

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

3. System quality report

3.1. Verification performance (Telephone Channel)

VERIDAS has evaluated its voice biometrics model (das-Peak 2020Q3 version) with an internal Telephone audio 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 these 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	0.66	1	0.36	1	0.08
0.55	0.73	0.79	0.73	0.48	0.76	0.23
0.6	0.52	0.99	0.53	0.62	0.56	0.32
0.65	0.36	1.23	0.37	0.74	0.38	0.4
0.7	0.24	1.74	0.25	0.85	0.27	0.49
0.75	0.17	2.39	0.16	1.05	0.17	0.59
0.8	0.1	3.14	0.1	1.4	0.1	0.68
0.85	0.05	4.5	0.05	2.13	0.05	0.9
0.9	0.02	7.42	0.01	3.44	0.01	0.99
0.95	0.0012	15.04	0.0014	7.57	0.0012	2.32

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

For example, if the use case is 5 seconds to enroll and 5 seconds to verify, with a threshold equal to 0.6, it is obtained FPR=0.52% and FNR=0.99%. In this case, 99.01% of the comparisons of a person's voice and its corresponding voice registration will be considered as the same person, and only 0.52% of the cases comparing voices to different persons will be incorrectly classified as the same person.

3.2. Verification performance (Lossless Audio)

VERIDAS has evaluated its voice biometrics model (das-Peak 2020Q3 version) with Voxceleb 2 test database with different duration enrollment audios and different duration audios test (253.188.434 comparisons) in different acoustic conditions obtaining the values of False Positive Rates (FPR) and False Negative Rates (FNR) with different threshold values. Due to the elevated number of speakers, environments and speaker session variability, the FNR is worse than the telephone channel.

The FPR is the probability to accept a non legit person and the FNR is the probability to reject a legit person. With these 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	6.38	1	4.73	1	3.23
0.55	0.73	7.55	0.73	5.64	0.76	3.9
0.6	0.52	9.09	0.53	6.78	0.56	4.76
0.65	0.36	10.87	0.37	8.18	0.38	5.91
0.7	0.24	12.98	0.25	10.01	0.27	7.25
0.75	0.17	15.65	0.16	12.3	0.17	8.93
0.8	0.1	19.14	0.1	15.17	0.1	11.32
0.85	0.05	23.69	0.05	19.29	0.05	14.64
0.9	0.02	30.06	0.01	25.39	0.01	20.03

0.95	0.0012	40.79	0.0014	36.08	0.0012	29.85

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

For example, if the use case is 5 seconds to enroll and 5 seconds to verify, with a threshold equal to 0.6, it is obtained FPR=0.52% and FNR=9.09%. In this case, 90.91% of the comparisons of a person's voice and its corresponding voice registration will be considered as the same person, and only 0.52% of the cases comparing voices to different persons will be incorrectly classified as the same person.

3.3. Identification performance

VERIDAS has evaluated its voice biometrics model (das-Peak 2020Q3 version) with a internal database with 256 speakers and 5904 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 1.000 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 **97.64 %**. 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 (%)	99.53	99.19	98.88	98.65	98.44	98.2	97.98	97.86	97.64

3.4. Voice Authenticity (Anti-spoofing)

The API also includes an anti-spoofing detector model designed to avoid fraudulent access. Given a wav file with voice, the detector estimates the likelihood of the voice authenticity. The performance of the system has been measured for the more important and common spoofing technique: replay attack (a voice capture from a device speaker) through smartphone speakers.

The evaluation has been performed on another internal database with 140094 authentic audios and 109484 spoofing audios, achieving a 98.62% accuracy. The anti-spoofing performance is shown in the following table. Notice the performance of the system is shown for different authenticity thresholds, i.e., 1.00 means authentic and 0.00 means spoof attempt.

Based on the following Table, with an operation point at 0.60, the 0.42% of authentic cases will be rejected, and the 2.6% of replay-attack spoofing attempts will be correctly misclassified as authentic.

Authenticity Threshold	FPR(%)	FNR(%)
0.05	23.97	0.0028
0.1	12.33	0.0056
0.15	8.25	0.0158
0.2	6.18	0.028
0.25	5.06	0.057
0.3	4.33	0.085
0.35	3.82	0.107
0.4	3.42	0.138
0.45	3.12	0.177
0.5	2.91	0.23
0.55	2.74	0.31
0.6	2.60	0.42
0.65	2.50	0.54
0.7	2.43	0.75
0.75	2.35	1.07
0.8	2.29	1.64
0.85	2.26	2.80
0.9	2.21	5.93
0.95	2.15	19.04

4. Voice Verification Use Case

This section describes a voice verification use case, as an example. The das-Peak service can receive queries for a registration or verification request. The enrollment and verification processes are explained below.

4.1 Enrollment

During the enrollment process, the customer is asked for an audio recording by reading a random phrase. This audio can be captured through the voice capture SDK provided by VERIDAS (available on iOS and Android platforms) or by other means. **The reading of this phrase is done only once and takes a few seconds** (about 5s).

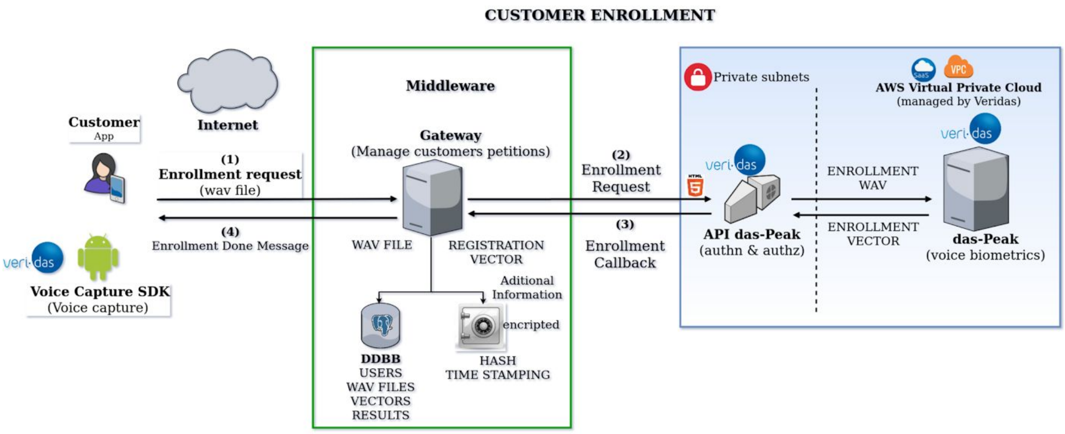
Voice capture SDKs do not perform any communication process with third parties, but delegate the recorded content to the APP invoking the SDK. This allows das-Peak's client to fully control the transfer of its customers' data from the mobile device to the back-end.

After that, the APP communicates the recorded audio to the das-Peak client middleware. This middleware communicates with the VERIDAS cloud to obtain the biometric vector of the new registered user.

Once the biometric vector is received, das-Peak's client stores the customer biometric information in its database:

- On the one hand, the biometric vector returned by das-Peak.
- On the other hand, the registration audio itself. This will allow the biometric vector to be recalculated when improvements are made to the VERIDAS cloud biometric engine.
- It is also necessary to save the hash of the biometric model used to generate the voice credential. This information is very useful to check if the voice credential needs to be updated.

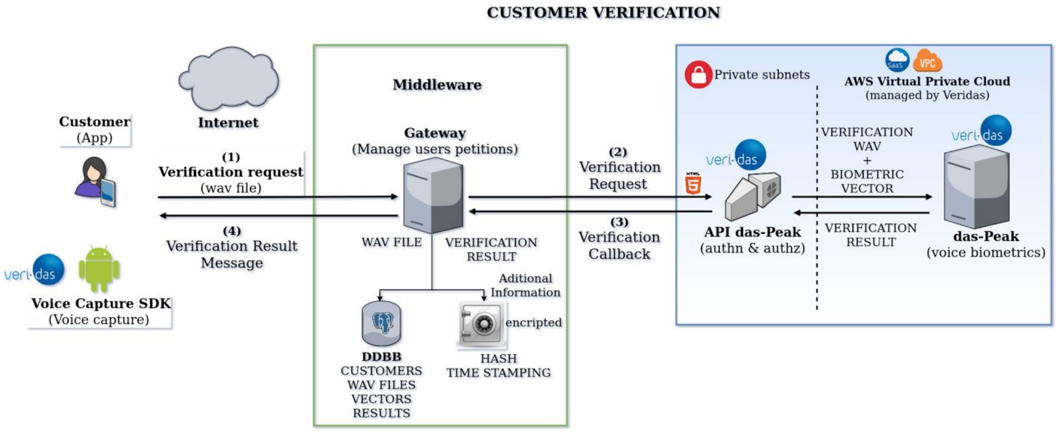
A proposed architecture scheme for the biometric registration process is presented below.



4.2 Verification

The verification process can be carried out from customer’s mobile APP. As in the enrollment process, it is possible to use VERIDAS voice capture SDKs. After capturing and sending to the middleware the voice captured in the verification process, the client sends to the das-Peak cloud service the content captured during the verification, as well as the biometric vector of that user that was previously stored in the client database during the enrollment phase. When using the capture SDKs, it is necessary to add the calibration parameter "lossless audio" to optimize the result of the biometric comparison.

As a result of this operation, das-Peak returns to the middleware the result of the biometric comparison, obtaining a decimal value between 0 and 1. The higher the score, the greater the certainty to be the same person. A proposed architecture scheme for the biometric verification process is presented below.



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 or audio in base64 code.
- 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.
GET	/models	Get a list hashes of all available models in das-Peak service.
POST	/models/metadata	Provides metadata information about a specific hash model.
POST	/models/calibration	Get info about calibration modes that a specific model supports.
POST	/models/metadata/from-credential	Provides metadata information about a specific voice credential, among others, model hash used to create it.
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 highest score ID and a list of all results. A result contains credential ID and its score.
POST	models/{hash}/credential/wav	Generate a biometric credential from a given audio input, using the hash model. It is possible to obtain hash with endpoint /models.

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

Example:

https://api-work.eu.veri-das.com/daspeak/v1/alive

Response: 200

Empty response.

Response: 500

Server error response.

Content-Type: application/json

exception	error message
ServerError	Unexpected server fatal error

6.2. Get list of available models

The service receives a GET request with no params, and returns the available embedding model hashes.

GET /models

Example:

https://api-work.eu.veri-das.com/daspeak/v1/models

Response: 200

Returns list of models sorted by their numeric tag (descending order).

Example:

```
{
 "version": "1.0",
 "models": [
  "1b40a9b479b131e7acb9cab797f929e28eb5dffac86ce1d71d83c564",
  "38da15f1b61fb5800c5928f6f1437aed7a0b0e7921fa6bb7852c5783"
 ]
}
```

Response: 500

Server error response.

Content-Type: application/json

exception	error message
ServerError	Unexpected server fatal error

6.3. Obtain metadata from a model

This endpoint is used to obtain the metadata info of a model.

POST `/models/metadata`

Request Body

Request for model metadata information.

Name	Req.	Type	Description

hash	yes	string	A hash given as hex digest, same as returned by {GET /models}

Example:

https://api-work.eu.veri-das.com/daspeak/v1/models/metadata"

Response: 200

Returns metadata info of a model given its hash.

Content-Type: application/json

Name	Req.	Type	Description
version	Yes	string	API version
metadata	yes	Dictionary	returns “hash”:model hash input and “description”: model information.

Example:

```
{
  "version": "1.0",
  "metadata": {
    "hash": "1b40a9b479b131e7acb9cab797f929e28eb5dffac86ce1d71d83c564",
    "description": "Speaker Recognition model created on 10/02/2020 that works with 8 kHz audio"
  }
}
```

Response: 400

Error: bad request.

Content-Type: application/json

exception	error message
error	The model is not available.
exception	ModelNotAvailable

6.4. Obtain calibration modes of a model

This endpoint is used to obtain the info about calibration modes that a specific model supports.

POST `/models/calibration`

Request Body

Request for model metadata information.

Name	Req.	Type	Description
hash	yes	string	A hash given as hex digest, same as returned by {GET /models}

Example:

https://api-work.eu.veri-das.com/daspeak/v1/models/calibration"

Response: 200

Returns calibration modes of a model given its hash.

Content-Type: application/json

Name	Req.	Type	Description
version	yes	string	API version
calibrations	yes	Dictionary	Results calibration models that this model supports. Different calibration types are: [“lossless-audio”,”telephone-channel”,”no-calibration”]

Example:

```
{
  "version": "1.0",
  "calibrations": [
    "lossless-audio",
    "telephone-channel",
    "no-calibration"
  ]
}
```

Response: 400

Error: bad request.

Content-Type: application/json

exception	error message
error	The model is not available.
exception	ModelNotAvailable

6.5.Obtain metadata from a voice biometric credential

This endpoint is used to obtain the info about metadata from a given voice credential.

```
POST /metadata/from-credential
```

Request Body

Request for voice credential metadata information.

Name	Req.	Type	Description
credential	yes	string	voice credential returned by {credential/wav}

Example:

https://api-work.eu.veri-das.com/daspeak/v1/metadata/from-credential"

Response: 200

Returns metadata information of a given voice credential.

Content-Type: application/json

Name	Req.	Type	Description
version	yes	string	API version
metadata	yes	Dictionary	returns “hash”: model has been used to create the credential and “description”: model information.

Example:

```
{
  "version": "1.0",
  "metadata": {
    "hash": "1b40a9b479b131e7acb9cab797f929e28eb5dffac86ce1d71d83c564",
    "description": "Speaker Recognition model created on 10/02/2020 that works with 8 kHz audio"
  }
}
```

Response: 400

Error: bad request.

Content-Type: application/json

exception	error message
error	Incorrect Padding.
exception	ValueError

6.6. 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.

Request header

Request for voice authenticity checking.

Name	Req.	Type	Description
X-Check-Authenticity	No	Header	This header activates the voice authenticity check. If it is present in the query, in the response there will be a authenticity value in range [0,1] indicating if the voice is authentic (close 1), or a spoof attack (close to 0).

X-Veridas-RTag	No	Header	This header allows to tagger this query with a personal string. This tag needs to be a ASCII string with a maximum size of 64 characters.

example:

https://api-work.eu.veri-das.com/daspeak/v1/credential/wav

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	API version
model	yes	Dictionary	Dictionary with hash and mode fields, information about the model used to generate the voice credential.
authenticity	No	number	voice authenticity in range [0,1]. when this number is close to 1 indicates it is authentic and, if it is close to 0, it means spoof.

Example:

```
{
 "credential": "APUhOWxROgu",
 "model": {
  "hash": "e7acb9cab797f929e0",
  "mode": "speaker-recognition-8k"
 },
 "version": "1.0"
 "authenticity": 0.98036
}
```

Response: 400

Request format error.

Content-Type: application/json

exception	error message

InputException	The wav is not mono
InputException	The sample rate is not supported, must be 8 Khz or 16 Khz
InputException	The bits per sample is not supported, must be 16 bits/sample
InputException	The wav duration is longer than 30 seconds
SignalNoiseRatioException	Noise level exceeded
VoiceDurationIsNotEnoughException	Voice duration is less than 3 seconds

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.

6.7. Compute similarity between voice biometric credential and audio input

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

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 das-Peak service.
audio_to_evaluate	yes	WAV file	Audio to evaluate with the speaker voice. As multipart/form-data, it should be a WAV file, as

			application/json, the file content encoded in base64.
calibration	no	string	Calibration type to compare credential and audio input. The calibration modes are: [“lossless-audio”,”telephone-channel”,”no-calibration”]. By default “telephone-channel”. If SDKs are used for audio capturing, it is necessary to use “lossless-audio”.

Request header

Request for voice authenticity checking.

Name	Req.	Type	Description
X-Check-Authenticity	No	Header	This header activates the voice authenticity check. If it is present in the query, in the response there will be an authenticity value in range [0,1] indicating if the voice is authentic (close 1), or a spoof attack (close to 0).
X-Veridas-RTag	No	Header	This header allows to tagger this query with a personal string. This tag needs to be a ASCII string with a maximum size of 64 characters.

Example:

https://api-work.eu.veri-das.com/daspeak/v1/similarity/credential2wav

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
score	yes	number	A probability number in range [0,1]
version	yes	string	API version
model	yes	Dictionary	Dictionary with hash and mode fields
authenticity_to_evaluate	No	number	audio voice authenticity in range [0,1]. when this number is close to 1 indicates it is authentic and, if it is close to 0, it means spoof.

Example:

```
{
 "score": 0.91,
 "model": {
  "hash": "1b40a9b479b131ef929",
  "mode": "speaker-recognition-8k"
 },
 "version": "1.0"
 “authenticity_to_evaluate”: 0.98036
}
```

Response: 400

Request format error.

Content-Type: application/json

exception	error message
InputException	The wav is not mono
InputException	The sample rate is not supported, must be 8 Khz or 16 Khz
InputException	The bits per sample is not supported, must be 16 bits/sample
InputException	The wav duration is longer than 30 seconds
SignalNoiseRatioException	Noise level exceeded
VoiceDurationIsNotEnoughException	Voice duration is less than 3 seconds

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.

6.8. Compute similarity between two audio inputs

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 WAV 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 WAV file, as application/json, the file content encoded in base64.
calibration	no	string	Calibration type to compare credential and audio input. The calibration modes are: [“lossless-audio”,”telephone-channel”,”no-calibration”]. By default “telephone-channel”. If SDKs are used for audio capturing, it is necessary to use “lossless-audio”.

Request header

Request for voice authenticity checking.

Name	Req.	Type	Description

X-Check-Authenticit y	No	Header	This header activates the voice authenticity check. If it is present in the query, in the response there will be an authenticity value in range [0,1] indicating if the voice is authentic (close 1), or a spoof attack (close to 0).
X-Veridas-RTag	No	Header	This header allows to tagger this query with a personal string. This tag needs to be a ASCII string with a maximum size of 64 characters.

Example:

https://api-work.eu.veri-das.com/daspeak/v1/similarity/wav2wav

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
score	yes	number	A probability number in range [0,1]
version	yes	string	API version
model	yes	Dictionary	Dictionary with hash and mode fields
authenticity_reference	No	number	voice authenticity of the audio reference in range [0,1]. when this number is close to 1 indicates it is authentic and, if it is close to 0, it means spoof.
authenticity_to_evaluate	No	number	voice authenticity of the audio to evaluate in range [0,1]. when this number is close to 1 indicates it is authentic and, if it is close to 0, it means spoof.

Example:

```
{
 "score": 0.81,
 "model": {
  "hash": "1b40a9b479b131e7acb",
  "mode": "speaker-recognition-8k"
 },
 "version": "1.0"
 "authenticity_reference": 0.98036
 "authenticity_to_evaluate": 0.9915
}
```

Response: 400

Request format error.

Content-Type: application/json

exception	error message
InputException	The wav is not mono
InputException	The sample rate is not supported, must be 8 Khz or 16 Khz
InputException	The bits per sample is not supported, must be 16 bits/sample
InputException	The wav duration is longer than 30 seconds
SignalNoiseRatioException	Noise level exceeded
VoiceDurationIsNotEnoughException	Voice duration is less than 3 seconds

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.

6.9. Compute voice identification between a wav input and a list of voice 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 WAV file, as application/json, the file content encoded in base64.
credentials_list	yes	Dictionar y	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"
calibration	no	string	Calibration type to compare credential and audio input. The calibration modes are: ["lossless-audio","telephone-channel","no-calibration"]. By default "telephone-channel". If SDKs are used for audio capturing, it is necessary to use "lossless-audio".

Request header

Request for voice authenticity checking.

Name	Req.	Type	Description
X-Check-Authenticity	No	Header	This header activates the voice authenticity check. If it is present in the query, in the response there will be a authenticity value in range [0,1] indicating if the voice is authentic (close 1), or a spoof attack (close to 0).
X-Veridas-RTag	No	Header	This header allows to tagger this query with a personal string. This tag needs to be a ASCII string with a maximum size of 64 characters.

Example:

https://api-work.eu.veri-das.com/daspeak/v1/identification/wav2credentials

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.
version	yes	string	API version
model	yes	Dictionary	Dictionary with hash and mode fields
authenticity	No	number	voice authenticity in range [0,1]. when this number is close to 1 indicates it is authentic and, if it is close to 0, it means spoof.

Example:

```
{
 "result": {
  "id": "User001",
  "score": 0.90
 },
 "scores": [
  {
   "id": "User001",
   "score": 0.90
  }
  {
   "id": "User002",
   "score": 0.10
   }

 ],
 "model": {
  "hash": "1b40a9b479b131ef929",
  "mode": "speaker-recognition-8k"
 },
 "version": "1.0"
  "authenticity": 0.98036

}
```

Response: 400

Request format error.

Content-Type: application/json

exception	error message

InputException	The wav is not mono
InputException	The sample rate is not supported, must be 8 Khz or 16 Khz
InputException	The bits per sample is not supported, must be 16 bits/sample
InputException	The wav duration is longer than 30s
InvalidAudio	Exceeded Noise Level
InvalidAudio	The duration of the voice is not enough
InputException	Length of credentials list exceeds the maximum allowed. (Nmax=10).

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.

6.10. Generate a voice biometric credential with a specific model

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

POST `models/{hash}/credential/wav`

Parameters

Name	In	Description	Example
hash	path	A hash given as hex digest, same as returned by {GET /models}	1b40a9b479b131e7acb

Example:

*https://api-work.eu.veri-das.com/daspeak/v1/*models/{hash}/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.

Request header

Request for voice authenticity checking.

Name	Req.	Type	Description
X-Check-Authenticity	No	Header	This header activates the voice authenticity check. If it is present, in the query will be a authenticity probability [0,1] indicating if the voice is authentic being 1, or if correspond to a spoof attack close to 0.
X-Veridas-RTag	No	Header	This header allows to tagger this query with a personal string. This tag needs to be a ASCII string with a maximum size of 64 characters.

Response: 200

Returns one voice biometric credential for the given audio file, API version,dictionary with hash and mode of the model used to create it.

Content-Type: application/json

Name	Req.	Type	Description
credential	yes	string	A biometric credential string
version	yes	string	API version
model	yes	Dictionary	Dictionary with hash and mode fields
authenticity	No	number	voice authenticity in range [0,1]. when this number is close to 1 indicates it is authentic and, if it is close to 0, it means

			spoof.

Example:

```
{
 "credential": "APUhOWxROgu",
 "model": {
  "hash": "e7acb9cab797f929e0",
  "mode": "speaker-recognition-8k"
 },
 "version": "1.0"
 "authencitity": "0.9830"

}
```

Response: 400

Request format error.

Content-Type: application/json

exception	error message
InputException	The wav is not mono
InputException	The sample rate is not supported, must be 8 Khz or 16 Khz
InputException	The bits per sample is not supported, must be 16 bits/sample
InputException	The wav duration is longer than 30 seconds
SignalNoiseRatioException	Noise level exceeded
VoiceDurationIsNotEnoughException	Voice duration is less than 3 seconds

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.

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.