PerSay's Position Paper

For

The Speaker biometrics and VoiceXML 3.0 Workshop

Author:

Michael Salmon

VP of Product Development

PerSay

1. Introduction

PerSay is a leading vendor of Voice Biometric platforms.

In the last 8 years PerSay has deployed dozens of voice biometric applications, based on its VocalPassword product.

VocalPassword is a product that was designed to provide accurate and secured voice biometric services for self service applications usually in the IVR (but also for web services).

It uses Text Dependant, Text Independent and Text Prompted technologies to provide Verification, Identification and Fraud detection services.

PerSay has very rich experience in Voice Biometric deployments and as such see itself obligated to share this experience in order to assist defining standards that will address the market needs.

2. Integration

1.1 Voiceprints data interoperability

PerSay's position is that the only biometric data that should be transferred between the client (IVR) and the server (SIV engine/platform) is the raw audio recordings.

Therefore PerSay thinks it is irrelevant to define a standard way to transfer the voiceprints themselves.

The reason is that the voiceprints' data is different for any vendor. This means that the voiceprints themselves cannot be used by any other engine or platform so there is no use to standardize the way they are transferred.

1.2 Audio transferring

Any SIV standard should support 2 methods of audio transferring:

- Inline – In most cases the verification is done over a relatively short recording of the speaker saying few words/phrases or digits. It would be most convenient in those cases to enable recording the speech into file or to memory and when the recording stops to send the speech either as a link to a file or as a buffer for processing.
- Streaming – In few cases (especially with text independent engines) the verification requires longer speech recordings, and it might be that all the audio in that session will be required. In this case it is required that the protocol will support continuous streaming of audio to the SIV engine by sending chunks of audio buffers to the SIV server

Commands such as Start, Pause, Resume and Stop should control the audio's streaming.

1.3 Software programming interface

PerSay thinks that defining a standard API to use voice biometric is very important because of the following reasons:

- Standard API could be used in non-telephony environment as well. This way the same module can be used to verify speaker when he speaks to an IVR, surfs the web or uses his voice recorded by his mobile phone to access a service.
- An IVR vendor could use this standard SIV API to integrate its IVR with the SIV engine/platform to implement the VXML 3.0 SIV tags; it can later use the same integration code to switch to another vendor.
- Standard API could significantly shorten the time required to test and compare different SIV vendors. Currently in many project the phase that takes most of the time is the evaluation, which is often done off-line with pre-recorded audio files, a standard API could be used to implement tools to perform experiments on multiple SIV engines with no need to integrate separately with every engine.
- Our experience shows that integrating using web service is very easy and takes relatively short time to complete. It can be said that it will take significantly less time to the average programmer to integrate SIV engine using web service than to integrate it using VoiceXML.

Since web service (SOAP) is currently the de-facto standard protocol for application's programming interface, PerSay thinks that the standard API should be based on the Web Service protocol.

Here are few examples of elements that should be included in such an API:

- Enroll - Add audio sample to the samples collection (used for training a voiceprint)
- Train a voiceprint
- Check if speaker is ready to train
- Check if speaker is already trained
- Adapt the voiceprint with new recording
- Verify – should return score and decision and optionally additional information about the reasons for the decision.
- Identify – 1 to many comparison on a predefined list (such as sharers of the same bank account)
- Identify - on a dynamic list of speakers (for example: The speaker is asked to say his name, Identify will be performed on the best N results of the ASR engine).
- Watch list identify.
- Manage group of speakers for identification, this includes create/delete group, add/remove speaker from group, get the list of the group's members etc.

There are many parameters that define how enrollment/verify or identify should be performed, as for instance, the decision threshold(s) to be used, and often more than one set of parameters is used in parallel. For example: For risky transactions high thresholds should be used and for less risky transactions lower thresholds should be used.

We can assume that these parameters will be different for every SIV vendor and it would be impossible to standardize them.

In order to simplify it PerSay recommends that in the API the user would be able to define which predefined set of configuration parameters he wants to use, this way instead of defining many parameters for each API method, one parameter only would be used to refer to a given parameters set.

The API must be session aware so the same audio can be sent and processed once but be used more than once in that same session, for example the audio can be compared to voiceprints of known fraudsters before used for enrollment.

1.4 MRCPv2

MRCP is a network protocol rather than a programming API, it cannot be used in non-telephony environments as it is based on the SIP protocol.

The fact is uses RTP to transfer the audio makes it much harder for implementation because in most cases there is no need for real time audio streaming but rather simple recording to a file or memory and then sending it as one block of data for processing.

Although this protocol is around for few years, in all the dozens of integrations PerSay have supported and performed over the years the using of MRCP was not even considered.

In order to integrate VocalPassword with an IVR all it is required is to call 4-5 basic web service (SOAP) methods. This can be done in no-time, but implementing SIP, RTP etc. is much more complicated.

PerSay opinion is that this protocol cannot be used as the standard SIV programming API because it will not be adopted widely.

Using RTP as a method of audio acquisition should be supported by the programming API as another option besides audio file reference and audio buffer

2 Multi factor Decision

PerSay thinks that in most of the applications voice biometric decision would be used in conjunction with additional factors to obtain the most reliable decision.

It could be few Voice Biometric results that should be weighted (such as Text dependant and Text prompted verifications), it could be a non biometric information such as CLI or verified information provided by the caller, or any combinations of them.

We think that the SIV engine is not the ideal component to weight multi authentication factors; however SIV platform may include a “Decision Engine” component.

We think that another standard API should be defined to standardize the “Decision Engine” interface, so any SIV vendor can implement this interface (PerSay today has its own “Decision Engine” interface) or to interact with external “Decision Engine” using this standard interface. It is probably out of the scope of this position paper to suggest definition to this interface.

3 Security

PerSay’s experience shows that the customers normally want the voiceprints to be stored in the SIV platform’s database and not in their databases.

This approach makes the integration much easier and faster.

According to this approach, the question how to protect the biometric voiceprints in out of the scope of the VoiceXML protocol or any other protocol that is used to integrate SIV engines/platforms to external applications.

The security mechanisms should meet pure security standards as any other sensitive data stored in databases such as credit card numbers.

Although credit cards can be cancelled and replaced unlike voice, it is important to mention that the voiceprints themselves cannot be reverse engineered so the enrollment audio cannot be extracted from them and also not the identity of the speaker owning it, so by “stealing” the voiceprint one cannot access your account.

It is recommended to encrypt the voiceprint data using a key that contains the speaker ID. This way even if someone has access to the database he cannot switch voiceprints between speakers.