



# **Speaker Identification and Verification (SIV) Requirements for VoiceXML Applications**

**Draft Version 2.0 - May 22, 2007**

**Editors:**

Claudia Daboul, T-Systems  
Pallavi Shinde, Tuvox

**VoiceXML Forum Speaker Biometrics Committee**

Judith Markowitz and Ken Rehor, Co-Chairs

<http://www.voicexml.org/biometrics>

## About the VoiceXML Forum

Voice Extensible Markup Language (VoiceXML) is a markup language for creating voice user interfaces that use automatic speech recognition (ASR) and text-to-speech synthesis (TTS). Since its founding in March 1999, the VoiceXML Forum has continued to develop, promote and to accelerate the adoption of VoiceXML-based technologies via more than 150 member organizations worldwide.

Tens of thousands of commercial VoiceXML-based speech applications have been deployed across a diverse set of industries, including financial services, government, insurance, retail, telecommunications, transportation, travel and hospitality. Millions of calls are answered by VoiceXML applications every day.

The Forum's primary focus areas include:

- Promoting the adoption of VoiceXML-based technologies
- Cultivating a global VoiceXML ecosystem
- Actively supporting standards bodies and industry consortia, such as the W3C and IETF, as they work on VoiceXML and related standards, such as CCXML, X+V, MRCP, and speech biometrics.

For more information on the VoiceXML Forum visit the website at <http://www.voicexml.org>.

## Disclaimers

This document is subject to change without notice and may be updated, replaced or made obsolete by other documents at any time.

The VoiceXML Forum disclaims any and all warranties, whether express or implied, including (without limitation) any implied warranties of merchantability or fitness for a particular purpose.

The descriptions contained herein do not imply the granting of licenses to make, use, sell, license or otherwise transfer any technology required to implement systems or components conforming to this specification. The VoiceXML Forum, and its member companies, makes no representation on technology described in this specification regarding existing or future patent rights, copyrights, trademarks, trade secrets or other proprietary rights.

By submitting information to the VoiceXML Forum, and its member companies, including but not limited to technical information, you agree that the submitted information does not contain any confidential or proprietary information, and that the VoiceXML Forum may use the submitted information without any restrictions or limitations.

## Revision History

Date	Description
Sept 14, 2005	Initial public draft
April 17, 2006	Revised Draft version 1.1
May 10, 2006	Revised Draft version 1.2
September 15, 2006	Revised Draft version 1.3
May 22, 2007	Version 2.0, published on VoiceXML Forum Website

## Contributors

The following people contributed content and reviews of this document

Dave Armstrong, Authentify  
Homayoon Beigi, Recognition Technologies  
Chuck Johnson, AnyTransactions  
Arvind Kizhanatham, Diaphonibics  
Martin Eckert, T-Systems  
Jim Larson, W3C Voice Browser Working Group  
Judith Markowitz, J. Markowitz Consultants  
Brian Novack, AT&T  
Ken Rehor, independent consultant  
Michael Salmon, Persay  
Rajesh Sharma, Verizon  
Valene Skerpac, iBiometrics  
Cathy Tilton, Daon  
Jim White, Nuance Communications  
Ran Zilca, independent consultant

## Table of Contents

1 Purpose .....	6
2 Scope.....	6
2.1 What This Work Covers .....	6
2.2 What This Work Does Not Cover .....	6
2.2.1 Other Biometrics .....	6
2.2.2 Non-Biometric Security .....	7
2.2.3 Privacy .....	7
2.2.4 Security.....	7
3 Relationship with Other Standards.....	8
3.1 SIV as Standalone and an Extension of VoiceXML.....	8
3.2 MRCP v1 and v2 .....	9
3.3 Biometric Standards .....	10
4 Requirements.....	13
4.1 Basic Functions and Properties.....	13
4.1.1 Basic Functions .....	13
4.1.2 Basic Properties of Voice Models .....	14
4.1.3 Generic Speaker Verification Properties.....	14
4.2 SIV Sessions .....	17
4.2.1 Phases of an SIV Session .....	17
4.2.2 Start and End of an SIV Session .....	18
4.2.3 Decision Making Algorithms.....	18
4.3 Interaction Turns / Concurrent SIV/ASR Processing.....	20
4.3.1 Input Items .....	21
4.3.2 Completion Events.....	21
4.4 Control of Audio Processing .....	22
4.4.1 Pause and Resume .....	22
4.4.2 Rollback.....	22
4.4.3 Misrecognized Audio .....	23
4.4.4 Enrolled Phrases .....	23
4.5 Return Results.....	23
4.7 Concurrent SIV Sessions .....	24
4.8 Nested SIV Sessions.....	24
4.9 Multiple Voice Models per Speaker .....	25
4.10 Multi-Factor Systems and Applications .....	25
5 Use Cases .....	26
5.1 Enrollment .....	27
5.1.1 Successful Initial Enrollment (Text Independent).....	27
5.1.2 Unsuccessful Initial Enrollment (Text Independent).....	28
5.1.3 Initial Enrollment (Text Dependent) on the Identity .....	29
5.1.4 Initial Enrollment (Text Dependent) Single Passphrase.....	29
5.1.5 Initial Enrollment (Text Dependent) Multiple Phrases .....	30
5.1.6 Initial Enrollment (Text Prompted/Challenge-Response) .....	31
Enrollment for Group Authentication.....	31
5.2 Verification.....	31
5.2.1 Text-Independent Verification .....	31

5.2.2 Ongoing Verification .....	32
5.2.3 Text-Prompted Verification .....	33
5.2.4 Text-Dependent Verification .....	33
Other verification examples .....	34
5.3 Group Authentication.....	34
Other Group Authentication Examples .....	34
5.4 Adaptation .....	34
5.4.1 Simultaneous Verification and Reference Model Adaptation .....	34
5.4.2 Supervised Adaptation with Low Scoring Samples .....	35
5.5 The Database Function IsEnrolled .....	36
5.6 SIV Processing on Recorded Speech .....	36
5.7 Buffering.....	37
5.8 Concurrent SIV Processing .....	37
5.8.1 SIV Sessions Tied to ASR Result Alternatives .....	37
5.8.2 Combining Different SIV-Engines in Concurrent Sessions .....	39
5.9 Nested SIV Sessions.....	39
5.10 Rollback.....	40
5.10.1 Text-Prompted Verification with Rollback .....	40
5.10.2 Text-Prompted Verification with Replay Attack .....	41
5.11 Return Results.....	41
5.12 Multi-Factor Authentication.....	45
5.12.1 SIV Plus Knowledge .....	45
5.12.2 Multi-Factor and Multi-Modal Authentication .....	46
6 References .....	46
6.1 Related Documents.....	46
6.2 External References.....	47

## Table of Figures

Figure 3.1 Relationship of SIV Extension to VoiceXML .....	8
Figure 3.2 Tags for SIV as part of VoiceXML and standalone .....	9
Figure 3.3 Relationship of SIV Extension to MRCP .....	10
Figure 3.4 Relationship of SIV and generic biometric standards in a VoiceXML application.....	11
Figure 3.5 Relationship of SIV Extension to Biometric Standards in a BioAPI Application .....	12

# 1 Purpose

The purpose of this document is to provide a set of requirements for Speaker Identification and Verification (SIV) systems that can be used, but are not limited, to create an SIV specific extension to the VoiceXML language. The requirements herein are supported by a comprehensive listing of use cases that demonstrate how the required features can be used.

This document is the lead document of a series of related work by the VoiceXML-Forum's Speaker Biometric Committee (SBC). The other documents of this series are listed in Section 6.1.

The present document assumes prior knowledge of SIV technology. An introduction to SIV technology can be found in the related document [Introduction]. A glossary of SIV-related terminology is published as a separate document [Glossary].

## 2 Scope

### ***2.1 What This Work Covers***

The focus of this document is on SIV. These requirements cover the following speaker biometric modalities:

- Speaker Classification
- Speaker Verification
- Group Authentication
- Closed Set Speaker Identification
- Open Set Speaker Identification

and the following basic types of SIV systems:

- Text independent
- Text dependent/constrained
- Text prompted (challenge-response)

### ***2.2 What This Work Does Not Cover***

#### **2.2.1 Other Biometrics**

We acknowledge the growing use of SIV with other biometrics and view their incorporation into an SIV application as the fusion of multiple inputs. We see this as related to SIV use cases (see sections 5.12.1 "*SIV Plus Knowledge*" and 5.12.2 "*Multi-Factor and Multi-Modal Authentication*") rather than support of specific non-SIV technologies.

## 2.2.2 Non-Biometric Security

This specification does not attempt to address the general issue of authentication but it is compatible with standards governing those non-SIV security measures. The use of non-biometric security, such as PINs, passwords, and tokens may, however, contribute to the final decision made by an SIV application. (see section 5.12 *Multi-Factor Authentication*)

## 2.2.3 Privacy

Privacy is a complex and emotion-laden topic. It is related to law, culture, and other non-technical arenas. The recording and subsequent protection of spoken audio is of particular concern. Both developers and organizations must maintain an awareness of legislation in this and related areas. Methods to address these issues are beyond the scope of this document.

A good privacy-related reference web site is from the International Biometric Group's BioPrivacy Initiative (<http://www.bioprivacy.org/>). We also recommend that developers consult the Privacy Principles of the International Biometric Industry Association (<http://www.ibia.org>); Directive 95/46/EC, the data-privacy protection legislation enacted by the European Union ([http://europa.eu.int/comm/justice\\_home/fsj/privacy/index\\_en.htm](http://europa.eu.int/comm/justice_home/fsj/privacy/index_en.htm)) and privacy regulations and legislation in their own countries and localities.

## 2.2.4 Security

SIV systems provide a form of user identification and authentication to support a variety of applications with different security requirements. The proper use of SIV requires that the developer and implementing organization adhere to generally accepted security best practices. This includes performing risk and threat assessments along with the development and maintenance of policies and procedures to mitigate vulnerabilities. SIV must preserve the confidentiality, integrity and availability of sensitive information contained in the application, system, platform, and network levels.

The separation of a SIV reference model from its identifying information, data encryption, watermarks, date stamps, and liveness tests are just a few of the methods that can improve the security of SIV systems and data against a variety of vulnerabilities.

Documents specific to biometrics and voice security include:

*ANSI X9.84 Biometric Information Management and Security for the Financial Services Industry* (<http://www.ansi.org>)

Common Criteria Biometric Evaluation Methodology Working Group  
*Biometric Evaluation Methodology: Common Criteria Common Methodology for Information Technology Security Evaluation*  
(<http://www.cesg.gov.uk/>).

ISO 19092 *Financial services - Biometrics - Part 1: Security framework*  
(<http://www.ansi.org>)

NIST Special Publication 800-58, *Security Considerations for Voice Over IP Systems*, (<http://csrc.nist.gov/publications/nistpubs>).

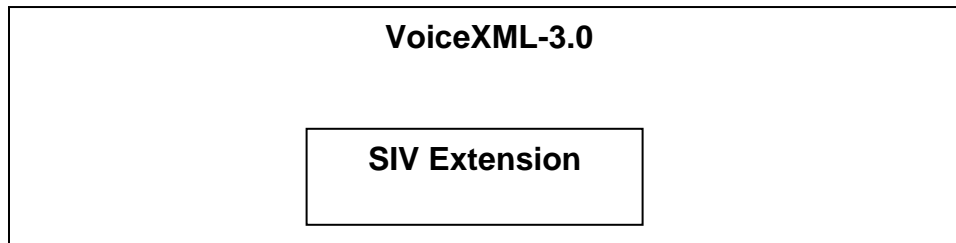
## 3 Relationship with Other Standards

The specification that will be developed from the requirements herein must support inter-operability with the other standards with which it will be used. The authors of this document (who include representatives from biometrics, speech processing and other interested industries) have examined high-level correspondence between these requirements and the following standards: VoiceXML, MRCP, CBEFF and BioAPI.

The group that creates the SIV specifications is charged with performing a more in-depth comparison with these and other relevant standards that identifies inconsistencies and resolves them.

### 3.1 SIV as Standalone and an Extension of VoiceXML

This document focuses on requirements for an SIV module as part of the upcoming version 3.0 of VoiceXML.



**Figure 3.1 Relationship of SIV Extension to VoiceXML**

The question of whether the SIV specification is written so that it can function both within VoiceXML-compliant deployments and independent of VoiceXML as a standalone language is an implementation issue that is not addressed here. If it is determined that the SIV specification can be written to support both VoiceXML and standalone SIV deployments the binding of SIV to the external applications shall be accomplished through a pair of tags: an SIV-VoiceXML tag and an SIV-non-VoiceXML tag. These two SIV variants are represented in figures 3.2a and 3.2b

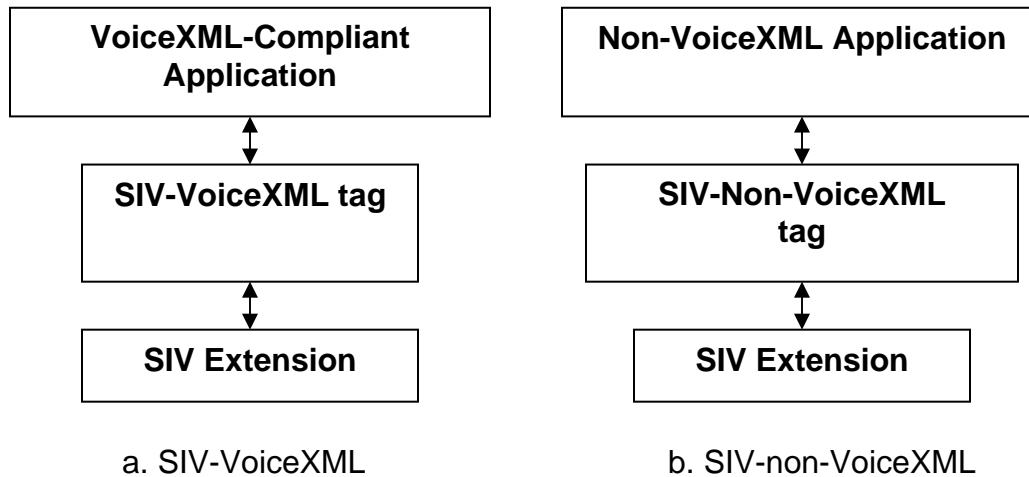


Figure 3.2 Tags for SIV as part of VoiceXML and standalone

### 3.2 MRCP v1 and v2

MRCP v1 and v2 are IETF protocols that control the operation of individual media processing resources, like ASR, TTS and recorders. MRCP v2 includes SIV. The use cases and requirements in this document assume that the underlying platform may conform to MRCPv2. The following types of SIV-related functions are therefore required:

**SIV sessions:** The notion of SIV sessions spanning multiple interaction turns where the basic SIV function and the claimed identity for verification are unchanged.

**Buffering:** The ability to retain information that may be needed by later SIV processing without knowing ahead of time whether authentication or enrollment is required.

**Rollback:** The ability to undo the last processed SIV turn.

**Simultaneous audio:** The ability to perform two or more different operations (e.g., Speaker Verification and ASR) on the same audio in the same interaction turn.

The SIV extension may be used with a VoiceXML browser that is connected to a platform that employs the ASR, TTS, recording, and SIV resources. If that platform is an MRCPv2 platform, all the SIV-related functionality that MRCPv2 exposes should be supported through the VoiceXML authoring interface. (see figure 3.2)

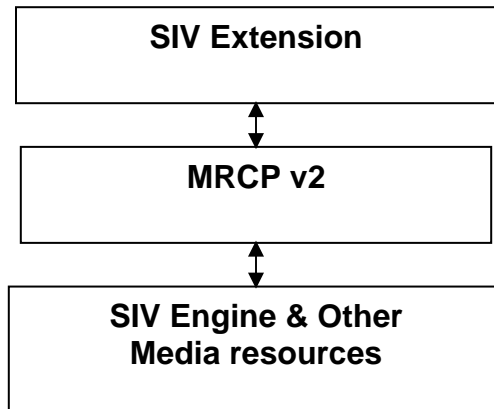


Figure 3.3 Relationship of SIV Extension to MRCP

### 3.3 Biometric Standards

SIV must be able to be used in conjunction with other biometric standards so that it may be used in multi-biometric environments. Many of these environments employ existing biometric standards developed by other standards bodies at the informal, national, and international levels. The core biometric standards are

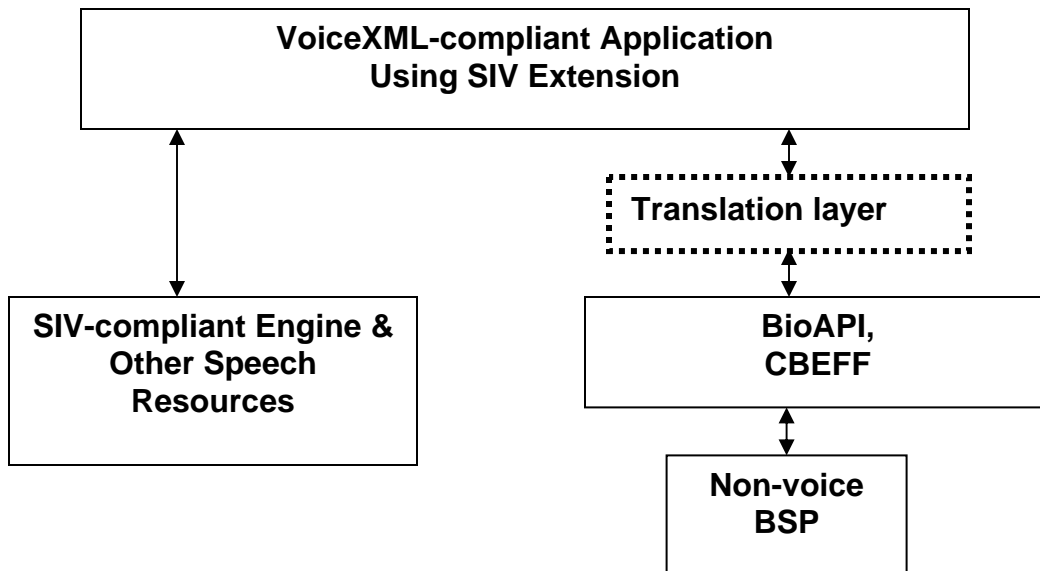
**BioAPI Specification** This standard defines a standard interface between a software application and underlying (generic) biometric technologies. There are two versions of this standard: the US version (*ANSI INCITS 358-2002*) and the international version (*ISO/IEC 19784-1*). The biometric data structure output from BioAPI is CBEFF-compliant.

**Data Exchange Format for Speaker Recognition** Two projects are working on this subject. One standard is being developed by the VoiceXML Forum and ANSI's biometrics committee INCITS M1. It creates an XML structure for interchange of raw data for SIV and it recognizes CBEFF. A similar project was approved by ISO in January, 2007. It aims to generate speech data interchange format(s) for SIV. One data interchange format will support raw speech; other possibilities could include formats for interchange at the feature vector level. The draft document from M1 has been contributed to the ISO project

**Common Biometric Exchange File Format (CBEFF).** This defines a standard method of packaging biometric data, to include standard header data elements. There are two versions: the US version (*ANSI/INCITS 398-2005*) and the international version (*ISO JTC001-SC37-N-1181*). Compliance with CBEFF no longer requires that an application transmit the entire CBEFF header with the data but it does require that the header

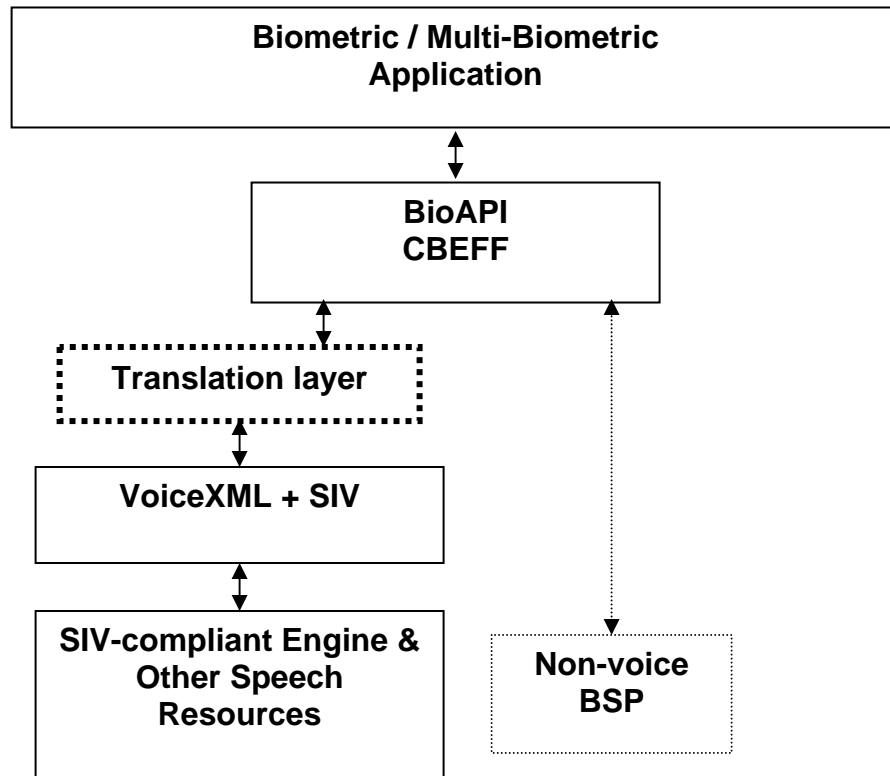
sent with the data be CBEFF compliant. For SIV, compliance means using the header in the *Data Exchange Format for Speaker Recognition*.

In a multi-biometric environment, an application may utilize a single, common interface to interact with all biometric technologies, including voice. This is especially true when biometric fusion is used. In this case, it may be useful for a compatibility layer to exist between BioAPI and SIV. In this way, the voice technology would act as a BioAPI Biometric Service Provider (BSP). Figure 3.3 represents the relationship between VoiceXML + SIV with BioAPI and the associated biometric standards in a VoiceXML application.



**Figure 3.4 Relationship of SIV and generic biometric standards in a VoiceXML application**

Figure 3.4 shows how VoiceXML with the SIV extension would figure in a biometric or multi-biometric application that complies with BioAPI.



**Figure 3.5 Relationship of SIV Extension to Biometric Standards in a BioAPI Application**

Compatibility with BioAPI has other practical purposes. Many US government programs incorporating biometrics require conformance to the above standards and compatibility will allow voice-based technologies to compete in these environments.

BioAPI does not directly access distributed architectures. However, there is a BioAPI Internetworking Protocol which will interface BioAPI components on different platforms.

BioAPI supports the application control of the UI but new amendments introduced at the international level by ISO need to be evaluated.

Two other ANSI projects of interest are Biometric Identity Assurance Services (BIAS) and the Study Report on Biometrics in E-Authentication. BIAS is developing an open framework for deploying and invoking identity assurance capabilities for services-based frameworks such as Web-Services and is being written in XML. It plans to support

- Service-Oriented Interfaces to identity assurance mechanisms,
- Biometric and Biographic data management,

- Remote binding and invocation services,
- Platform independent standard, with Web Services as the initial (primary) binding,
- Multi-biometric support (e.g., voice, finger, face, iris),
- Integrated with tokens for identity assertion, and
- Integration mechanisms with other standards (e.g. BioAPI).

The Study Report on Biometrics in E-Authentication examines the strengths and vulnerabilities of biometrics, including SIV, and could, therefore, offer valuable support to both developers of the SIV extension to VoiceXML and developers of SIV deployments. It is currently in draft form.

Although SIV is not intended to be compatible with ANSI X9.84-2003, *Biometric Information Management and Security for the Financial Services Industry*, this standard provides guidelines for the secure implementation of biometrics within a transaction-oriented environment and is, therefore, highly useful as a companion to SIV. An international version of this standard is ISO 19092.

## 4 Requirements

First of all we note that whatever the SIV extension to VoiceXML will look like in detail, all SIV features should cleanly integrate with the VoiceXML 3.0 declarative model and should make it easy to program applications using SIV.

We consider that all the functions we present in the requirements document MUST be supported by the VoiceXML language. However, since SIV technologies vary greatly, it seems obvious that not all VoiceXML implementations will support (nor will all applications need) all SIV types in every configuration. Thus, the V3 language must enable all SIV functions, but individual platform configurations may not include all these functions.

### 4.1 Basic Functions and Properties

#### 4.1.1 Basic Functions

SIV must support the following functions:

- Enrollment
- Verification
- Identification
- Supervised Adaptation

We recommend that the following data base functions be supported as well since they support the ability of the SIV technology to maintain the voice model database in the background:

- Copy Voice Model
- Is Enrolled
- Delete

SIV specific auditing functions should at least include logging of SIV results.

#### 4.1.2 Basic Properties of Voice Models

Voice models should have the following basic attributes

- Creation Date
- Modified/Adapted Date
- unique ID
- Security to detect/prevent tampering, binding methods, etc.
- Version (for tracking technology dependency)
- Adaptable (Y/N)
- Minimal verification score for adaptation (since this can change over time) [optional]
- Model type (text dependent, text independent, text prompted)
- Required utterance if text dependent [optional]
- Application level attributes (e.g. name/value pairs – like device type, language, etc.)

#### 4.1.3 Generic Speaker Verification Properties

Note: A complete data and event model for SIV engines is under development by the Speaker Biometrics Committee. This section is provided as preliminary information, and may eventually be published as a separate document.

SIV generic properties need to be defined, similar to the generic recognition properties.

Value	Type	Def.	Min. value	Max. value	Description
voice-model-identifier	string or list of strings	empty string			<ul style="list-style-type: none"> <li>• verification or supervised-adaptation: claimed identity of the caller, must point to existing voice-models</li> <li>• for verification (group authentication) it can be a list</li> <li>• for identification it must be a list or empty</li> <li>• for enrollment it must be a non-empty string, need not be known until commitment of the newly created</li> </ul>

					<p>voice-model</p> <ul style="list-style-type: none"> <li>for adaptation it must point to an existing voice-model</li> </ul>
group-identifier	string or list of strings	empty string			<ul style="list-style-type: none"> <li>can be used instead of lists of voice-model-identifiers, if the SIV-engine maintains information about groups for group-authentication or identification</li> <li>for enrollment: one or more groups where the newly created voice-model belongs to</li> </ul>
decision threshold	integer in the interval [0, 100].	empty			<p>For verification: determines the minimum verification score for which the speaker is accepted. If set then return results contain a decision in addition to the score. (See section 4.5).</p> <p>If decision-threshold is not set then the application has to make the decision based on the scores returned by the SIV engine.</p> <p>For identification: If this property is set, "open set identification" is performed. The speaker is rejected, if his score is below the threshold.</p>
adaptation-threshold	integer in the interval [0, 100].	empty			<p>For verification: determines the minimum verification score for which the speaker-model is adapted. (Use-case 5.4.1)</p> <p>If adaptation-threshold is not set then the application can still use the supervised-adaptation function on buffered data. (Use-case 5.4.2)</p>
maxnbest-speakers	integer	1	1		<p>For identification. The maximum size of the nbest-array of identified speakers. If "decision threshold" is specified as well, the actual number of speakers returned is the minimum of the number of "accepted" speakers and the number specified in maxnbest-speakers</p>
required-phrase	string	empty string			<p>Specifies the phrase that has to be spoken by the caller. If the utterance is not equal to the required-phrase then the utterance will not be used for verification. This only works with SIV engines that have an ASR inside (colocated resources).</p> <p>ToDo: decide whether NL-Results can be used and only top choice (N-Best) will be used.</p>
buffer-utterances	bool	false			<p>is used to indicate that all following utterances are buffered for possible use in later Speaker Verification</p>
use_buffered_utterances	bool	false			<p>Flag indicating whether buffered utterances should be used for verification.</p>
input-waveform-uri	uri	empty			<p>used for processing utterances from file</p>

abort-model					indicates the desired behavior of the verification resource upon session termination. If "true", any pending changes to a voice model due enrollment or verification adaptation are discarded. If "false", the pending changes for an enrollment session or a successful verification session are committed to the voice model repository.
no-input-timeout	integer				sets the length of time from the start of the verification timers until the declaration of a no-input event Comment (Martin): Only when using standalone resources. If using colocated resources ASR timeouts are used.
speech-complete-timeout	integer				same as Speech Recognizer header Speech Complete Timeout Comment (Martin): Only when using stand alone resources. If using colocated resources ASR timeouts are used.

Table 4.1 Generic SIV Properties

Value	Type	Def.	Min. value	Max. value	Description
num-min-verification-utterances	integer	1	1	?	minimum number of valid utterances before a decision is given for verification. Will be ignored if decision-threshold is not defined.
num-max-verification-utterances	integer	3	1	?	number of valid utterances required before a decision is forced for verification. The verification resource MUST NOT return a decision of 'undecided' once Num-Max-Verification-Utterances have been collected and used to determine a verification score. Fixed length verification is performed when num-max-verification-utterances = num-min-verification-utterances. Will be ignored if decision-threshold is not defined.
num-min-verification-frames	integer	1	1	?	minimum number of valid frames before a decision is given for verification. Will be ignored if decision-threshold is not defined.
num-max-verification-frames	integer	3	1	?	number of valid frames required before a decision is forced for verification. The verification resource MUST NOT return a decision of 'undecided' once num-max-verification-frames have been collected and used to determine a verification score. Fixed length verification is performed when num-max-verification-frames

					= num-min-verification-frames. Will be ignored if decision-threshold is not defined.
initial-negative-threshold	integer in the interval [0, 100].	empty			Specifies the lower border of the gray area (see section 4.6.6). If the verification score is lower then the initial negative threshold then the decision is rejected. It is necessary to set also initial-positivethreshold. Will be ignored if decision-threshold is not defined. Will not be used in case of fixed length verification.
initial-positivethreshold	integer in the interval [0, 100].	empty			If set then variable length verification is enabled. Defines the initial positive threshold. If the verification score is higher then the initial positive threshold then the decision is accepted and the verification session is finished, otherwise unsure and the verification session is still alive. It is necessary to initial-negative-threshold. Will be ignored if decision-threshold is not defined. Will not be used in case of fixed length verification.
negative-threshold	integer in the interval [0, 100].	empty			Specifies the lower border of the gray area for an individual turn. (See section 4.2.3)
positive-threshold	integer in the interval [0, 100].	empty			Specifies the lower border of the gray area for an individual turn. (See section 4.2.3)

Table 4.2 Decision Making Properties for Multiple Turn Verification

## 4.2 SIV Sessions

An SIV function such as verification or enrollment usually continues over several utterances until enough speech data is accumulated to reach a verification decision or to commit the voice model.

A sequence of interaction turns, where SIV results are continuously accumulated or a voice model is continuously build up or adapted is called an SIV session.

### 4.2.1 Phases of an SIV Session

An SIV session has three phases:

**Designation** Specifying the claim(s) that need to be used and the required type of processing (for example enrollment, verification). The designation phase should allow preprocessing and buffering utterances for later use. (See use case 5.7 for an example of buffering the caller's identity claim.) Some results might be returned from preprocessing even

before an identity claim is established, e.g. the speakers age, gender or input device.

**Audio Processing** This is the heart of the session. In the *audio processing* phase, processing is done on the session parameters that are specified in the *designation* phase (e.g. continuing to score the reference model-id “12345” on incoming audio) and is done in a “loop” over interaction turns that may last for one or multiple turns.

At the end of each audio processing turn an intermediate result (i.e. the result of matching) is returned by the engine. This intermediate result may be used to make a decision about what to do next. The application has access to both the incremental and the cumulative results that are generated following each turn (section 4.6).

**Cleanup** At the end of an SIV-session any intermediate data stored in memory is purged.

#### 4.2.2 Start and End of an SIV Session

The programming model needs to include directives to mark the start and end of an SIV-session.

An SIV-session can extend over more than one VoiceXML-document<sup>1</sup>. SIV-sessions can even be "open ended". (See for example Section 4.3.2 in [SIV-Apps] for an SIV-session with “application scope”.) An SIV-session ends at the latest when the containing VoiceXML-session ends.

Database operations as part of an SIV-session (e.g. storing a voice model) can take place even after hangup.

A verification session does not end automatically when a decision is returned by the engine (see e.g. use case 5.10).

The application may declare a transition or an event associated with a specific decision.

In case of an enrollment or adaptation session, the application must have the option to abort the session without creating or modifying the voice model.

#### 4.2.3 Decision Making Algorithms

There should be an option to specify for individual turns the thresholds for authentication and adaptation.

In order to achieve the optimal combination of dialog length versus accuracy, the application should be able to decide dynamically about the end of the session.

Therefore the outcome of a verification operation can not only be “accepted” or “rejected”, but also “inconclusive” (“more data needed”). The decision about the outcome can either be made by the application based on numerical scores returned by the SIV engine or it can be made by the SIV engine using integrated decision making algorithms configured through properties set by the application.

---

<sup>1</sup> When used in reference to the World Wide Web, a *document* is any file containing text, media or hyperlinks that can be transferred from an HTTP server to a client program.

Table 4.2 contains such properties for the commonly used decision making algorithms described below.

### Examples for making decision by application

Properties in this case:

decision threshold= NULL

num-min-verification-utterances -> will be ignored

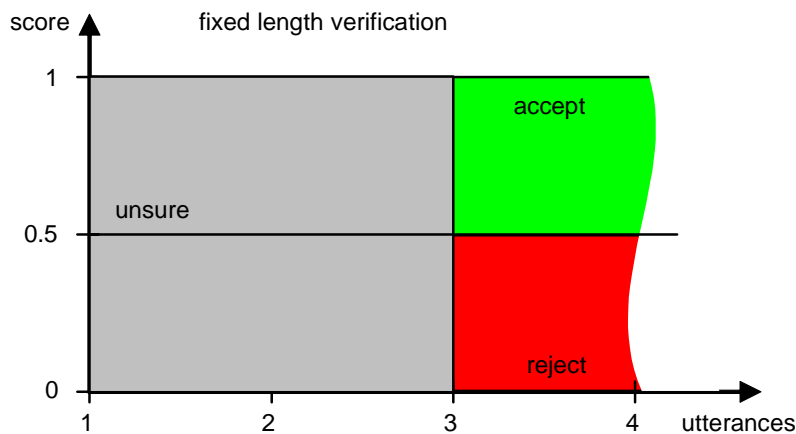
num-max-verification-utterances-> will be ignored

initial-negative- threshold-> will be ignored

initial-positive- threshold-> will be ignored

### Examples for making decision by SIV engine

#### 1. Fixed Length Verification:



Properties for above example:

decision threshold=0.5

num-min-verification-utterances=3

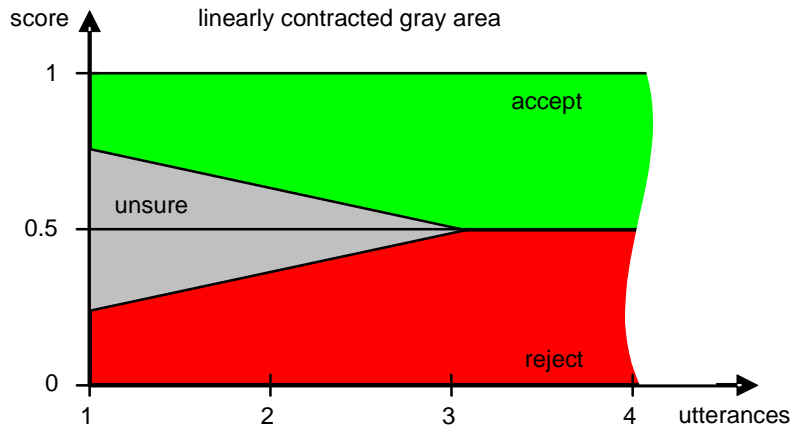
num-max-verification-utterances=3

initial-negative-threshold=NULL

initial-positive-threshold=NULL

#### 2. Linearly Contracted Gray Area:

The gray area defined by the initial negative and positive thresholds is contracted linearly with the number of utterances/frames until it is reduced to a point equal to the decision threshold, when the maximum number of verification utterances/frames is reached.



Properties for above example:  
 decision threshold=0.5  
 num-min-verification-utterances=1  
 num-max-verification-utterances=3  
 initial-negative-threshold=0.25  
 initial-positive-threshold=0.75

### 3. Gray Area Specified For Individual Turns:

The gray area is specified by the application for each turn independently. A decision is forced when the gray area is reduced to a point. This approach is more flexible than the one from the previous example. The following settings would lead to the same behavior:

negative-threshold[1]=0.25  
 negative-threshold[2]=0.375  
 negative-threshold[3]=0.5  
 positive-threshold[1]=0.75  
 positive-threshold[2]=0.625  
 positive-threshold[3]=0.5

### 4.3 Interaction Turns / Concurrent SIV/ASR Processing

An application developer must be able to specify at the individual turn level that one or more of the following types of processing need to be performed concurrently:

- ASR
- Audio recording
- Bridged transfer
- Buffering (SIV)
- Verification (SIV)
- Identification (SIV)
- Enrollment (SIV)

- Adaptation (SIV)

Concurrent processing of other forms of audio processing (e.g., channel detection, speaker classification) should also be permitted but remain optional.

Specifically it is required that any of the basic types of SIV processing can take place concurrently with ASR, recording and bridged transfer but also standalone. Buffering (SIV-preprocessing) can take place concurrently with ASR, recording and bridged transfer as well as with all basic types of SIV processing. Buffering can not be done standalone.

These requirements are independent of the specific way in which the platform may choose to do the concurrent processing. It's desirable to be able to stream in the audio five ways (in the most general case) but a platform may choose to record and then process the recorded utterance with some latency.

Also note that there is currently no explicit support of simultaneous, real time processing in BioAPI. Such operations would need to be done “under the covers” from BioAPI’s perspective. This is likely to change as multi-modal and multi-biometric deployments become more widely deployed.

#### **4.3.1 Input Items**

In VoiceXML, interaction turns translate into input items. The current input items defined in VoiceXML 2.1 are: <field>, <subdialog>, <object>, <record>, and <transfer>. The shorthand construct <menu> is technically not a form input item, but it does accept input from the user.

SIV operations are usually performed on <field> and <record>. The <transfer> item should allow background SIV-processing for the bridged transfer mode. The <subdialog> and <object> items can also have SIV performed on them. These elements may also contain SIV-sessions internally, which would then be concurrent SIV-sessions.

Furthermore the programming model should allow SIV-processing on buffered data, recorded data and on live speech not concurrently processed by ASR, record or bridged transfer. For example this could be realized by a <verify>-element or by allowing a <field> without a grammar.

#### **4.3.2 Completion Events**

Additional events should be added to the specification and shorthand notation for SIV event handlers should be defined. For example, there should be separate completion events. MRCP v2 includes RECOGNITION-COMPLETE and VERIFICATION-COMPLETE which are separate events and could be used in the SIV extension. This is important because the SIV and the ASR may not be synchronized to end simultaneously.

The specification should include separate *filled* and *noinput* handlers for SIV and ASR. They should have different names so that they can be distinguished from one another. This separation is especially important for systems in which the ASR and SIV are separate (stand alone) engines because they may return different results. For example, if the caller provides a minimal response the ASR may report that there was no input while a text-independent SIV might find the input very useful for its purposes.

Some events are vendor specific which means that the specification should be extensible with regard to events.

Note: Usually an interaction turn with concurrent processing of several resources ends only when all involved resources have sent a completion event. Then the respective event-handlers should be executed in an order prescribed by the application and not in the order the completion events for the utterance were received.

SIV processing which runs concurrently with bridged transfer behaves differently: There it can be useful to process intermediate results from the SIV engine while the bridged transfer is still going on. For example the verification-result might be displayed to the operator while he is talking to a customer.

#### **4.4 Control of Audio Processing**

A speaker verification or identification decision is made using one or more turns in the *audio processing* phase. SIV-processing accuracy increases with the amount of available audio. This does not mean that all audio should always be available to the SIV system.

The audio that can be used for SIV processing should be controlled by the application programmer within the boundaries of the SIV technology and the nature of the implementation.

##### **4.4.1 Pause and Resume**

SIV-sessions can be paused for some turns and resumed on another turn. Buffering may be turned on or off for individual turns independently.

##### **4.4.2 Rollback**

SIV operating in text-dependent or text-prompted mode should be configured to accept only the audio that is provided in response to the SIV request. If that SIV is used with standalone or co-located ASR it may be configured to use ASR to verify that the speaker provided the expected input.

There should be a mechanism to “undo” or roll back the processing performed on the last turn in the SIV session. One of the most common uses for rollback is to ignore the last input based on ASR results. (see use cases 5.10) This is especially important for text-independent and text-prompted SIV systems. Those systems do not restrict the verification/identification input to a single required phrase.

MRCP v2 supports rollback for the last utterance. To maintain high level correspondence with MRCP v2 it's recommended to limit the number of rollbacks to one utterance. Another reason for this limitation is that currently no SIV engine seems to support the rollback of more than one utterance.

#### **4.4.3 Misrecognized Audio**

SIV operating in text-independent mode may be configured to use any available audio for SIV processing, including misrecognized audio and retries.

#### **4.4.4 Enrolled Phrases**

For text-dependent SIV with no concurrent ASR, a voice-model can contain several different fixed phrases from one speaker. The application must be able to specify the phrase associated with each interaction turn both for enrollment and for verification or identification. See use case 5.1.5 for an example. This mechanism should not be restricted to phrases that agree with enrolled phrases, but should allow recombination of enrolled phrases. For example, if a speaker has enrolled the numbers "21" and "32", an engine might be able to score also the numbers "22" and "31", if it is told by the application to expect those numbers.

### **4.5 Return Results**

The runtime results that are returned after each turn within an SIV session should be made available to the application developer (shadow variables seems to be a viable way of returning results). The structure of the result should include result elements such as:

- isValid: if the utterance was used for the session
- validityReason: reason why the utterance was invalid (e.g. wrong password, playback indication, bad data.)
- cumulative and incremental scores
- decision : "accept", "reject" or "inconclusive"
- decision reason (e.g. wrong password, playback indication for "reject", insufficient data or bad data for "inconclusive".)
- duration
- identified speaker for identification (n-best list)

In addition, a mechanism should exist to return proprietary information from the platform (e.g. ECMA Script variables) such as:

- detected gender
- handset detection,
- channel detection, etc.
- flag: more data needed,

- request for specific data: if “decision” is “inconclusive” (more data needed) this may specify words or phrases that have a good phonemic content to allow the engine to reach a decision

This will enable the application to handle a case where more than one speaker received high score or none of the speakers received a high score.

This information can be in a single record, such as a CBEFF for voice.

Some SIV-engines generate suggestions on the words or phrases to prompt for, to get the optimal combination of phonemes from the speaker. If such a “request for specific data” is generated before any speech data has been processed, it takes the form of a vendor specific voice model property (see section 4.1.2).

#### **4.7 Concurrent SIV Sessions**

The specification should allow concurrent SIV-sessions.

We define “concurrent” sessions as two or more SIV sessions running in parallel and producing their results independently. For example, an application might initiate two independent verification sessions performed by two separate engines operating on the same voice model or it might begin SIV sessions of different types (e.g., enrollment and verification). (See use cases 5.8.1 and 5.8.2).

#### **4.8 Nested SIV Sessions**

The specification should allow nested SIV-sessions.

An SIV-session is called “nested” if it is initiated by another session’s engine which uses the results of the nested session to produce it’s own results. The main example for a nested session is that of an open set identification session which relies on a nested verification session to back up it’s result (use case 5.9). Since the application declares only the containing session directly through VoiceXML, one or more specific session parameters are needed to configure a nested call. Through these parameters an application might, for example, direct the engine to call preferred expert engines for handling parts of the utterance. For example, the application might instruct a verification engine to call another engine that is an expert on digits whenever digits appear in the utterance. In the case of open set identification (use case 5.9) the application can specify through such specific parameters which verification engine is used for backing up the identification result.

As the examples suggest, usage of nested sessions by an application implies first of all that the main SIV engine is capable of initiating nested sessions and using their results. Therefore this feature will not be supported by all platforms, in which case a platform should generate a suitable failure message.

Nested session calls must be used with extreme care to avoid infinite loops. An infinite loop might occur when the main SIV engine calls another engine that calls the main SIV engine.

#### 4.9 Multiple Voice Models per Speaker

The specification should include the option to create more than one voice model per speaker. This option will be particularly used by application with enhanced security and accuracy. For example, this feature will provide the ability to have one voice model for land line and another for cell phone for highly accurate verification.

When performing simultaneous adaptation with verification or identification, it should be possible to adapt any or all reference models for a speaker.

#### 4.10 Multi-Factor Systems and Applications

The SIV specification must support multi-factor systems and applications. The concept of having multiple security factors is well established. Security factors are described as

Knowledge factor/Something you know (e.g., password, PIN, mother's maiden name)

Possession factor/Something you have (e.g., card, token, key)

Something you are/Biometric factor (e.g., SIV, hand geometry)

The commercialization of global positional technology has led to the addition of a Location factor ("where you are") to this list. According to Speaker Identification and Verification (SIV) Glossary (see Section 6.1 Related Documents) multi-factor authentication is defined as

the combination of two or more authentication techniques that together form a stronger or more reliable level of authentication. This usually involves combining two or more of the following types:

- Knowledge factor, "something an individual knows"
- Possession factor, "something an individual has"
- Biometric factor, "something an individual is"
- Location factor, "where you are"

The definition refers to multi-factor authentication as "usually" combining more than one of these as is shown by the sample combinations in Table 4.2

	<b>Knowledge</b>	<b>Possession</b>	<b>Biometric</b>	<b>Location</b>
<b>Knowledge</b>		Password + Token (e.g., RSA SecureID)	Password + text- dependent SIV	PIN + wireless location
<b>Possession</b>			Breathalyzer + SIV (criminal offenders)	Electronic bracelet + GPS
<b>Biometric</b>				Outbound calling + SIV
<b>Location</b>				

Table 4.2 Traditional Multi-factor Authentication

Many of the examples in Table 4.2 are well-established multi-factor deployments. Text-dependent SIV, for example, is inherently two-factor because it requires the user to say a pre-determined password/passphrase. Wireless carriers have profiling technology that examines the geographical location of two sequential calls placed from the same phone. If it is physically impossible for the caller to move from location 1 (e.g., New York) to location 2 (e.g., San Francisco) in the time that elapsed between the calls. An alcohol breathalyzer with SIV has been in use for monitoring home-incarcerated offenders convicted of DUI (driving under the influence of alcohol) since 1989. The combination of GPS with electronic bracelets for criminal offenders is more recent and allows monitoring of offenders beyond a single location. Similarly, SIV is combined with outbound calls to ensure that community-released are where they are scheduled to be.

Some of these applications have been extended to triple-factor authentication. Use case 5.12.2 contains a security session on a user's screen that is combined with an outbound call to that person. The user must be at the specified telephone (location factor), provides a (usually) text dependent voice sample (knowledge and biometric factors), plus she/he is asked to read the dynamically-generated alphanumeric pattern displayed on the screen (knowledge factor).

In the past ten years, the concept of multi-factor has been expanded to include systems and deployments that employ multiple techniques that fall into the same security factor. Traditional examples of this can be seen in the request for multiple bits of knowledge by call centers in banks.

There are also a growing number of multi-biometric systems that are considered to be multi-factor systems. These include products and deployments that combine SIV with lip movement patterns and facial recognition. Recently SIV + SIV systems have also been included in the multi-factor scope when the SIV engines have been developed independently.

Deployments that combine sensory modalities (vision + sound) must also be supported (see Use case 5.12.2). The combination of SIV with lip movement and facial recognition is another form of multi-modal authentication that currently exists.

These examples clearly demonstrate that multi-modal cases exist today. They are also part of a growing trend that needs to be built into the flexibility of the specification.

## 5 Use Cases

The following are use cases in a conversational telephony system where SIV technology is utilized. The use cases aim at covering the space of possible interaction scenarios with a system authored in VXML using the SIV extensions

## 5.1 Enrollment

### 5.1.1 Successful Initial Enrollment (Text Independent)

basic function: (initial) enrollment [4.1.1]

text independent mode

standalone SIV with concurrent ASR [4.2]

control of audio (rejected utterance allowed) [4.4]

result property: duration [4.6]

In this example a series of questions is presented to the user. Every time the user answers, the audio is captured and sent for SIV processing resulting in the creation of a reference model in memory. The voice model is committed at the end of the interaction. This example involves a text independent engine, meaning that speaker authentication may be performed regardless of the spoken content (including words not spoken before).

C: Welcome, please say your account number.

H: 12345678

*<Speech is recognized, and a blank voice model is created in memory, labeled 12345678>*

C: What is your date of birth?

H: December 25<sup>th</sup> 1969

*<speech may be recognized, the voice model is updated with the initial turn, and an initial voice model is now available in memory>*

C: In which city and state is your branch located?

H: Excuse me?

*< speech may be recognized, the voice model in memory is updated again>*

C: I need to know the location of your banking branch. Please say the city and state

H: Oh I see, it's in Yonkers, New York

*< speech may be recognized, the voice model in memory is updated again>*

C: What is you mother's maiden name?

H: Smithers

*< speech may be recognized, the voice model in memory is updated again>*

C: What is the name of your high school?

H: Degrassi High

< speech may be recognized. The application decides when the model is to be committed. This may take the form of controlling the number of utterances requested or it may be based on statistical measures, such as the duration of the collected audio. Based on those statistics the engine may recommend to the application that the model be committed but the application makes the final decision.>

C: Thank you. You are now enrolled in the system. Goodbye.

### 5.1.2 Unsuccessful Initial Enrollment (Text Independent)

basic function: (initial) enrollment [4.1.1]

text independent mode

standalone SIV with concurrent ASR [4.2]

control of audio (rejected utterance allowed) [4.4]

aborted enrollment [4.3.1]

This example is the same as the previous example, except that the enrollment process fails, and the model is not committed.

C: Welcome, please say your account number.

H: 12345678

< speech is be recognized, and a blank voice model labeled 12345678 is created in memory>

C: What is your date of birth?

H: December 25<sup>th</sup> 1969

< speech may be recognized, the voice model is updated with the initial turn, and an initial voice model is now available in memory>

C: In which city and state is your branch located?

H: Excuse me?

< speech may be recognized, the voice model in memory is updated again>

C: I need to know the location if your banking branch. Please say the city and state

H: What?

<recognition fails, the voice model in memory is updated again>

C: Please say the name of the city where your banking branch is located.

H: ...

<Recognition fails. No speech is detected and the voice model in memory stays the same>

C: I am sorry, we need to verify all of your personal information to complete this enrollment process. Please call again when you have the information available. Good bye.

<The voice model in memory is deleted and not committed. The application hangs up>

### 5.1.3 Initial Enrollment (Text Dependent) on the Identity

basic function: (initial) enrollment [4.1.1]  
text-dependent mode  
standalone SIV with concurrent ASR [4.2]  
cumulative scores [4.6]

Unlike text independent technology, text-dependent (and text constrained) speaker authentication typically requires a repetition of a pass-phrase for enrollment. The steps of updating a voice model in memory and committing on successful enrollment is followed in the same way described previously.

C: Hello, please say your account number

H: 12345678

*< speech is recognized using ASR. The system goes through the sequences of tests needed to ensure that the speaker is the person whose ID is 12345678 and, if that's successful, a blank voice model labeled 12345678 is created in memory >*

C: Please repeat you account number

H: 12345678

*< speech may be recognized, the voice model is updated with the initial turn, and an initial voice model is now available in memory >*

C: Please repeat it again

H: 12345678

*< speech may be recognized, the voice model in memory is updated again >*

C: repeat again

H: 12345678

*< speech may be recognized, based on an indication coming from the engine, it is decided that the voice model may be committed. >*

C: Your voice model has been created. Thank you.

### 5.1.4 Initial Enrollment (Text Dependent) Single Passphrase

basic function: (initial) enrollment [4.1.1]  
text-dependent mode  
standalone SIV without ASR [4.2]

The enrollment may be performed in the security department or customer service department of the company/organization. The enrollee has been approved to move to this step

C: Hello, this is the voice enrollment portion of your registration. Please enter your dynamic registration number using the touchtone keys of your telephone.

H: enters 12345678

*< a blank voice model labeled with the ID or account number associated with that dynamic registration number is created in memory >*

C: Thank you. Please think of a password that is at least two-seconds long and press the # key when you are ready to enroll

H: presses #

C: Please say your password after the tone <TONE>

H: This is my password

C: Please repeat your password <TONE>

H: This is my password

C: Please repeat your password one more time <TONE>

H: This is my password

C: Your voice model has been created. Thank you.

*< the voice model may be committed. >*

### **5.1.5 Initial Enrollment (Text Dependent) Multiple Phrases**

basic function: (initial) enrollment [4.1.1]

text-dependent mode

standalone SIV without ASR [4.2]

*<begins with the same sequence as for 5.1.4>*

C: Please say the city or town in which you were born?

H: January

*< a voice model extension is created for this question. >*

C: What is your favorite color?

H: blue

*< a voice model extension is created for this question. >*

C: What is your mother's maiden name?

H: Smith

*< a voice model extension is created for this question.>*

C: You have successfully completed the enrollment process. Thank you.

### **5.1.6 Initial Enrollment (Text Prompted/Challenge-Response)**

basic function: (initial) enrollment [4.1.1]

text-prompted mode

standalone SIV without ASR [4.2]

This approach is used for higher security applications and for offender monitoring/tracking. The enrollment is performed in the corrections facility and in the presence of a corrections official. These systems may or may not include ASR to do internal checking that the person correctly repeated the sequences.

*<Official manually enters the offender's code or otherwise initializes the enrollment process which initiates a blank voice model>*

C: say 12345

H: 12345

C: say 93845

H: 93845

*< continues asking for numbers. Most systems have a specified number of sequences they enroll, for example 12. The sequences may be words or phrases, such as "Chicago, Illinois" The voice model may be committed >*

## **Enrollment for Group Authentication**

Group Authentication is suitable for situations where a single account includes multiple users like jointly-owned bank accounts and cell phones with more than one user. The enrollment procedure is likely to be similar to any of the other enrollment procedures except for the fact that it is attaching more than one voice model to an ID.

## **5.2 Verification**

### **5.2.1 Text-Independent Verification**

basic function: verification [4.1.1]

text-independent mode

Speaker verification may be used to verify a user's identity prior to allowing access to a transaction system or other secured system, site, container, etc.. The

following example shows such a verification use case for a user that is already enrolled. In this example the engine is text independent. The user in the example is rejected since the verification score does not meet the acceptance threshold.

C: Hello, what is your name?

H: Jonathan smith

*<speech is recognized, the voice model labeled "Jonathan smith" is retrieved>*

C: Tell me how you got to work this morning

H: I got stuck on the LIE for 45 minutes

*< speech may be recognized, the speech is scored against the voice model and scores very poorly resulting in a rejection decision>*

C: I am sorry, but I cannot allow you to perform any transactions today. Good bye.

### 5.2.2 Ongoing Verification

basic function: verification [4.1.1]

text-independent mode

ongoing background verification after successful authentication [4.3.1]

incremental scores [4.6]

Speaker verification may be performed concurrently with ASR. Since text independent speaker verification does not impose any constraints on the content that the user may say, authentication need not be done in an isolated pre-transaction stage. It may be performed in addition to pre-transaction authentication, or instead of it, *while* transactions are being conducted. In the following example an imposter attempts to break in to an account using a recording. She passes the pre-transaction user authentication, using a recording of the genuine user's voice. When she begins to perform transactions, ongoing verification detects her different voice and blocks transaction processing.

C: Hello, please say your account number:

H: 12345678

*<speech is recognized, and the voice model labeled 12345678 is retrieved>*

C: What is your mother's maiden name?

H: Smith (plays the recording)

*< speech may be recognized, the speech is scored against the model 12345678 resulting in an acceptance decision>*

C: What would you like to do today?

H: I would like to transfer all of my funds to account number 87654321

<speech is recognized and verified, the new score is low resulting in rejection>

C: Sorry, no can do. Please call again later.

### 5.2.3 Text-Prompted Verification

basic function: verification [4.1.1]

text-prompted mode

challenge response

The system prompts the user to say one or more randomly selected (from the enrolled items) or randomly generated (from non-enrolled items) digits, combination lock sequence (e.g., 58 34), words, or phrases. Most systems use ASR to verify that the person said the correct sequence.

C: Greetings. Please say your account number

H: 12345678

<voice model 12345678 is retrieved.>

C: please say 12345

H: 12345

C: Ok, now please say 93845

H: 93845

C: Thank you. You are verified...

### 5.2.4 Text-Dependent Verification

basic function: verification [4.1.1]

text-dependent mode

can be realized with standalone SIV without ASR [4.2]

C: please enter your ID using the touchtone keypad

H: <enters ID>

C: Please say your password

H: This is my password

<verification is performed and is successful>

C: Thank you.

<reference model may be updated>

## Other verification examples

See use cases 5.10.1 and 5.10.2.

### 5.3 Group Authentication

basic function: verification [4.1.1]

result contains an identity (not only a decision) [4.6]

*<A user calls the system and a list of possible users corresponding to the caller-id is retrieved. In this case the caller-id serves as the account number>*

C: Hello, please choose from the following options : upgrade service, pay bill, or tech support

H: Upgrade Service

*<identification takes place between the three reference models corresponding to the account: mom, dad, and Joe. The identification result determines that the speech is within the group and that Joe is the best-match reference model. Since Joe is underage he is not authorized to purchase upgrades>*

C: Joe, you cannot purchase upgrades, sorry. Is mom or dad home?

H: hangs up...

### Other Group Authentication Examples

The verification use case 5.2.3 could also be an example for group authentication where it is irrelevant who the identified person is but only that the person belongs to the group of allowed speakers.

The example above, in contrast, is one where the identity of the individual does matter, because it relates access rights of the speaker.

An example for open-set-identification not restricted to a small group of speakers, can be found in 5.9.

## 5.4 Adaptation

### 5.4.1 Simultaneous Verification and Reference Model Adaptation

basic function: supervised adaptation [4.1.1]

Instead of enrolling a reference model from scratch, it is possible to update an existing reference model from incoming speech. When adapting a user's reference model, there should be a high level of certainty that the speech originated from the true user. Otherwise, the voice model may become contaminated with the voice of a different user and will be prone to false-match errors. It is often convenient to use speech that is available from authentication to create a temporary adapted reference model. The decision to commit the updated model must, however, wait until verification is complete and the verification results are available. Then it may be determined whether all necessary requirements for committing the updated model have been satisfied

(e.g., the verification score meets or exceeds a designated threshold). This example shows simultaneous adaptation and verification in which a commitment requirement has not been met.

C: Hi, who are you?

H: I am Nick

*<speech is recognized, and the voice model labeled "nick" is retrieved>*

C: Hey Nick, what is your pass code?

*<speech may be recognized, the speech is verified against the original voice model, and a new adapted voice model is also created. The speech scores high enough to accept the user, but not high enough to update the model. The new adapted voice model is discarded>*

H: Cool. Let's listen to some new music.

## 5.4.2 Supervised Adaptation with Low Scoring Samples

Basic Function Supervised Adaptation [4.1.1]

decision thresholds (gray area) [4.1.3]

The caller dials a designated 800 phone number to access their account. The phone number of the caller's phone is captured via ANI. The system prompts the user for their account number. The account number is verified via speaker verification and the preregistered phone number matches one on file. However, the voice verification score falls within a *gray area*. The *gray area* indicates that the user has not identified positively since the score is lower than the threshold but falls within a range of scores which does not rule out the user as authentic. In this case a preregistered security question is used to make the decision. If the speaker is authenticated through the security question, the application adapts the voice model presumably resulting in a better verification score on the next call.

C: Greetings. Please say your account number

H: 9147620291

*<speech is recognized, the voice model labeled '9147620291' is retrieved, verification is performed against the voice model and scores 0.93 which does not meet the threshold of .94 required for a positive verification decision but is within the gray area specified by the engine vendor (between .90 and .94). Since the Caller Id is known to belong to the account number, a challenging question is posed.>*

C: Thanks. Now say your mother's maiden name.

H: Mary Smith

*<speech is recognized, the answer is correct, the speaker is accepted and the voice model is adapted.>*

C: Thank You

## **5.5 The Database Function IsEnrolled**

basic function: (initial) enrollment [4.1.1]

database function: isEnrolled [4.1.1]

If a user attempts to authenticate without first enrolling the system needs to determine that there is no reference model for that individual. This example uses text-dependent verification with ASR but the same general pattern can apply to any of the other enrollment use cases shown above.

C: Hello, please say your account number

H: 12345678

*<system does not find a reference model for that ID>*

C: A voice model for 12345678 has not yet been created. Would you like to create one now?

H: Yes

*<system goes through the sequence of tests need to ensure that the speaker is the person whose ID is 12345678. If that is successful, a blank voice model labeled 12345678 is created in memory>*

C: Please repeat your account number

H: 12345678

*< speech may be recognized, the voice model is updated with the initial turn, and an initial voice model is now available in memory>*

C: Please repeat it again

H: 12345678

*< speech may be recognized, the voice model in memory is updated again>*

C: repeat again

H: 12345678

*< speech may be recognized, based on an indication coming from the engine, it is decided that the voice model may be committed.>*

C: Your voice model has been created. Thank you.

## **5.6 SIV Processing on Recorded Speech**

verification on recorded utterance [4.3.2]

Authentication requires the specification of a single identity claim (speaker verification) or multiple identity claims (group authentication). Many use cases involve collecting a spoken identity claim from the user. The audio collected during the identity claim can be used for SIV purposes. However it cannot be used until the claim utterance is recognized. It is possible to address this problem by recording the speech from the claim utterance, and subsequently verifying the recorded utterance.

C: Please say your account number

H: 12345678

*<ASR and recording take place. The voice model "12345678" is fetched based on the recognized claim. The recorded speech is sent offline to the engine to be verified with identity claim "12345678", scores high and is accepted>*

C: Thank you, you are verified.

## **5.7 Buffering**

buffering in the designation phase [4.3.1]

The problem with this solution is that verifying the recorded speech offline means a peak of computation following the end of the turn (unlike the ongoing computation that typically takes place *while the audio is collected*). In general, it may be possible that some of the computational stages of speaker verification may be performed in the engine without knowing the identity claim and the type of SIV operation that will need to be performed in the future. Since in this example we know ahead of time that SIV processing will be needed it is beneficial to perform the common pre-processing that is always performed inside the engine during the course of the first and not wait for the recognition result. This mechanism is called "buffering". This example shows how an engine that supports buffering can perform a one-turn SIV session:

C: Please say your account number

H: 12345678

*<ASR and buffering take place. The voice model "12345678" is fetched based on the recognized claim. The engine is requested to verify from the internal buffer with identity claim "12345678". The returned score is high and the user is accepted>*

C: Thank you, you are verified

## **5.8 Concurrent SIV Processing**

### **5.8.1 SIV Sessions Tied to ASR Result Alternatives**

Concurrent SIV sessions [4.3.3]

In this example the caller is asked for his account number for verification. The account number is a digit sequence of variable length but usually will have around 12 digits. For example it might be a mobile phone number. Since misrecognitions may occur with longer digit sequences, combining N-Best-processing with SIV leads to better recognition results for the first utterance. If the verification result is "rejected" for the first ASR-result alternative, but the verification result is "accepted" for the second ASR-result alternative, that number becomes the primary candidate for number the caller said. In general the result alternatives get reordered by a combination of recognition and verification scores.

The caller might still be asked for confirmation of the number, but the success rate will be higher than without SIV-processing *before* the confirmation.

C: Greetings, please say your account number.

H: 0 1 6 9 2 2 7 3 5 8 0 9

<N-Best processing returns a list like this:

Result Alternative	ASR Confidence
0 1 6 9 2 2 7 3 5 8 0 9	0.95
9 1 6 9 2 2 7 3 5 8 0 9	0.94
0 1 6 0 2 2 7 3 5 8 0 9	0.83
9 1 6 0 2 2 7 3 5 8 0 9	0.80
0 1 6 9 3 2 7 3 5 8 0 9	0.75
9 1 6 9 3 2 7 3 5 8 0 9	0.75
0 1 6 0 3 2 7 3 5 8 0 9	0.74
9 1 6 0 3 2 7 3 5 8 0 9	0.74
0 1 6 9 2 3 7 3 5 8 0 9	0.73
9 1 6 9 2 3 7 3 5 8 0 9	0.73

...

Result alternatives that don't correspond to voice models are excluded. For the remaining alternatives, the utterance is verified against the voice model simultaneously resulting in a list like the following.

Result Alternative	ASR Confidence	Verification Score	Combined Score
0 1 6 9 2 2 7 3 5 8 0 9	0.95	-0.05	0.370
<del>9 1 6 9 2 2 7 3 5 8 0 9</del>	<del>0.94</del>		
0 1 6 0 2 2 7 3 5 8 0 9	0.83	-0.50	-0.590
<del>9 1 6 0 2 2 7 3 5 8 0 9</del>	<del>0.80</del>		
<del>0 1 6 9 3 2 7 3 5 8 0 9</del>	<del>0.75</del>		
9 1 6 9 3 2 7 3 5 8 0 9	0.75	0.75	0.187
0 1 6 0 3 2 7 3 5 8 0 9	0.74	-0.06	0.250
<del>9 1 6 0 3 2 7 3 5 8 0 9</del>	<del>0.74</del>		
0 1 6 9 2 3 7 3 5 8 0 9	0.73	0.10	0.560
9 1 6 9 2 3 7 3 5 8 0 9	0.73	-0.25	-0.140

The combined score is a weighted combination of recognition and verification scores (here  $\frac{1}{2} \cdot \text{ASR} + 2 \cdot \text{SIV}$ ). In this case the 9<sup>th</sup> result alternative "wins" with a combined score of 0.56.>

C: Your account number is 0 1 6 9 2 3 7 3 5 8 0 9. Is this correct?

## 5.8.2 Combining Different SIV-Engines in Concurrent Sessions

Concurrent SIV sessions [4.3.3]

Multiple Voice models per Speaker [4.9]

This example corresponds to application 4.1.3. "PIN-less Call-Center authentication" from [SIVAPPS]

In this application name-spotting (hot word recognition) associated with text-dependent verification (on enrolled-names) and text-independent identification are performed concurrently.

The text-independent identification can itself be divided into two concurrent sessions, a small group identification session on the voice models associated with the transmitted CLI (if provided), and a large group identification session on the rest of the enrolled clients. The small group identification session will usually come up with a result faster than the large group identification session. If the small group identification fails and the large group identification needs more data, the operator might ask for the callers name explicitly.

*<Client calls the bank not from her home phone.>*

Operator: Best-Bank Smith. Hello.

H: Hello, this is Sally Johnson. I would like to transfer some money from my account.

*<The small group SIV session returns "rejected" because the caller is not calling from her home account, The large group identification session does not have enough data yet. The name spotting session isolated the name "Sally Johnson" and the text-dependent verification session confirms that the speaker is the person she claims to be. All three voice models for the speaker will be adapted with data from this call.>*

Operator: Ok, Mrs. Johnson. How much do you want to transfer?

...

*<A few minutes later another call from the same client comes in. This time the caller forgets to say her name again.>:*

Other Operator: Best-Bank Williams. Hello.

H: Hello. I need to transfer some more money from my account.

*<This time the caller can be identified by the large group identification session alone, since the voice model was updated after the first call.>*

Other Operator: Ok, Mrs. Johnson ...

## 5.9 Nested SIV Sessions

Basic function: identification [4.1.1]

Nested session [4.3.4]

In this example Open-Set Identification is implemented through nested sessions. In this scenario, Joe Smith who is not in the database talks to the system.

C: Hello. How's the weather where you are?

H: Oh, it's sunny today!

*<The application submits the speech sample to the SIV engine (OSID-Engine) to perform an open-set identification of the speaker. During the identification process the OSID-Engine does a nested call to another SIV engine which specializes in closed-set identifications (CSID-Engine) and obtains the identity of the unknown speaker based on the best identification score against a particular reference model in the enrolled database. In this example the CSID-Engine returns "John Doe" as the identity of the speaker. The OSID-Engine takes this identity and makes another nested call to a Verification Engine (V-Engine) and passes the resulting identity "John Doe" and the speech pattern of "Joe Smith" and asks the V-Engine to do a verification of this individual. The V-Engine comes back with a rejection. The OSID-Engine takes the results and responds as follows.>*

C: Sorry, but I have never met you in the past and I am not allowed to give you access to my files. It was nice talking to you! Good bye!

## 5.10 Rollback

### 5.10.1 Text-Prompted Verification with Rollback

basic function: verification [4.1.1]

text-prompted mode

challenge response

rollback [4.6]

C: Greetings. Please say your account number

H: 12345678

*<voice model 12345678 is retrieved.>*

C: please say 12345

H: 12345

C: Ok, now please say 93872

H: 12345

*<The ASR recognizes that the person said the incorrect sequence and there is a rollback. NOTE: This could be an attack with recorded speech..>*

C: Sorry, I didn't hear the correct phrase. Please say 93872.

H: 93872.

C: OK, you are verified...

## 5.10.2 Text-Prompted Verification with Replay Attack

basic function: verification [4.1.1]

text-prompted mode

challenge response

rollback [4.6]

C: Greetings. Please say your account number

H: 12345678

*<voice model 12345678 is retrieved.>*

C: please say 12345

H: 12345

C: Ok, now please say 93872

H: 12345

*<The ASR recognizes that the person said the incorrect sequence and there is a rollback>*

C: Sorry, I didn't hear the correct phrase. Please say 93872.

H: 12345

C: Please hold while you are transferred to a security agent....

## 5.11 Return Results

Supervised Adaptation [4.1.1]

Return Result [4.6]

text-prompted mode

knowledge based challenge

co-located SIV and ASR [4.2]

automatic rollback through preset "prompted phrase" [4.4]

Runtime results are returned by the SIV Engine after each utterance. This use case illustrates the variety of results that can occur.

Example: With the user's agreement, the SIV Application places an outbound call to the pre-registered landline phone where the user resides. The Application interacts with the user to enroll a voice model. First, the user is asked to speak a temporary shared secret – a 10 digit confirmation code. The SIV engine returns this result as a series of name-value pairs after processing the utterance:

Field Name	Value	Notes
IsValid	1	Whether utterance was used to create voice

		model. 0 = invalid utterance; 1 = valid utterance
ValidityReason	""	Optional Field: Reason utterance was invalid. E.g. ASR did not match requirements
VerificationScore	0	Verification score of this utterance. This is the verification score against an existing voice model. In this example, since none exists, this score is 0
CumulativeVerificationScore	0	Cumulative verification score for this session. Same as above
Duration	2710	Duration of this utterance, In milliseconds
Decision	0	Optional Field: Tri-state value. 0 = inconclusive; 1 = accept; 2 = reject;
DecisionReason	""	Optional Field: Additional information about decision.
Gender	1	Optional Field: Tri-state value. 0 = inconclusive; 1 = Female/Youth; 2 = Male;
ChannelType	1	Optional Field: 0 = unknown; 1 = landline; 2 = cell; 3 = VoIP; more values?
LinguisticContent01	"8720361459"	Optional Field: ASR #1 most likely spoken content
LinguisticContent02	"8720361499"	Optional Field: ASR #2 most likely spoken content

The App analyzes the result and sees that it needs more utterances from the user. Now the user is prompted for an account number - 8374015374182934

Field Name	Value	Notes
IsValid	0	Whether utterance was used to create voice model. 0 = invalid utterance; 1 = valid utterance
ValidityReason	"MISMATCH"	Optional Field: Reason utterance was invalid. E.g.

		ASR did not match requirements
VerificationScore	60	Verification score of this utterance. This is the verification score against an existing voice model.
CumulativeVerificationScore	60	Cumulative verification score for this session.
Duration	3280	Duration of this utterance, In milliseconds
Decision	0	Optional Field: Tri-state value. 0 = inconclusive; 1 = accept; 2 = reject;
DecisionReason	"MORENEEDED"	Optional Field: Additional information about decision.
Gender	1	Optional Field: Tri-state value. 0 = inconclusive; 1 = Female/Youth; 2 = Male;
ChannelType	1	Optional Field: 0 = unknown; 1 = landline; 2 = cell; 3 = VoIP; more values?
LinguisticContent01	"8374019374182934"	Optional Field: ASR #1 most likely spoken content
LinguisticContent02	"8374019374082934"	Optional Field: ASR #2 most likely spoken content
LinguisticContent03	"8374019374082534"	Optional Field: ASR #3 most likely spoken content

The App sees that the account number did not match, so re-prompts the user for the account code.

Field Name	Value	Notes
IsValid	1	Whether utterance was used to create voice model. 0 = invalid utterance; 1 = valid utterance
ValidityReason	""	Optional Field: Reason utterance was invalid. E.g. ASR did not match requirements
VerificationScore	68	Verification score of this utterance. This is the verification score against an existing voice model.

CumulativeVerificationScore	64	Cumulative verification score for this session.
Duration	2950	Duration of this utterance, In milliseconds
Decision	0	Optional Field: Tri-state value. 0 = inconclusive; 1 = accept; 2 = reject;
DecisionReason	"MORENEEDED"	Optional Field: Additional information about decision.
Gender	1	Optional Field: Tri-state value. 0 = inconclusive; 1 = Female/Youth; 2 = Male;
ChannelType	1	Optional Field: 0 = unknown; 1 = landline; 2 = cell; 3 = VoIP; more values?
LinguisticContent01	"8374015374182934"	Optional Field: ASR #1 most likely spoken content

This was a good utterance, and was captured into the voice model. Now the App asks for the user's phone number.

Field Name	Value	Notes
IsValid	1	Whether utterance was used to create voice model. 0 = invalid utterance; 1 = valid utterance
ValidityReason	""	Optional Field: Reason utterance was invalid. E.g. ASR did not match requirements
VerificationScore	81	Verification score of this utterance. This is the verification score against an existing voice model.
CumulativeVerificationScore	69	Cumulative verification score for this session.
Duration	2420	Duration of this utterance, In milliseconds
Decision	1	Optional Field: Tri-state value. 0 = inconclusive; 1 = accept; 2 = reject;
DecisionReason	""	Optional Field: Additional information about decision.

Gender	1	Optional Field: Tri-state value. 0 = inconclusive; 1 = Female/Youth; 2 = Male;
ChannelType	1	Optional Field: 0 = unknown; 1 = landline; 2 = cell; 3 = VoIP; more values?
LinguisticContent01	"3127259298"	Optional Field: ASR #1 most likely spoken content
LinguisticContent02	"3127255298"	Optional Field: ASR #2 most likely spoken content

The utterance was good, and the SIV engine has determined that the user is successfully enrolled. This ends the session.

## 5.12 Multi-Factor Authentication

### 5.12.1 SIV Plus Knowledge

basic function: verification [4.1.1]

text-prompted mode

concurrent SIV and ASR processing [4.2]

In a conversational system it is possible to analyze the incoming speech both for ASR and speaker verification purposes. When a single question is asked and both biometric and knowledge evidence agree a user acceptance/rejection decision is easy (e.g. the case where the speaker verification biometric score is high and the answer is correct). However, if evidence is contradicting (e.g. high score, incorrect answer), it is beneficial to have a mechanism that decides dynamically on whether to accept the user, reject, or present a new question. In the ideal situation, this mechanism would also decide which question to ask.

The following example shows such a session resulting in acceptance:

C: Greetings. Please say your account number

H: 12345678

*<speech is recognized, and voice model 12345678 is retrieved. Also a profile that includes the answers to a set of security questions for user 12345678 is retrieved. >*

C: What is your favorite color?

H: Indigo

*<speech is recognized by mistake to be "I don't know", resulting in a wrong answer; however the SIV score is high. The system decides to ask an additional question>*

C: What is your social security number?

H: 123-45-9898

*<speech is recognized and the answer is determined to be correct. The updated speaker verification score is still very high. >*

C: You are verified. Welcome to the Securest Bank

### **5.12.2 Multi-Factor and Multi-Modal Authentication**

In this application the user begins the verification session online at a laptop or desktop computer. The screen displays a dynamically-generated alphanumeric security code. The system places a call to the landline phone at that location (location factor), requires a text dependent voice sample (knowledge and biometric factors), plus she/he is asked to read the security code on the screen (knowledge factor).

Desktop/laptop C: DISPLAY "SECURITY CODE: 1 2 L 4 7 Q"

C: <voice model for user is retrieved>

C: < calls landline phone at location. It has already rect>

C: Please say your name

H: Mary Smith

C: please say the security code displayed on your screen

H: 1 2 L 4 7 Q

C: < uses speech recognition to process sequence>

C: Thank you. You have been authenticated.

## **6 References**

### **6.1 Related Documents**

[Apps] "Speaker Verification and Identification Applications" by VoiceXML Forum Speaker Biometrics Committee

[Architecture] "Speaker Verification and Identification Architectures and Data Structures" by VoiceXML Forum Speaker Biometrics Committee (in preparation)

[Best Practices] "Speaker Verification and Identification Best Practices Document" by VoiceXML Forum Speaker Biometrics Committee (in preparation)

[DEFF] "Data Exchange File Format for SIV" by VoiceXML Forum Speaker Biometrics Committee

[Glossary] “Speaker Identification and Verification (SIV) Glossary” by VoiceXML Forum Speaker Biometrics Committee

[Introduction] “Introduction to Speaker Verification and Identification” by VoiceXML Forum Speaker Biometrics Committee

## **6.2 External References**

*ANSI X9.84 Biometric Information Management and Security for the Financial Services Industry* <http://www.ansi.org> (go to the standards store)

*BioAPI version 1.1* <http://www.bioapi.org/DownloadsPage1.html>

Biometric Consortium <http://www.biometrics.org> (biometric standards activities, general biometrics information)

*Biometric Evaluation Methodology: Common Criteria Common Methodology for Information Technology Security Evaluation* <http://www.cesg.gov.uk/> (go to “Applied Security Technologies” and select “biometrics”)

*Canadian Privacy Act* <http://laws.justice.gc.ca/en/P-21/text.html>

*CBEFF* <http://www.itl.nist.gov/div895/isis/bc/cbeff/>

Common Criteria Biometric Evaluation Methodology Working Group *Biometric Evaluation Methodology: Common Criteria Common Methodology for Information Technology Security Evaluation* (<http://www.cesg.gov.uk/>. (go to “Applied Security Technologies” and select “biometrics”)

“Cross Jurisdictional and Societal Aspects of Biometrics” – working document by ISO SC37 WG6 (*Non-Technical Aspects of Biometrics*)  
<http://www.itoc.usma.edu/workshop/2005/bios/biometrics.html>

European Union Data Privacy Directive *95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data*  
<http://www.privacilla.org/business/eudirective.html>

IBG BioPrivacy Initiative <http://www.bioprivacy.org/>

IBIA Privacy Principles <http://www.ibia.org/aboutibia/privacyprinciples.asp>

ISO 19092 *Financial services - Biometrics - Part 1: Security framework*  
(<http://www.ansi.org>)

MRCP version 2 draft 12 <http://www.ietf.org/internet-drafts/draft-ietf-speechsc-mrcpv2-12.txt>

NIST Special Publication 800-58, *Security Considerations for Voice Over IP Systems*, <http://csrc.nist.gov/publications/nistpubs/index.html>