

InterpreXer™ 2.0

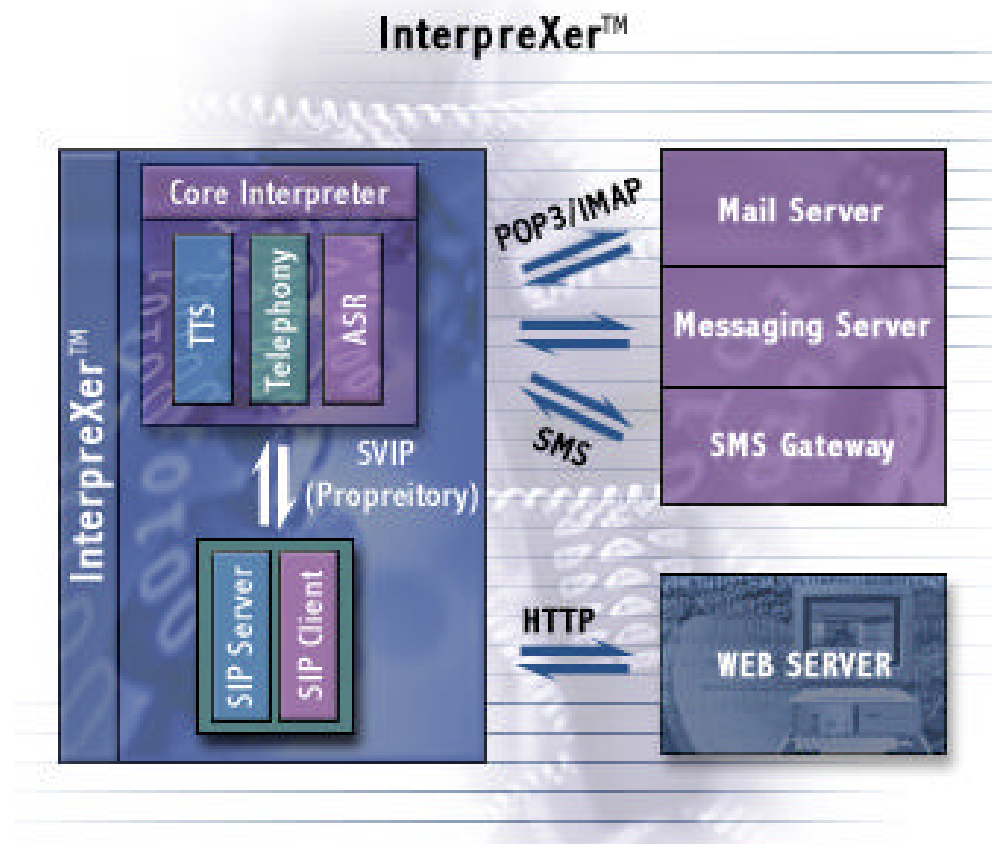
Phonologies VoiceXML 2.0 Browser

Overview

Phonologies' InterpreXer™ is a VoiceXML 2.0 compliant Voice Browser, based on open Industry standards like VoiceXML and SIP. InterpreXer™ Open and Distributed Architecture provides an easy interface to integrate any third party ASR, TTS and Telephony platforms, making it highly flexible and scalable.

InterpreXer™ Distributed Architecture is based on a client - server model and consists of five main components: Interpreter Server, RecServer, TTSObject, one or more SIP Channel Servers and an Web Application Server (Remote). These servers can be deployed in different configurations and combinations depending on the requirements.

At the core of the InterpreXer™ is the Interpreter Server, which fetches documents from the Application Server and initiates a dialogue with the user. InterpreXer™ uses various "interfaces" for Recognition, Prompt Playback and Call control functionality. These interfaces are highly flexible and can be programmed to work with any third party ASR / TTS / Telephony Platforms. Thus, being independent of ASR / TTS / Telephony platforms, InterpreXer™ can work with any network architecture.



InterpreXer™ 2.0

Phonologies VoiceXML 2.0 Browser

Features

Vendor Independent

InterpreXer™ is the first VoiceXML 2.0 compliant browser based on an Open and Distributed Architecture, giving Platform Integrators and Value-added Resellers the flexibility to integrate with platforms of their choice. Currently it is available pre-integrated with ASR Engines of Nuance and IBM, TTS Engines of AT&T and IBM and Telephony platform of Cisco.

VoiceXML Extensions

InterpreXer™ provides “hooks” to integrate with messaging servers like SMS, FAX and Mail. It is possible to send SMS / FAX / MAIL messages just by using VoiceXML tags. It also provides hooks to extend the logging interface.

Cache Management and Cookies

InterpreXer™'s Internet Interface fully implements HTTP/1.1 Caching specifications. This decreases waiting time by caching frequently used prompts and grammars, improving user experience. InterpreXer™ also provides Session-based cookie functionality by inheriting the web browser functionality and provides JavaScript functions like SetCookie() and GetCookie() to add, delete and manage cookies.

OA&M and Reporting Services

All elements of InterpreXer™ are SNMPv3 enabled and can be monitored using any SNMP agent. InterpreXer's OA&M suite can be used to configure these elements, monitor their health status, and generate reports. These elements generate traps indicating critical or major alarms. It provides access to near realtime CDR logs and application logs, which can be exported to different formats.

Security

InterpreXer™ supports industry standard security protocols HTTPS and SSL making it an ideal platform for V-commerce transactions.

Browser Plug-ins

InterpreXer™ provides Open API to extend its functionality by adding User defined browser plug-ins. These Plug-ins are a way of extending the <object> tag and are implemented as shared objects. Each plug-in is mapped to one or more mime-types. When, Interpreter comes across an <object> tag it executes appropriate plug-in depending on its mime-type. InterpreXer™ provides a tool, the "Plug-in Manager", to configure and manage these plug-ins.



Phonologies

The Voice of Technology

InterpreXer™ 2.0

Phonologies VoiceXML 2.0 Browser

Features (Contd...)

Recording-on-Demand

InterpreXer™ has a built-in audio logging facility to save user utterances corresponding to a particular prompt for enhancing prompts and tuning the Speech Recognition Engine.

IP-ready Platform

InterpreXer™ uses SIP for internal signaling and also supports other VoIP protocols like H.323 and MGCP / Megaco. Phonologies' InterpreXer™ is a "plug-n-play" platform that can work with any IP enabled Telephony Platform.

Rule based Call Routing

InterpreXer™'s SIP-Channel executes specific VoiceXML applications based on user-defined rules like time-of-the-day, ANI, DNIS etc. The administrator can define multiple rules and prioritize them, and even "busy-out" the SIP Channels for maintenance or block users based on ANI / CLI.

Grammar Management

InterpreXer™ validates all the grammars(GrXML and ABNF) before sending them to the ASR Engine, and it also shows any inconsistencies. It supports Hotword, dynamic grammars and barge-in.

Scalability

In a Client-Server architecture our system can easily be scaled by setting up server farms or increasing CPU and memory of the existing server units

Resource pooling

Interpreter connections are pooled and are not bound to a particular telephony channel. This ensures availability of the application and optimal use of resources.

Internationalization

InterpreXer™ is language independent, and supports ASCII, Latin-1 and UTF-8 Unicode character sets (SBCS & DBCS) for VoiceXML content (including prompts and grammars).



InterpreXer™ 2.0

Phonologies VoiceXML 2.0 Browser

Interpreter Server

Interpreter Server is the heart of the InterpreXer™, managing all interfaces of the Voice Browser. It is developed in C / C++ and runs on a Linux Box.

Operational flow, when the platform receives a call:

- The SIP-Channel connects to the Interpreter Server that is available and can serve (Interpreter Server is not directly "tied" to any physical Telephony Port).
- The SIP-Channel then passes the call information along with the starting VoiceXML URL to the Interpreter Server.
- After this, the Interpreter Server fetches the VoiceXML document and initiates a dialogue with the user.

Interpreter Server consists of six main subsystems:

- XML Parser – to parse VoiceXML, GrXML and SSML documents.
- ECMAScript Interpreter – to evaluate ECMAScript expressions.
- DTMF Recognizer – to recognize DTMF stokes in compliance with VoiceXML 2.0, supporting GrXML and ABNF Grammars.
- Tornado (an Internet component) for document fetching, caching and cookie management.
- Resource Manager – Manages all the media resources.
- Logging

In addition to these, the Interpreter Server has interfaces for Recognition, Prompt Playback and Call control.

Interpreter Server, while in a call, can track and log the dialogue path (flow) followed by a user. This information can be very useful in determining user behavior, tastes and preferences, which helps in developing user-centric applications and increased customer satisfaction.

Interpreter Server can be configured to use multiple RecServers and TTSObjects. When more than one server is available, Interpreter Server connects to the server with maximum resources.

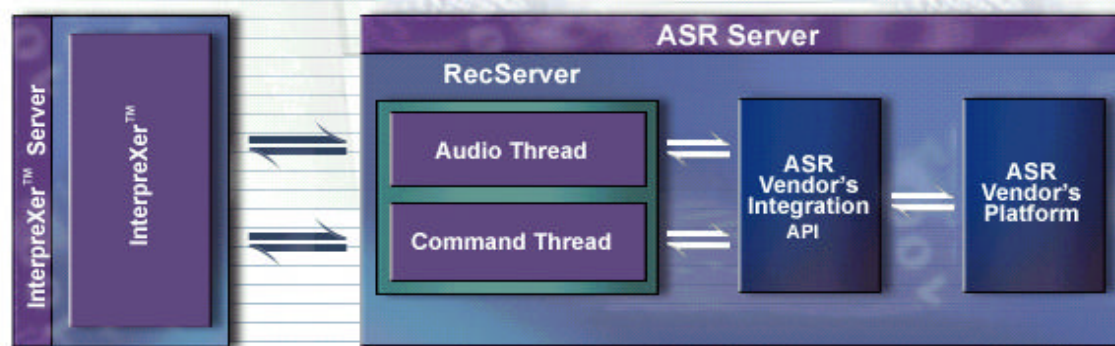


InterpreXer™ 2.0

Phonologies VoiceXML 2.0 Browser

RecServer - Automatic Speech Recognition Interface

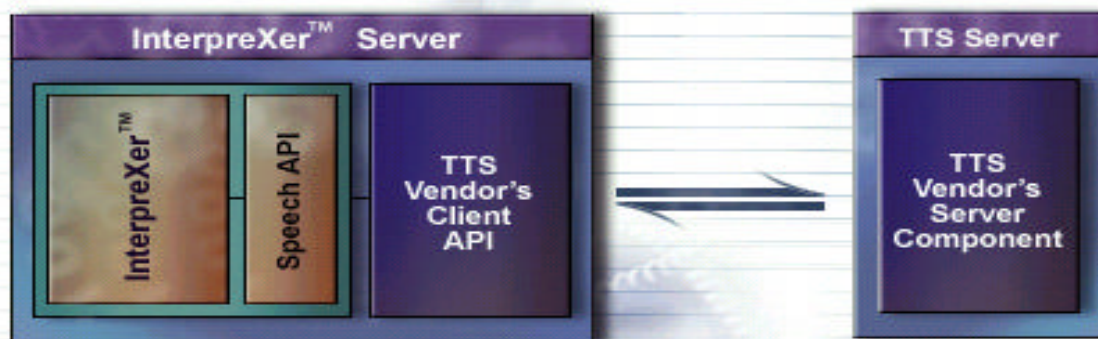
InterpreXer™ provides Open Interface API to integrate any ASR engine. "RecServer" is the implementation of this Interface API, which acts as a mediator between the Interpreter Server and the ASR Engines. The RecServer program runs two threads, one for Audio Data transfer, and the second for accepting commands and manipulation of grammars etc. Interpreter Server has a recognition client, which communicates with the RecServer through a text protocol derived from HTTP.



RecServer has been designed keeping in mind modern ASR engines with GrXML / ABNF support, as well as legacy ASR engines with proprietary grammars. RecServer is a lightweight program that can run on the same Server as the ASR vendor's platform as well as on a different server.

TTSObject - Text-to-Speech Interface

InterpreXer™ TTS interface implementation is slightly different compared to ASR interface. Most vendors of TTS Engines ship a Client - Server program with the package. Thus, the integrator needs to implement the Vendor's client API using InterpreXer™ TTS API, as a shared object (TTSObject) and hence, there is no need to write a Server Program. Unlike RecServer, the TTSObject resides on the Interpreter Server.

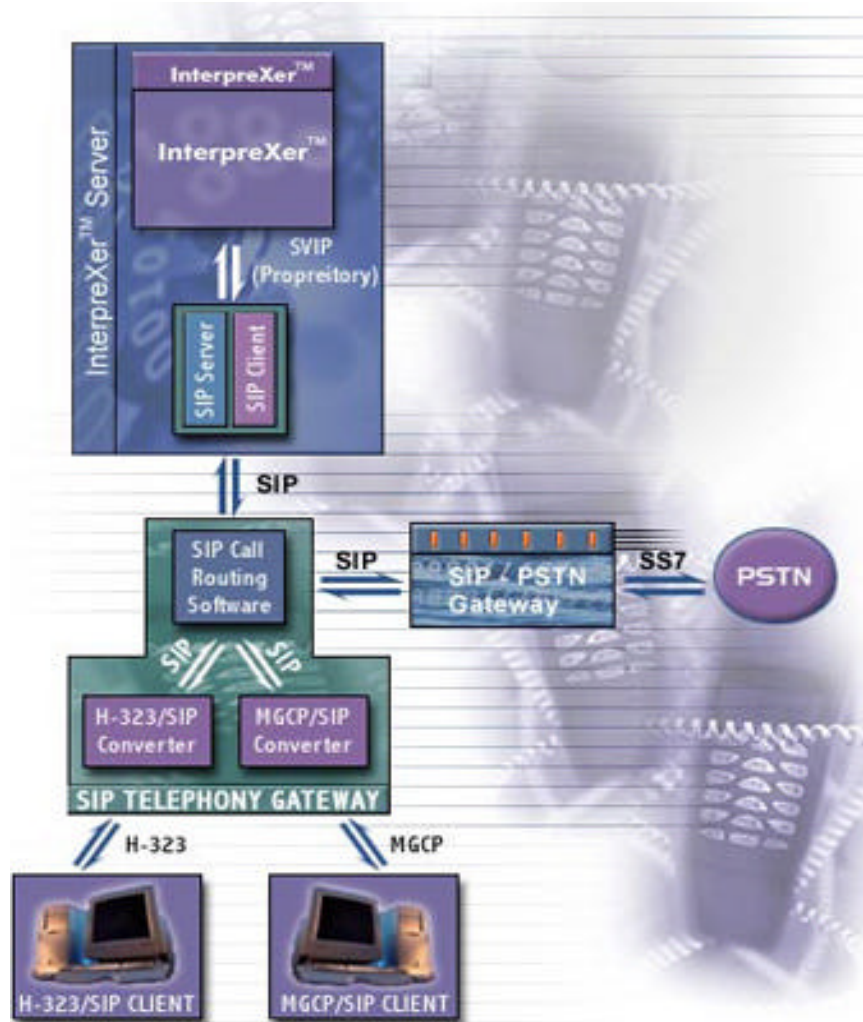


InterpreXer™ 2.0

Phonologies VoiceXML 2.0 Browser

SIP Channel - Telephony Interface

SIP-Channel is the telephony interface of the InterpreXer™ and can be thought of as a SIP-Softphone waiting for a call. Each SIP-Channel has a SIP URL and is mapped to a particular VoiceXML URL. SIP-Channel can be contacted by a SIP-Softphone or from a PSTN Phone through a VoIP Gateway, and can be monitored and controlled by third-party call control applications (as shown in the figure below). When SIP-Channel receives a call, it sends a request to the Interpreter Server. The SIP-Channel then sends call information like ANI, DNIS and the initial VoiceXML URL (based on user-defined rules) to the Interpreter Server. All communication between Interpreter Server and SIP-Channels is through a proprietary text protocol "SVIP". Once the Interpreter Server initiates a dialogue with the user, SIP-Channel will be used for audio playback, recognizing DTMF strokes and capturing audio for recognition etc. SIP-Channel is also responsible for sending routing requests to third party call control applications, and effecting a call transfer (blind and bridge).



InterpreXer™ 2.0

Phonologies VoiceXML 2.0 Browser

Specifications

Application Protocols

- VoiceXML
- GrXML
- SSML

Signaling

- SIP
- H.323
- MGCP
- TCP/IP

Speech Technology

- ASR
 - Nuance 8.0
 - IBM Speech Technology
- TTS
 - AT&T Natural Voices
 - IBM Speech Technology

Operating Systems

- Linux 7.2

Standards

- HTTP/1.1
- SNMP v3
- SIP
- H.323
- Megaco / MGCP
- RTP
- ECMAScript

Management

- Event Log
- Provisioning
- Configuration
- Performance
- Dialog Tracking
- Audio logging

"Phonologies™", "The Voice of Technology™", and "InterpreXer™" are registered trademarks of Phonologies (India) Private Limited. Other company products and service names may be trademarks or service marks of others.



Phonologies

The Voice of Technology