Dialogic

White Paper

Developing and Deploying Next Generation Interactive Response Solutions

Executive Summary

Interactive Voice Response (IVR) systems have been a mainstay application for the delivery of automated self-service in the telecommunications industry for well over three decades. IVR not only provides automated self-service, but also serves as a portal or access point for a range of communication applications, with the most notable being agent-assisted contact centers and conference bridges. While IVR may appear to be a mature solution, there are exciting new things on the IVR horizon. As fixed and wireless voice communications networks, protocols, and endpoints continue the rapid transition to packet-switched IP and true end-to-end IP communication sessions, new opportunities are emerging to innovate around interactive response beyond basic voice communications. With this, there comes a need to enable new ways to develop and deploy solutions that align with today's IP centric world and development communities.

This whitepaper covers new communication channels for interactive response and profiles as well as new development and deployment options that leverage the trends in networks, media processing and computing technology; it also reaches out to the solution developers who seek to innovate around these opportunities and into these networks.

Table of Contents

Overview: Interactive Response – It's not just narrowband voice anymore

Interactive Voice Response (IVR) systems have been a mainstay application for the delivery of automated self-service in the telecommunications industry for well over three decades. IVR not only provides automated self-service, but also serves as a portal or access point for a range of communication applications, with the most notable being agent-assisted contact centers and conference bridges. While IVR may appear to be a mature solution, there are exciting new things on the IVR horizon. As fixed and wireless voice communications networks, protocols, and endpoints continue the rapid transition to packet-switched IP and true end-to-end IP communication sessions, new opportunities are emerging to innovate around interactive response beyond basic voice communications. With this, there comes a need to enable new ways to develop and deploy solutions that align with today's IP centric world and development communities.

Multi-Modal Interactions Aren't New

The evolution of automated self-service applications has also already seen the embrace and integration of new modes of communication beyond traditional voice as they have become established in consumer and business markets – in other words, this is not new. Fax, e-mail, and web chat are among the notable modes of communications that have been and continue to be widely used to automate customer interactions and communicate relevant and timely information. And of course, website interactions themselves have had among the most significant impact on enabling customer access to information and self-service for all types of transactions.

Text Messaging

In mobile networks, SMS or 'texting' has become a well-established and often preferred mode of communication for many young people, displacing voice and e-mail as their primary form of interacting with their peers, and often with their parents as well. More recently, SMS use has successfully migrated from personal use and into business applications for location-based solicitations, proactive customer service notifications and transaction confirmations. Its efficiency in network use and as a communications tool is well recognized, so integrating SMS into automated self-service is seen as a logical step towards engaging customers via a cost-effective and convenient communications mode which they have already accepted and are comfortable using, particularly for simple transactions and messages. While mostly independent from the migration to underlying IP networks, SMS nevertheless is a mobile communication mode with ongoing opportunities for automating customer interactions.

Smartphones and Tablets

Looking beyond SMS in the mobile realm, the reach of IP networks and protocols is also extending rapidly over Wi-Fi and 4G networks, opening the door for real-time end-to-end IP communications sessions, particularly from smartphone and tablet devices. Full, multimedia soft clients for these devices are appearing in app stores, enabling over-the-top IP communication sessions. Many organizations servicing large consumer bases have developed their own apps for these devices to more deeply engage these customers and leverage the rich functionality the mobile devices offer. Whether the path to servicing customers with smartphones and tablets is ultimately through dedicated apps, or instead through the mobile browser remains to be seen. However, interactive response solution providers need to consider the smartphone user as a multimedia-enabled endpoint - HD voice and video capable - and perhaps eligible for a new level of self-service through which rich, multimedia self-service interactions with service providers and businesses will be *expected*.

Those who have experienced the intuitive user interface and responsiveness of the latest generation of iPhones and Android-based smartphones will understand the passion they evoke from their rapidly growing user bases. Generally speaking, the most useful and compelling apps tend to be those which are well integrated into the capabilities of the device, leveraging technology such as global positioning, mobile voice, mobile internet, mobile video and the viewing orientation of the device. Whether an interactive response system includes a corresponding smart phone client app, or relies on the smart device browser as the local interface, leveraging the balance of the available functions to engage the smartphone user in a manner that delights is becoming a "best practice" to follow.

Speech Interface Status

Not to be overlooked, speech recognition is an increasingly popular user interface, and ongoing improvements in recognition, implementation and application are helping to overcome technological limitations that in the past had produced user frustration. Siri, Alexa and Cortana are examples of natural speech interfaces that allow access to content and information. The movement to mobile devices in general is increasing the usage of speech technologies by enabling more hands -free ('touchtone-less') operation, which often can be more convenient than using touch tones when talking on a mobile phone. Speech recognition also in some cases allows users to comply with local laws, requiring hands-free operation of a communications device while operating a motor vehicle.

Speech may also become another means of controlling and navigating smartphone applications and multimedia sessions that are delivered from a network based, interactive response server. As smartphones become avenues for mobile payments, speech recognition may even be used as a source of verification, such as a voice print ID, in place of a PIN where an individual's unique speech tones, like a fingerprint or other biometric marker, will serve as a secure verification code.

Unified Communications

While VoIP is being widely used to replace PSTN voice calls across public and private networks, the industry is moving to interconnect the 'islands of VoIP' directly, and to eliminate the narrowband bottleneck of circuit-switched networks. As IP networks are extended and peered, they begin to afford more end-to-end, real-time IP communications sessions between customers and the service and solution providers on which they depend. With this, new opportunities are emerging to leverage real-time IP modes of communications for customer self-service.

As an example, the contiguous set of unified communications (UC) tools can come into play. UC clients that enable presence, IM, high fidelity HD Voice, video, web collaboration and conference from a single interface can be leveraged for new types of automated interactions. While today a web browsing session can be enhanced with agent text chat sessions and voice via 'click-to-dial' (with callbacks placed over the PSTN), it is becoming more realistic to initiate a full, interactive multimedia session through a UC client that allows web browsing, presence (knowledge expert / agent availability), instant messaging (chat), voice, video, and web collaboration. Not surprisingly, it is now easy to see how a contact center agent could gradually engage a customer and elevate the interaction to a multimedia session that ends successfully for both parties.

While UC clients are usually enabled by the SIP protocol, dominant peering UC clients such as Microsoft's Skype and Skype for Business can be leveraged in this way as well. Given the large, global numbers around Skype users and Skype traffic, and the extensive use of HD voice and video in connection with Skype, it would appear to be a natural communications channel for interactive response developers to exploit.

Social Networks

The intersection of interactive response and social networks is emerging as well, with marketing and support organizations hard at work to determine how best to leverage this new communication channel with their customers. Whether customers will choose to use Facebook, LinkedIn, Twitter or other social networking sites to directly interact with an organization depends on factors such as how the communications channels within the social networking sites evolve beyond messaging and chat. At this stage, social networking channels are being used to 'like' as well as 'flame' organizations, but much of that activity is at a public broadcast level rather than via a one-to-one interaction. Of course, both are an invitation to engage, whether to deepen a 'like' to true brand loyalty or to dowse a 'flame' before it ignites a public relations fire.

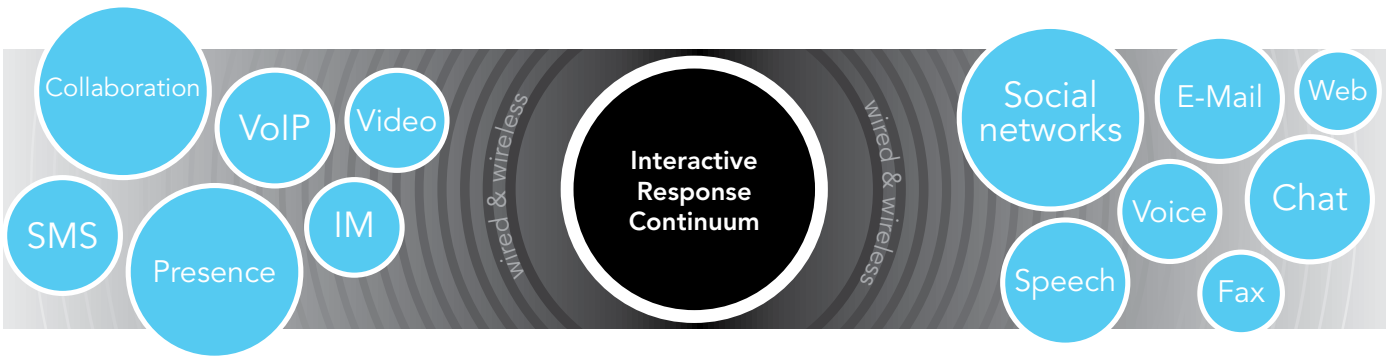

Figure 1. Interactive Response channels continue to expand to accommodate and leverage new communications modes as they emerge in wired and wireless networks.

Rolled Up

All of the channels shown in Figure 1 can lead to an expanded concept of automated customer interactions, as well as the multi-modal communications continuum for those interactions. It's not just voice anymore, and that creates new opportunities to deliver more innovative and effective self-service solutions.

Interactive Response Decision Engine

Given the scope of communications channels by which an organization can (or may have to) interact with its customers to deliver automated self-service, a decision engine is needed. The response decision engine manages the interaction based on the communication mode initiated, customer profile (contact info), and customer preference, which are learned or requested. While some customers may be best served via voice, DTMF interface and live agent options, others may be better suited with a mobile application that enables chat, voice, video and SMS. An interactive response decision engine that chooses the correct method of engagement based on the situation and customer preference allows an organization the flexibility to provide a customer experience that matches expectations.

A customer engagement decision engine must have at its disposal the media and signaling technologies capable of delivering the messages and initiating or responding to interactive sessions requests, be it by narrow-band voice, text messaging, chat, or a full IP multimedia session. Considering the range of communication channels an organization may want to leverage, abstracting the media processing and signaling channels from the application logic of an interactive response system may be advised, in turn enabling new communication channels to be added to a solution as needs dictate.

Interactive Response Development Options

Traditional Interactive Voice Response and IVR toolkits have been developed using computer telephony software development kits (SDKs) and C-level programming APIs. While low-level, high-control programming interfaces remain a robust and available means of developing next generation interactive response systems, many development communities exist with expertise in newer programming models, particularly those that have evolved and emerged from web development models.

Benefits of using newer programming approaches to Interactive Response development include being able to develop more rapidly and within a simpler development environment. These benefits are helping to bring Interactive Response capabilities and techniques to new development communities, enabling more integration with business process applications and services, or 'mashed up' with web-based consumer applications.

Development Options and Deployment Models - Evolution

Development options and deployment models for interactive response have evolved over the past decade. Virtually all of the generational programming models are still viable options for developing interactive response solutions. The choice of a development approach today is driven by factors such as the complexity of the solution, density, deployment architecture and the business model necessary to bring the solution to market.

Legacy Computer Telephony APIs and All-in-One Architectures

Computer Telephony (CT) was at the heart of many legacy IVR applications. IVR is about *computer* databases being queried by *telephony* systems that deliver data audibly to the user. Traditional CT vendors enabled this process via specialized computer telephony boards made up of hardware, software and APIs designed to be integrated in a single server with the application co-resident. This first-generation "all-in-one" design had proven particularly efficient for low to mid density solutions (4 to 120 channels) deployed on customer premises and interfaced with a PBX or directly to the PSTN. The advent of IP telephony and software-based media has reduced the need for specialized hardware (which processed media on DSPs and provided TDM and analog interfaces to circuit-switched networks). The all-in-one architecture persists, and has become much more efficient as the entire media and application process is software based and can be virtualized into private or public cloud deployments. This pure software approach can simplify development and deployment in many ways.

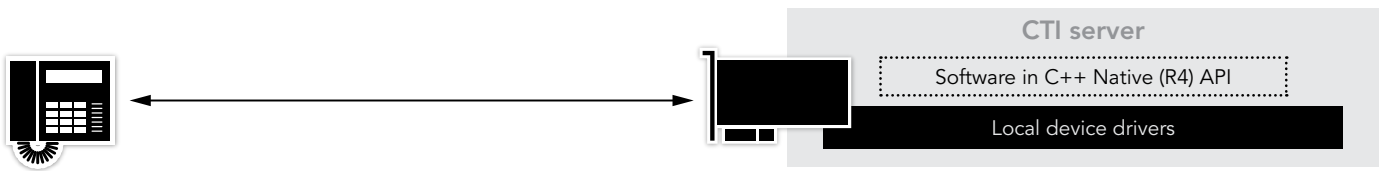

Figure 2: Legacy CTI Server (board-based media and signaling)

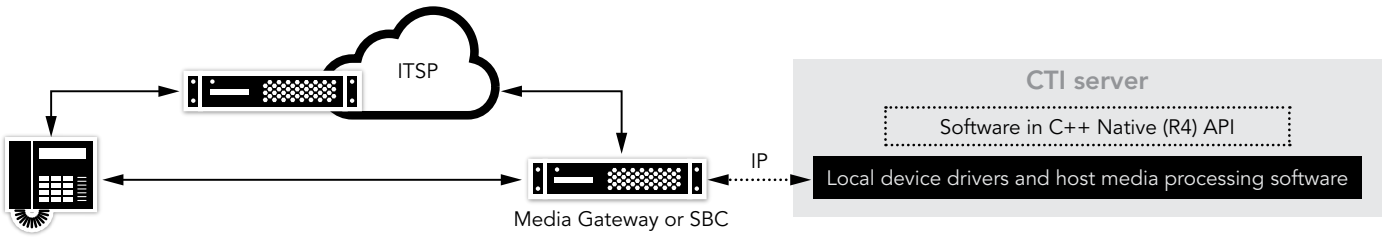

Figure 3: Legacy CTI Server (software-based media and IP signaling)

In legacy CT "all-in-one" architectures, the following conditions generally exist:

- Signaling (SIP, SS7, ISDN, ...) and media (voice and video) processing are managed on the same server
- One integrated software application encompasses signaling, media control and business logic
- The integrated application is programmed using a proprietary (e.g., C language) Application Programming Interface (API)

The "all-in-one" design and deployment approach remains practical and efficient for low to mid density deployments, and can even help solution providers make the transition to hosted and cloud communications through virtualized, all-in-one IVR servers allocated per subscriber.

Standard Media Control Protocols, Application Servers and Distributed Architectures

The Legacy model of developing and deploying interactive response imposes a variety of limitations for today's solution providers. For high density, hosted and cloud-based solutions, a distributed architecture is desirable and IP communications protocols such as SIP and RTP generally lend themselves to distributed architectures more easily than traditional circuit-switched environments and protocols. Decomposing the solution set can enable greater scalability and reliability, as redundant elements can be deployed and taken in or out of operation without disrupting service. To enable distributed architectures, the application and media processing are often separated, and new standard media control protocols are employed. Network connectivity is moved to border elements, such as media gateways and session border controllers, which can provide further deployment flexibility, and reliability.

On the interactive response development front, a variety of higher level programming environments for interactive response are being used to simplify the developer's task and to help reach a wider developer audience. Becoming more common and mainstream is the use of an application server and a separate media server. The media server paradigm is different than the 'traditional' way of implementing voice and video services in several ways, and this document explains the underlying concepts, and the reasoning behind the use of separate application servers and media servers.

When using a media server, the media processing is performed by the media server and the signaling and business logic reside on a separate platform, namely the application server. There is not necessarily a fixed relationship between signaling resources and media resources. Media resources on the media server are controlled using a standardized API, possibly via XML-based messages, which, in turn, are programmed by software on the application server. Popular languages for programming the application server include C/C++ and Java.

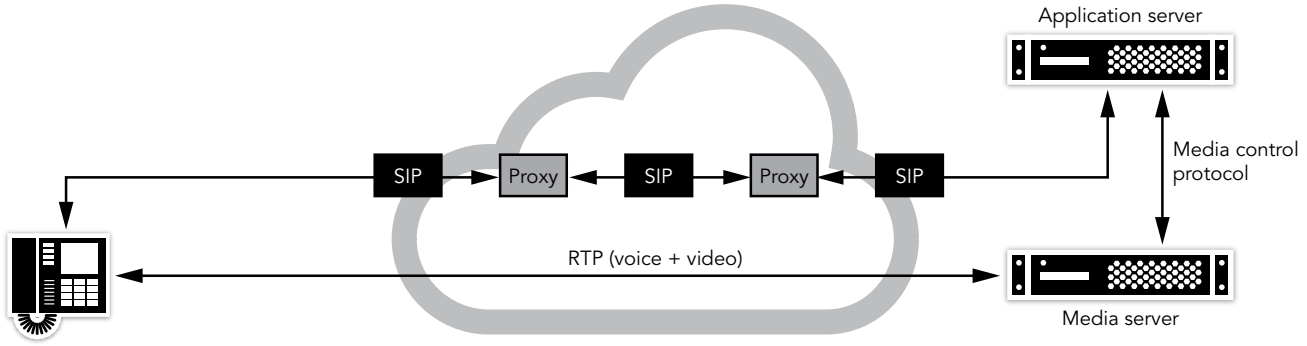

Figure 4: SIP Application Server and Media Server Architecture

The Application Server

The "business logic" is programmed in an application that runs on an application server, which also handles the caller's SIP call. When SIP calls are made in this architecture, however, the voice traffic (and video data, if present) is *not* routed through the application server. When an application server processes two SIP call legs – one with the user's phone, the other with a media server – the two media-capable devices are each told to transmit RTP data directly to each other. The application server uses a configuration known as a SIP Back-to-Back User Agent, and the process of exchanging audio, video capabilities and IP addresses works using the Session Description Protocol – or SDP. In addition to signalling, the application server is responsible for managing the user accounts, call routing logic, and – importantly – billing.

In practice, the application server and media server will most likely be supported by a number of SIP proxies intermediating between a caller and the application server. These may be registrars, session border controllers, and, in the case of service provider networks adhering to the IP Multimedia Subsystem architecture (IMS), would include "Call Session Control Functions" – or CSCFs. And of course, if the caller is on a non-IP network, there will be a media gateway or a similar intermediary device translating TDM to VoIP in the architecture as well. However, for purposes of this document, we will defer discussion of the role of such intermediary devices to the section below on distributed architectures.

An advanced application server also generally will support parallelism, load-sharing, redundancy and high-availability, application staging and fall-back, logging, debugging, and a wide range of important operations requirements. The media server paradigm allows telephony applications to reside in the same space as existing (web) application servers, with specialized media tasks provided by the media server itself.

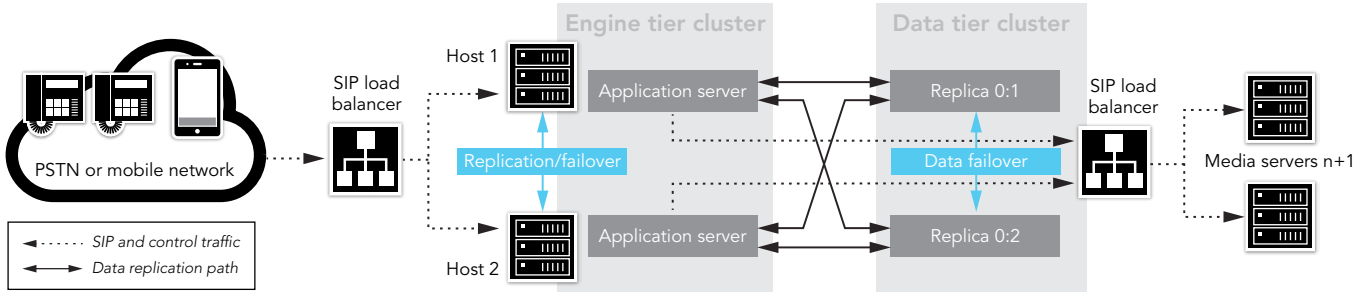

Figure 5: Distributed Architecture

Controlling the Media Server

With the role of the application server having been defined above, examining and comparing the methods for controlling the media server from the application server is covered next.

The application server can control the media server using various methods. This document will examine two principal methods, VoiceXML and MSML, as well as a third, emerging Java-based method: Java Specification Request 309 (JSR309). The choice between VoiceXML (VXML), MSML and JSR309 is not completely unrestricted and depends on factors such as the features of the service that the Interactive Response developer wants to provide.

- VXML
  - XML-Scripted Dialogues
  - Automatic Speech Recognition (ASR) and Text to Speech (TTS)
  - Video with Text Overlay
  - Real Time Streaming Protocol (RTSP) with VCR controls
- MSML
  - Enhanced conferencing
  - Programmatic control of IVR functions
- JSR309
  - Java Application Software
  - Programmatic Control of IVR Sequence
  - Media server interface independence (MSML/MSCL/MSCML/Proprietary)

When to consider VXML

A common reason to use VXML to control a media server is to take advantage of the off-line VXML design tools. Other reasons may include delegating the dialog implementation away from software engineers, or the need to not embed dialog structure in the application software (for example, where each dialog change would involve an application upgrade). When using Speech Recognition and TTS in interactive response dialogues, or when taking advantage of text overlay with video output, it is seen as "best practice" to use VXML to control the media server. This is also true when implementing a fast forward feature, or pause and rewind controls with an RTSP-sourced video or audio stream.

Because VXML is a W3C standard (World Wide Web Consortium), several organizations have developed tools that make it easier to create and manipulate VXML scripts:

- OpenVXML by OpenMethods
- xMP by Vicorp

When designing human-computer dialogues, a good deal of testing and tweaking of the dialogue occurs both before release and in response to quality feedback. This is often best done by a specialist who is not necessarily a software engineer, so intuitive design tools like those listed above can end up being very useful. Using design tools also can help lead to rapid development and optimization of bug-free dialogues, potentially in parallel with the design of application server software.

When to consider MSML

MSML is used to control the advanced conferencing features of a media server. Interactive Response dialogues can also be controlled programmatically using MSML. It may be necessary to have fine control of the dialogue in the application server program, in cases where the logic is too obscure or complex to be managed by VXML or a sequence of VXML scripts. (To make this decision, studying the VXML specification and some of the excellent tutorials and examples on the web may help.) To use MSML, the application software has to be capable of formatting commands in XML documents and interpreting the XML script that comes back from the media server. There are open source libraries that can help with this, and it is straightforward in Java. An available SIP stack will also be needed, but in most cases the application server will already have a SIP stack since it is typically communicating with other SIP user agents to handle the incoming and outgoing calls.

When to consider JSR 309

Java Platform, Enterprise Edition (Java EE or J2EE) servers are popular platforms for both enterprise and service provider applications. Java EE servers offer developers a rich set of services such as directory integration, reliable messaging, and easy-to-use SIP and HTTP protocol functions for communications. Many general-purpose SIP application servers are Java EE servers, from such companies as Oracle, IBM, Redhat, and others.

Because the Java EE platform is popular for use with communications applications, it is important for the platform to have the ability to control a media server. The dilemma has been which API to use. Should it be an XML-based interface such as VXML, CCXML, MSML, MSCML - or MSCL (a newer media control protocol, which the IETF is developing)? Instead of requiring a specific media server control interface choice from among the existing standards, JSR309 defines a standard Java interface for media server control.

JSR309 defines an abstract Java interface for the manipulation of audio and video streams and conferences. The interface design is very rich and general. Vendors of media servers, such as Dialogic, provide implementations that work with their respective media servers. If needed, vendors can develop their own JSR309 interfaces separately or over existing media server interfaces, such as MSML, MSCML, MSCL, or even via proprietary Media Server APIs (for example, a new RESTful API).

The JSR309 architecture assumes a distributed or IMS-like model where the Java EE server and media server reside on separate physical servers. Communications devices interact with the Java EE server and application through the SIP protocol. SIP Servlets (aka JSR289) enable Java EE applications to communicate easily using the SIP protocol. The application uses the JSR309 connector to control the remote media server.

JSR309 can make the adoption of media servers significantly easier for Java developers by delivering media control within a familiar environment so as to help reduce development time.

Choosing an Application Server

Here is a review of important characteristics of a SIP Application server:

- SIP stack (helpful if it's mature)
- SIP Back-to- Back User Agent (B2BUA – helpful if it has a straightforward API)
- Unless the application can be defined entirely by VXML and general-purpose logic, the app server should control the IP media server
- Media server control will involve SIP and XML manipulation
- In the case of Java, media server control can be the JSR309 API

The business logic of the application itself will have to be programmed, and the resulting application will most likely need access to external components such as databases or web services from other platforms. And the deployment platform - consisting of application server and media server(s) - will need important operational components, including security, high availability provisioning, logging, maintenance, application staging, etc.

Depending on the type of application developed and the deployment model chosen to deliver the resulting service, there may be other requirements for the application server, such as delivering HTTP content or providing a web services interface. What it boils down to is an application server is software, which means that given enough time, skill, experience and testing resources, it should be possible to develop the required application server functionality and robustness using a variety of approaches and programming languages.

However, for the purposes of this discussion, the options will be categorized as the Java Enterprise Edition (Java EE) application servers, and "*others*", namely other readily available proprietary application server software.

- Java EE
  - SipServlet JSR289
  - JSR309
- The "Others"
  - C / C++
  - Open Source
  - JAIN SLEE
  - Python
  - Perl

Because of space limitations, this document will not expand further on the "others" beyond what was covered above.

Application Servers: Java EE

Java Enterprise Edition (Java EE) is an environment that runs application software written in Java, bringing with it a vast library of objects that do many of the things touched upon within this document. As Java EE has evolved, it has provided some of the largest, reliable web-based services and has embraced SIP for telecommunications signaling. Mature Java EE Application servers generally have many years of development behind them and represent a collection of reliable components. There are several free-to-use, open-source Java Enterprise Edition Application servers that support SIP, as well as several commercially supported ones.

A Java EE Application server is standardized software that runs in a Java Runtime Environment and is, not surprisingly, written in the Java programming language. Many Java EE Application servers are capable of SIP signaling, as well as HTTP and numerous other interface protocols; hence, SIP-based telephony applications can be hosted on a Java EE platform with media capabilities – audio, video, conferencing and speech technology – being provided by an associated media server. In such a case, the Java EE platform amounts to a very large code base and has evolved into its current form due to the involvement and input from many entities.

The software on the application server is also ultimately responsible for implementing or controlling the other aspects of an application. As noted earlier, the application server could be using web services from another organization, or, in a service provider application, performing authentication, authorization and accounting, using the Diameter protocol. The application server may also be responsible for providing content

over HTTP or other network protocols (e.g., providing audio and video content and possibly dynamic VXML scripts) for interpretation by the media server. Many of the disparate roles can be delegated to separate web servers, and they are likely to be programmed in the same language and run on similar platforms.

If the application server and IP media server are to be deployed in an IMS architecture, there are Java libraries that provide access to other IMS components, such as Diameter servers and Presence servers, as well as helpful information for the important implementation of back-to-back user agents.

Java EE application servers (open source and licensed):

- Open Source Java EE Application servers:
  - SailFin
  - Mobicents
- Commercial Java EE Application servers:
  - Oracle Communications Converged Application server
  - IBM Websphere
  - Jboss
  - Redhat

Speech and Text-to-Speech Integration

Automated Speech Recognition (ASR) and Text-to-Speech (TTS) have continued to gain market acceptance as telephony user interfaces for interactive response systems. As the quality of speech engines has continued to improve, new protocols have gained market acceptance to help simplify the integration and use of the technology for self-service across a wide range of industries.

One such protocol is the Media Resource Control Protocol (MRCP), an RFC (i.e., RFC 4463) from the Internet Engineering Task Force (IETF) that gained market acceptance in the mid-2000s and represented a paradigm shift for telephony application developers using speech technologies. Instead of using vendor-specific commands to generate TTS or to perform ASR, MRCP commands are sent from the client to the server application. Also, instead of loading pre-recorded audio files locally, commands are sent to an MRCP server containing a URL of the file to be played to the caller. These commands generate an audio stream sent to the client application, which should then direct it to the caller.

MRCP uses a similar style of clear-text signalling as in HTTP, in which each message contains three sections: a first-line, a header, and a body. Like HTTP, MRCP uses a request (usually issued by the client) and response model. Responses can simply acknowledge receipt of the request or give other information regarding its processing. For example, an MRCP client may request to send some audio data for processing (say, for speech recognition), to which the server could respond with a message containing a suitable port number to send the data. Since MRCP does not have support for audio data specifically, this would be handled by some other protocol, such as Real-time Transport Protocol (RTP) or the Real Time Streaming Protocol (RTSP).

There are two versions of the MRCP protocol. Version 1 uses RTSP (discussed further below); Version 2 uses SIP as the control protocol.

Streaming Media Servers

Another important IETF protocol for interactive response, particularly as multimedia (video) gets applied to the customer self-service model, is the Real Time Streaming Protocol (RTSP). RTSP is a protocol (covered by IETF's RFC 2326) that is commonly used for streaming media from a media server to a client endpoint. RTSP is an application-level protocol for control over the delivery of data with real-time properties. RTSP provides an extensible framework to enable controlled, on-demand delivery of real-time data, such as audio and video. Sources of data can include both live data feeds and stored clips.

With the transmission of streaming content, the multimedia data payload itself is not a task of the RTSP protocol. Many RTSP servers use the Real-time Transport Protocol (RTP) for media stream delivery; however, some vendors have implemented proprietary transport protocols.

Two components that are implemented to take advantage of RTSP are a client and a server. An RTSP client has the ability to control the media stream offered by the RTSP server with a series of VCR-type commands like play, pause, fast forward, and rewind. In addition to providing media to RTSP clients, RTSP servers maintain network status information which is used to provide the best possible user experience. The RTSP client is typically implemented in a media server or application server, and the RTSP server is generally a standalone server device.

Interactive Response - Expanded Deployment Options

The availability of mature and flexible protocols and runtime environments, along with software-based multimedia processing that can deliver interactive response services and solutions, has opened up a wide range of deployment options to meet a similarly wide range of budgets, applications, and customer audience – whether the audience is narrow and well-defined or virtually global in reach. The following section details some of the expanded deployment options that are available.

Interactive Response and Distributed Architectures

IP-based architectures, networks and protocols have made it easier to deploy interactive response solutions and services in a more distributed fashion. High volume, high density service provider deployments in particular stand to benefit from a distributed architecture. Interactive Response systems are often decomposed into the following server and network components:

- Media Server
- Application Server
- Media Gateway or Session Border Controller
- Load Balancer

Decomposing what was once a "black box" approach to Interactive Response systems can bring about numerous benefits, including:

- No single point of failure
- Easier scalability – one to many, many to one
- High reliability
- Centralized resources
- Lower costs
- Location independence

Benefits of decomposed / distributed architectures include off-loading intensive and time-critical media processing. Dedicated IP media servers are designed to cope with large numbers of media processing channels. With the intensive media processing housed in dedicated servers, the application developer does not need to worry about the effect this might have on other processes.

Improved scalability can result from the ability to access multiple IP media servers per application server. By adding multiple IP media servers, an application can be scaled to cope with very large numbers of media processing channels. This typically does not have a significant effect on the application server. Conversely, a single IP media server can be shared by multiple application servers. In cases where the amount of media processing is small per application, a single IP media server can be shared between multiple application servers. Either way, the processing load generally is predictable and reliable.

An example of an application where a single media server can be shared is one where the application server is primarily routing calls, for least-cost or prepaid routing. In this case, the media server is deployed infrequently, to give announcements to the callers ("out of credit", "unavailable", etc.). Also, the IVR system may be used infrequently to update payments, change settings, etc. In these types of applications, the main application servers are routing thousands of calls each and they share a small pool of media resources. This is a microcosm of the IMS model, in which there are far fewer media servers than SIP endpoints.

A distributed architecture eliminates single points of failure, and enables redundancy and failover. Because media servers are distributed on a network, various redundancy models can be used to manage groups of media servers. This yields the potential for high availability and the reliability that is desired when providing services.

Geographic and topological placement of service infrastructure is also enabled, which can improve the response of the application. It may be necessary in some networks to have media servers located physically or topologically close to the endpoints accessing the system so that the audio or video payload can be carried more efficiently or more reliably to the user.

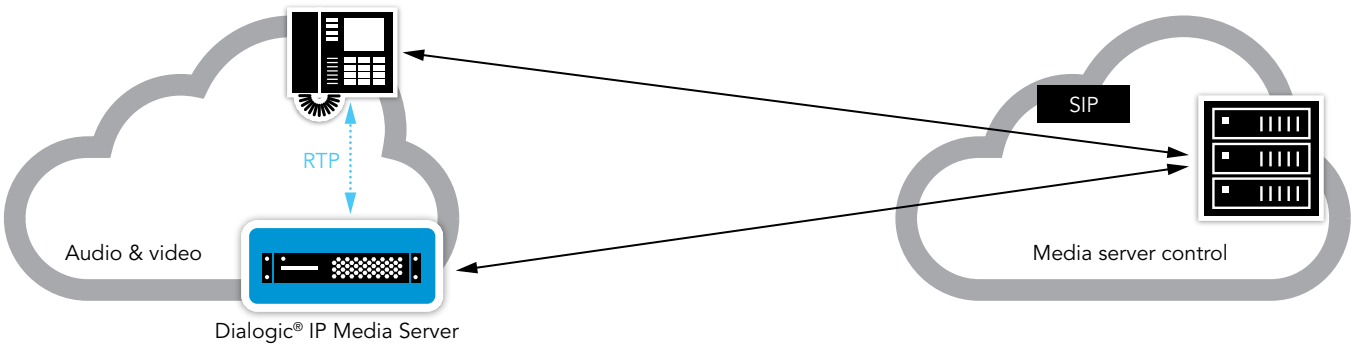

Figure 6: Media server deployed close to the receiving endpoint to improve network efficiency.

Those familiar with networking will appreciate that Figure 6 is a very simplified depiction of network topology. The broad point is that signalling and control information, being of far less volume than media and not "real-time" critical, can be carried over long distances or between networks, where real time voice and video may need to have fewer hops or better managed delays. Depicted in the network diagram is an example of a situation where a service can be managed by a central cluster of application servers with strategically deployed media servers, which can be intelligently located, either physically or logically in a large network, benefitting network traffic.

The distributed architecture also enables a logical independence from signaling. Control of media servers can be independent of the application servers that set up calls. From a business model perspective, media servers may be owned or leased by different organizations, such as a media server within the IMS architecture or a Platform as a Service (PaaS) provider owning the media servers to which independent application server owners can subscribe on a usage-based model.

There are many other potential applications (e.g., messaging, conferencing services, transcription services, and other third-party solutions) and it is possible that the application, or even the business, that controls media delivery or dialogues may be far removed from the application or business that sets up the initial SIP call. In those cases, the service typically will be provisioned using a web services API. In other words, once a user session is in progress, responsibility for driving the media content of the call may be delegated to an entirely different piece of software or even an entirely different organization.

Interactive Response and Virtualization

Regardless of where a data center running an Interactive Response solution or service is located, server virtualization can add an increased level of deployment and operational efficiency to the solution set. With the advent of robust software-based media processing, Interactive Response and virtually all voice applications have become software applications on an IP network running on standard datacenter server infrastructure. The ability to virtualize the application, media server and other server roles in the solution set can lower the total cost of ownership by reducing the number of physical servers. Deploying fewer servers also can lead to savings on electricity and cooling, as well as on rack space in the equipment room, wiring closet, or data center. In addition to saving money, using less equipment and power also is in keeping with green initiatives and goals for protecting the environment.

Interactive Response and SIP Trunking

Is the PSTN going to be retired? There are many theories and predictions about the longevity and continued usefulness of the PSTN; however, the acceleration of VoIP across public and managed broadband networks is undeniable, and holds with it the potential for more efficient customer self-service and interactive response system deployment, as well as the ability to clear the way for interactive multimedia sessions. SIP trunking services are being adopted at a rapid pace in key markets around the globe unencumbered by regulations restricting the use of VoIP, such as North America, Western Europe and Australia.

For interactive response solution providers, there are three notable benefits of architecting and deploying SIP trunk ready systems:

1) **Cost:** Both CAPEX and OPEX can be addressed by connecting an interactive response solution to SIP trunks instead of the PSTN.
   - Session Border Controllers (SBCs) vs. Gateways – The cost per session for an SBC is typically less than that of a media gateway of comparable density (depending on the vendor). This in turn can reduce the CAPEX of the solution.
   - SIP Trunk vs. PSTN – In many markets, a significant service cost savings can be made possible by using SIP trunks, with savings being realized often on both fixed monthly charges and per minute costs. This in turn can reduce the OPEX of the solution.

2) **Multimedia Sessions:** SIP trunking can enable a full continuum of scalable unified communications interactions, from simple presence through to full multimedia collaboration sessions.

3) **New Business Models:** Even with customer premise-based solutions, it is possible for a solution provider to expand its offering and revenue opportunity to include SIP service provisioning and billing, and thereby position itself to take more ownership of the recurring revenue generated by the interactive response solution. While entering the internet service provider business may seem like a daunting proposition, the barriers to entry can be significantly lower than becoming a PSTN CLEC, or 'private label'. SIP Trunking services are available for branding and reselling.

Platforms for Next Generation Interactive Response

To facilitate the development and deployment of next generation interactive response solutions and services, platform technology leaders such as Dialogic have developed advanced media processing and signaling platforms which can accelerate time to market, enable innovation, help optimize deployment models, and support competitive business models.

The primary platforms from Dialogic that are available to solution developers and service providers are:

- **Dialogic® PowerMedia® Products** – Dialogic® PowerMedia® Host Media Processing (HMP) Software and Dialogic® PowerMedia® Extended Media Server (XMS) products provide many core functions for automatically processing and improving IP-based business communications solutions such as next generation interactive response. These functions include: tone generation and processing, voice and video play and record (including compressed narrowband), toll quality and high definition, voice and video conference, echo cancellation, fax image coding and decoding. PowerMedia products perform media processing tasks on general-purpose servers without requiring the use of specialized hardware, and application program interfaces and software development kits are available from Dialogic.
- **Dialogic® BorderNet™ Session Border Controllers (SBCs)** – peering SBCs provide security, interoperability and reliability for SIP based multimedia communications wherever two disparate IP networks meet. Enterprise SBCs typically reside at the edge of an organizations private network to connect with public IP networks and SIP service providers for basic trunking and cloud based business communications services. Dialogic offers both of these types of session border controllers within the BorderNet SBC product family. And for mobile and fixed VoIP networks, Dialogic provides a highly efficient SBCs for robust security and session management.
- **Dialogic® Integrated Media Gateways (IMGs)** are carrier-ready VoIP gateways that supports both media and signaling in a single chassis. They allow service providers to add new telephony services quickly, and give them a clear migration path to an all-IP network.
- **Dialogic® Distributed Signaling Interface (DSI)** Products and Software – Dialogic's DSI Products and Software provide interfaces and protocols used in mobile text messaging solutions and high density PSTN network connectivity and control. Application program interfaces, software development kits and Web 2.0 interfaces are available from Dialogic. The different types of DSI Products and Software available from Dialogic include:
  - **Dialogic® Distributed Signaling Interface (DSI) Boards** – these enable distributed execution of SS7 message processing across board processors, application host CPUs, or separate signaling servers.
  - **Dialogic® Distributed Signaling Interface (DSI) Signaling Servers** – these enable affordable, high-performance, distributed, cross-platform, cross-operating system signaling applications.
  - **Dialogic® Distributed Signaling Interface (DSI) Signaling Software** – this supports a wide range of Signaling System 7 (SS7) and internet protocol capabilities to manage the signaling traffic between an application and the SS7 network over TDM or IP.

Summary

The breadth and depth of customer service applications and customer service needs on a global level, and the rapid expansion of communication modes, channels and networks, have spawned a wide array of Interactive Response technologies and methodologies. Important trends for Interactive Response solution and service providers to consider include:

- The rapid rise of end-to-end communications sessions over IP networks
- The widespread adoption of smartphones, tablets and associated mobile applications
- The availability of service creation environments for rapid service and solution development
- The increasing use of distributed architectures and standard signaling and control protocols
- The increasing power, flexibility and economy that are delivered by mature multimedia processing software
- The strong demand forecast for cloud-based 'X' as a Service models for Interactive Response
- The popularity of social networking and related Web techniques for integrating service solutions

Opportunities to innovate in customer service and Interactive Response abound with these market and technology trends. As a leading provider of enabling technology for interactive response for almost three decades, Dialogic believes it is uniquely positioned to provide the latest technologies to enable the next generation of Interactive Response solutions and services, as well as to bring to customers a depth of technical and business experience to help them make intelligent choices between the many options available to bring a new service or solution to market.

For More Information:

For more information on the Dialogic Products listed in this document, visit www.dialogic.com.

www.dialogic.com

For a list of Dialogic locations and offices, please visit: https://www.dialogic.com/contact.aspx

10/17 13162-02