

Next Generation Call Centers

Michael Chapman

Director of Product Management
Cantata Technology



cantataTM
TECHNOLOGY

table of contents

Introduction	3
Call Center Features	3
New Features	4
Next Generation Call Centers	5
Transitional Call Centers	7
Technology and Time-To-Market	7
Summary	8
About Cantata	9

Introduction

In today's call center environments, IT managers are confronted with many challenges ranging from optimizing call agent productivity and call center efficiencies to controlling overall operational costs. In most cases, call centers are geographically dispersed multi-site operations which often include remote agents. These are commonly referred to as virtual call centers. Another approach gaining momentum is the hosted call center which involves a service provider delivering both the multi-site connectivity along with the call center processing systems. However, all approaches whether they be centralized, distributed, or hosted require seamless customer facing interaction. In addition, capabilities for multimedia communication supporting a wide range of services is critically important as call centers evolve into multi-service contact centers. This whitepaper takes a deep look into the requirements, functions, and technical methodology of how call centers can be deployed to optimize resources in delivering high-value services.

Call Center Requirements

Call centers have unique requirements for media processing. Scenarios include the following.

- *IVR*
- *Transfer to Agent*
- *Consultation Transfer*
- *Multi-Party Session*
- *Coach-Agent-Customer*
- *New Features*

End User Goals

- Improve Customer Relations
- Reduce Agent Costs
- Reduce Telecommunication Costs

Call Center Features

IVR

Call center applications use interactive voice response (IVR) for customer interaction. IVR systems help optimize agent usage. The best reduction in agent costs is by enabling the customer to do self-service that removes the human agent from the transaction. Short of eliminating the agent entirely, the IVR system collects common information from the customer. This reduces the amount of time the agent needs to spend with the customer. Moreover, with the collected information, as well as identifying information from the customer, the call center system can select the agent with the best skills and availability, and then connect the customer.

Legacy IVR systems required the developer to manage IVR resources, such as prompt players, tone (DTMF) detectors, recording devices, and so on. This required the developer to write many lines of code for resource management. Moreover, the application call flow is at the level of play prompt, collect digits, validate digits, if wrong, play another prompt and collect more digits. This requires many lines of code to manage resources, implement the call flow, and validate each step of the caller interaction.

Transfer to Agent

As described above, an IVR system may transfer a customer to an agent. In addition, agents may transfer a customer to another agent, or back to an IVR system.

Consultation Transfer

The agent may wish to consult with the target of the transfer before connecting the customer to the second agent. During that time, the customer can hear digital silence, music on hold, or, with the capability of next generation systems, content customized for the customer.

Multi-Party Session

The agent may wish to bring the customer into a discussion with more than one agent. For example, the agent may wish to stay in the conversation with the customer and a manager.

Coach-Agent-Customer

This is one of the more challenging media mixing topologies. A common scenario is where the agent is speaking with the customer. The agent's coach is listening to the conversation and "whispers in the ear" of the agent. This way the coach can direct the conversation without the customer hearing the side conversation.

Sub-Conference

Although not often seen in the call center, this media mixing topology has two or more people speaking in a main conference, with two or more people in a private conference, where they can hear the main conference in the background. The people in the main conference cannot hear the people speaking in the subconference.

New Features

Video

IVR and self-service are popular modes of interacting in business-to-consumer and business-to-business situations in most areas of the world. However, some markets, such as finance in the Far East, demand more personal touch. Video calling, and Video interactive response, can address this cultural inclination.

Another use for video is for technical support applications. For example, a support technician in a call center can ask the customer to show them what is not working, rather than asking the customer to describe the problem. This can significantly reduce hold times and miscommunications.

Multi-Site, Global Call Centers

For some time we have had network-based call distribution systems that direct calls to multiple, remote sites. However, management, monitoring, and control of the call has been lacking. This is particularly acute for overseas call centers. Currently, one transfers a call to the remote call center, and one is at the mercy of the controls and capabilities of that center. Being able to monitor hold times, transfer activity, interaction times, and so on would give a significant improvement to the customer experience and satisfaction. An opportunity afforded to us by leveraging IP technologies is the ability to provide for on-demand capacity with minimal capital investment.

Call Recording in an IP Environment

The IP environment introduces challenges and opportunities compared to traditional call center environments. For example, the shared nature of IP infrastructure makes it relatively easy to snoop on a conversation. Likewise, there are well known weaknesses of the TDM infrastructure when it comes to signaling and media security. Thus many see a migration to IP as an opportunity to deploy true end-to-end encryption of signaling and media. Moreover, end-to-end authentication, a component of end-to-end encryption, enables customers to know they really are communicating with the business and enables business to know they are really communicating with one of their customers.

This opportunity for secure communications presents a challenge. For many reasons, call centers may wish to hide the actual agent endpoint identity. For example, the agent could be a work-at-home employee, using shared infrastructure for their call center and personal activities. Likewise, in the coach-agent-customer scenario outlined above, the call center does not want the customer to know the coach is on the call (other than the legal requirements in many jurisdictions mandating a statement that the call center may monitor or record the interaction).

Approaches for call recording include using proprietary clients that record the (endpoint-decrypted) media, packet-sniffing (mandatory unencrypted) media streams, or terminating the streams at a centralized device (where the user-facing streams are encrypted and call center-facing streams are unencrypted).

Next Generation Call Centers

Architecture

Next generation call centers use a three-tiered architecture similar to the IMS. Namely, they leverage application servers, data base servers, and media servers. Access from the legacy telephone network is through media gateways. By leveraging the IMS architecture call center application developers can leverage the rich tools available for creating and deploying their valuable services. Likewise, the enterprise gets deployment flexibility by being able to host the entire communications infrastructure on premises, use hosted services from a service provider, leverage SIP and IP control to keep proprietary customer data and call flows on premises and using service provider media resources, or a multi-site, combination of the above.

The media gateway converts TDM media and signaling such as ISDN PRI, SS7, or in-band signaling, to SIP signaling and RTP media.

The media server is a central resource. It is an optimized user interaction device. The media server interacts with the user by playing prompts, collecting digits, recording speech, performing automatic speech recognition (ASR), and performing text-to-speech (TTS) conversion. It also enables multi-user interactions by mixing streams with simple three-way or n-way conferences or more complex topologies such as sidebar, coach-whisper, and buddies. Media servers expose these services by using RFC 4240 (netann), VoiceXML, and RFC 4722 (MSCML). Media servers are stateless, shared resources. Application server make interaction requests of the media server, at which point the media server presents the information to the user, requests information from the user, and sends that information to the application server.

The application server responds to stimulus, either from an inbound call (SIP), the result of a user interaction (HTTP), or internally (i.e., to originate a call, as in an outbound call center). Application servers often use web technology, such as web application servers such as BEA WebLogic, IBM WebSphere, Ubiquity SIP, or Oracle Application Server as a platform for the call center application.

IVR

Previous papers from Cantata discuss the applications and services infrastructure. In particular, the focus on the number of lines of code and code complexity discussion highlights the benefits of using VoiceXML to create interactive voice response applications. Moreover, Cantata, in conjunction with other VoiceXML providers, has standardized VoiceXML extensions for creating interactive video applications.

VoiceXML 1.0 did support rudimentary call control, enabling applications to transfer a call to a destination. However, the computer science principles behind the benefits of using VoiceXML for describing user interaction makes VoiceXML wholly unsuitable for directing call control. It is for this reason that VoiceXML 2.0 deprecated the <transfer> tag. The main reason that call control does not belong in the media server is that call control is, by its nature, statefull application logic. Application servers, particularly in the call center environment must know detailed status of calls. This is particularly so for outbound call centers, where applications take different paths based on, for example, network busy versus line busy, whether the dialed number has an intercept (and which intercept triggered the call failure), whether an answering machine answered the call, and so on.

In the Next Generation call center, the application server uses third-party call control (3PCC) to analyze what to do with inbound calls, setup user interaction sessions with a media server, initiate outbound calls, and instruct the media server to bridge, mix, or otherwise treat calls. If the application server decides to start an IVR session, it issues a RFC 4240 (netann) request to the media server with the address of a VoiceXML script. The VoiceXML script describes the user interaction, such as prompts to play, information to collect (in the model of forms), and even sophisticated interactions such as tapered prompts. Tapered prompts are where the first time the media server prompts the user, the media server plays brief prompts. If the user has trouble entering the requested data, the media server plays more informative prompts. This cycle can continue for any number of iterations, depending on the needs of the application and the imagination of the user interface designer.

VoiceXML returns the collected information using a HTTP page fetch request. This is one of the reasons many application developers create their application on a SIP-enabled web application server. There are many common techniques for correlating the HTTP requests with the SIP session, such as tagging the HTTP URI, storing cookies, and so on.

Note that it is easy to integrate external, dynamic data into a VoiceXML script, by making that data available over an HTTP fetch. The format of the data need not be VoiceXML; it can be plain text or speech synthesis markup language (SSML), as well. Obviously, recorded data is the easiest to play to the user. The media server simply treats audio or video data in the same way it handles a prompt.

Blind Transfer to Agent

When an application decides to transfer a call to an agent, the application server has a few methods for signaling the call transfer, depending on the requirements of the call center. The application server can use 3PCC by removing the media server from the call with a BYE, issuing an INVITE to the agent's device, and proxies the agent's SDP to the customer (most often the media gateway). This has the benefit of the application staying aware of the signaling of the call. The importance of this is the application server is a single point for collecting whole-call statistics, such as queue times, IVR times, agent times, etc. In addition, this method frees media server ports, increasing the call volume the media server can potentially handle.

The application server can keep the media server in the call if desired. The usefulness of this is that it enables the media server to record the conversation for compliance purposes, enables the application to interact with the customer or the agent, and enables to application to bring in additional agents or resources. The application server keeps the media server in the call by using 3PCC towards the agent, but this time not dropping the media server from the call. If the customer is in an IVR session, the application server creates a RFC 4722 enhanced conference, brings the customer to the conference, and then proxies the agent's SDP to the conference. The application can record the conversation by requesting the media server record the conference.

Consultation Transfer

Typically, when the application server transfers the call, it may play music on hold, advertisements, information, and so on. As described above, the application server uses 3PCC to bring the agent to the media server. However, rather than just connecting the two parties, the application server requests the media server to interact with the agent. After the agent interaction, the media server brings the two call legs (customer and agent) together. As before, the media server can record the conversation for compliance purposes using RFC 4722.

Multi-Party Session

The application server can bring in or out as many parties as it desires. If the application server does not need call recording or special topologies, it can simply use RFC 4240 and invite all of the participants to the same conference URI. Conversely, for the application server to request call recording or special topologies (as described below), it uses RFC 4240 to set up an RFC 4722 enhanced conference.

Coach-Agent-Customer

This is a very common scenario for both inbound and outbound call centers. In the Coach-Agent-Customer scenario, while the agent speaks with the customer, the coach listens to their conversation. When the coach speaks, however, they "whisper-in-the-ear" of the agent and the customer cannot hear the coach, yet the agent hears both the coach and the customer at the same time.

Traditional methods of creating this topology, such as H.248 and MGCP derivatives, require the application server to plumb each direction of the RTP stream individually. While it is true that this enables any possible topology, it is at the expense of many lines of code. Moreover, it reduces or removes the ability of the media server to optimize the mixing algorithm. Finally, this is a layer violation, as the application must reach into the media plane to create the media plumbing. RFC 4722 provides a modern method of creating this, and other interesting topologies. Using RFC 4722, the application server identifies buddy lists. The media server then honors the application level requests of enabling each buddy to hear only their conversation. In the Coach-Agent-Customer scenario, for example, the Agent and Customer are in the conference, while the Coach and Agent are buddies. By default, everyone hears the conference, but the Coach's voice is not in the mix. Since the Coach and Agent are buddies, the Agent can hear the Coach.

Multi-Site Support

By using an IMS-like architecture, we can leverage the fact the application and media servers are not an integrated unit. A single application server cluster can control remote media servers. For a multi-site installation, one can put media servers on premises yet have the benefit of centralized application management. A more interesting possibility is to have a federation of application server and media servers at each call center, with a cluster of application routers in the network. The application routers are light application servers that use 3PCC (and possibly media servers for IVR) to route calls to the right call center. Because the application router is in the signaling path for the duration of the call, it is a natural point for collecting network-wide call statistics.

Other Capabilities

Next generation call centers, often referred to as contact centers, use presence, IM, e-mail routing, and so on. However, as this paper deals primarily with the media processing aspects of the call center, we will not touch upon them here.

Transitional Call Centers

Not all call centers are fully IP enabled. Moreover, many inbound call centers find number termination to be less expensive with TDM delivery, rather than SIP delivery. Moreover, some outbound call centers require low-level signaling manipulation and monitoring that the enterprise's VoIP service provider may not offer. In this situation, we suggest deploying SIP media gateways to bring the TDM traffic into the SIP realm. A media gateway enables the call center to connect with more than just simple ANI and DN spills. For example, one should look for an integrated media gateway that has full ISDN PRI and, for large call centers, SS7 support.

Some call centers require low level SS7 manipulation. For example, in order to reduce unnecessary resource consumption, a campaign-oriented call center may only allow a caller a set number of calls to the call center. On the first call, the caller gets the usual campaign treatment. This uses the full resources of the call center, such as IVR and possibly agents. On the second call, the caller gets a polite announcement thanking them for their participation in the campaign, but they already had their chance. This uses just a media server announcement resource. However, on the third and subsequent call attempt, the call center rejects the call at the SS7 lever. That is, before the use of any media resources or fees for call termination. For such an application, the call center deploys a SS7 signaling gateway. Note this again shows the benefit of separating the signaling logic from the media server. If TDM, and in particular, TDM signaling, terminates at the media server, it would be very difficult to deploy the above-mentioned application. That said, for legacy transition applications, one can deploy a multi-services platform that does the SS7 gateway, ISDN association, and bearer channel media processing all in an integrated platform.

Technology and Time-to-Market

In general, one can hack a solution together for any problem. However, time and time again, methodologies based on computer science and computer engineering principles are proven to result in much lower total cost of creating applications and systems. Often, such methodologies take a long time for acceptance, as they have a high startup cost. However, when a technology has a lower initial cost, it is straightforward to use it. Two computer science principles to consider here are programmer productivity and program complexity.

Over the past thirty years, practitioners and researchers in computer science and engineering have been measuring programmer productivity. Here, by programmer productivity we mean the number of lines of code delivered per day. By delivered, we consider the number of lines of code the programmer writes and debugs. If a programmer writes 2,000 lines of code in a weekend, but takes 90 days to debug it, we say they deliver 22 lines of code per day. An interesting result is programmer productivity has, for the most part, not changed over the past twenty years. Programmers deliver the same number of lines of code whether they write in Java, C++, COBOL, or assembly language. This result is why we use high-level languages. If a programmer delivers 40 lines of code per day on average, we really want those 40 lines of code to count. One can simply get more accomplished with 40 lines of Java than 40 lines of assembly language.

The relevance of this to the call center is evident when we look at ways of interacting with rich media resources. Some interfaces expose resources at a low level, offering total flexibility to the application developer. Examples of such interfaces are H.248 and MSML. The price of this flexibility is the developer has

to write code to manage the resources, manage connections between resources, media sources, and media sinks. Said differently, one must deliver more lines of code to do the same thing as with high-level interfaces such as VoiceXML and RFC 4722 (MSCML). In practice, the low-level interfaces require an order of magnitude more code than high-level.

One argument for the low-level interfaces is they enable total control. When SnowShore invented the high-level, SIP-controlled interface in 2001, the theory was that 85% of all applications could benefit from the high-level interface, and the rest could use the well-defined H.248 interface. However, the reality is that to date, an application that requires such a low-level interface does not exist. Another argument for low-level interfaces is they allow developers to use low-level capabilities in the future. However, by that logic one should write one's applications and systems in assembly language, because there is the possibility that one needs more control than that afforded by C++, C, or Java. This clearly does not happen, as the benefits of time-to-market and project cost swamp the theoretical benefit of coding in assembly language.

Summary

Call centers are very demanding environments requiring powerful functional capabilities to ensure a high level of agent productivity and operational efficiency. The economic and technical approach to the design and deployment of network resources and applications are important factors in achieving these objectives. In this paper, we've examined a number of key functional and network requirements, and discussed the critical technical considerations and proven methods of designing and deploying call center applications. The following sections provide a brief overview of Cantata's products which provide the building blocks for highly scalable and reliable call center services.

Cantata Products

SnowShore Media Server

The SnowShore Media Server by Cantata is a highly integrated SIP media server. Created by the pioneers of RFC 4240 and RFC 4277, the SnowShore Media Server is the only media server that offers IETF-published interfaces. In addition to the benefits of SIP, VoiceXML, and MSCML as described above, there is a wealth of application partners delivering solutions for call center, messaging, conferencing, pre-paid, gaming, and other value-added services. Moreover, the SnowShore Media Server is field proven with deployments worldwide and with multiple applications, proving the robustness of the interfaces and management capabilities of the platform.

The SnowShore Media Server is available on standard, open commercial platforms. We have the highest density media processing platform commercially available today. Cantata holds the number one market share for media server revenues according to Infonetics Research. In addition, depending on the quarter, in 2006 Cantata holds either the number one or two market share position in terms of ports shipped for media servers.

IMG 1010

The IMG 1010 combines signaling gateway and media gateway functions to give operators the critical "any-to-any" interworking functions for connecting and delivering services across legacy and IP fixed and wireless networks. The Cantata IMG 1010 is a highly integrated media gateway providing integrated SS7, ISDN, other TDM signaling technologies and SIP.

MSP 1010

The MSP 1010 is a modular, scalable and cost-effective media resource and signaling platform, designed for deployment in legacy, convergent, fixed or mobile networks as a classic IN/AIN service node or service switching point. The Cantata MSP 1010 is a programmable SS7 gateway ideal for roaming applications and lawful intercept, and also supports SIGTRAN, MAP, and SIP.

Cantata Standards Leadership

IETF

Cantata personnel wrote RFC 4240 (netann), RFC 4722 (MSCML), RFC 4730 (KPML), RFC 4483, RFC 4117, RFC 3752, RFC 3259, RFC 3458, RFC 3302, RFC 3250, RFC 2306, RFC 2302, RFC 2301. Cantata personnel have held or currently hold work group chair positions in mediactrl (application server – media server protocol), speechsc (media server – speech server protocol), an ifax (Internet Fax, which produced T.37 and T.38).

W3C

Cantata personnel contributed to VoiceXML 2.0, VoiceXML 2.1, and CCXML 1.0. Moreover, Cantata personnel created the Call Control Sub-Workgroup of the Voice Browser Work Group in the W3C.

3GPP

Cantata personnel contribute to CT1 and CT4, focusing on the MRF interfaces.

SIP Forum

Cantata personnel serve on the Board of Directors of the SIP Forum. The SIP Forum is working on many enterprise and call center issues such as device configuration and enterprise-carrier interconnect.

About Cantata

Cantata Technology, established in 2006 through the combination of Brooktrout Technology and Excel Switching Corporation, provides enabling communications hardware and software that empowers the creation and delivery of anytime, anywhere IP-based communications applications. Leveraging more than 20 years of experience, Cantata offers the broadest range of products, along with a worldwide network of partners that allows service provider and enterprise customers to develop new products, introduce new services and cost-effectively transition networks to IP. Headquartered in Needham, Mass., Cantata maintains multiple locations worldwide in North America, Asia and Europe.



cantata[™]
TECHNOLOGY

Corporate Headquarters

Needham, MA 02494

USA

Tel: +1 (781) 449-4100

Fax: +1 (781) 449-9009

Email: info@cantata.com

Cantata Technology maintains multiple locations worldwide in North America, Asia and Europe.

www.cantata.com

Cantata, Cantata Technology, and the stylized logo with and without the term Cantata Technology are trademarks of EAS Group, Inc., the parent company of Cantata Technology, Inc., or its subsidiaries.. All other products and services mentioned are the property of their respective owners.