Web-enabled Voice over IP Call Center
An Open Source Based Implementation

Stefan Kuhlins[1] and Didier Gutacker[2]

[1] University of Mannheim
Department of Information Systems III
D–68131 Mannheim
Germany
stefan@kuhlins.de
http://www.wifo.uni-mannheim.de/~kuhlins/

[2] OSI mbH
Rudolf-Diesel-Straße 3
D–55286 Wörrstadt
Germany
gutacker@osi-online.de
http://www.osi-online.de/

Abstract. This paper describes the concept and open source based implementation for combining Internet commerce solutions and World-Wide Web access to information (e.g., Internet and telephone banking) with a voice call button that allows immediate access to a call center agent from any PC over a single network connection via *voice over IP*. A special strength of this integrated solution is that it is context sensitive. Hence, it allows a kind of collaborative browsing where customer and agent work synchronously with the same document. We call this a "Web-enabled voice over IP call center."

1 Motivation

Internet transactions cost about one-tenth as much as live telephone transactions, therefore the economic benefit of a self-service Web site is apparent. Unfortunately, self-service is not satisfying to all customers; there will always be those who prefer to speak with natural persons such as call center agents.

Imagine the following scenario: a bank customer uses Internet banking and looks for offers on his bank's Web site. He finds a very interesting offer, but now he needs further information. In order to get answers to his questions as soon as possible he makes a call to his bank's call center. There he must authenticate himself again and it is necessary to tell the call center agent about the offer in question.

[*] Accepted for IEEE International Conference on Networking 2001 (ICN'01), July 11–13, Colmar, France, http://iutsun1.colmar.uha.fr/ICN01.html

In view of this scenario, the main benefit of a Web-enabled VoIP call center is that a customer only needs to click a button instead of making an ordinary call to his bank's call center. After clicking the button, a connection to the call center is established using *voice over Internet protocol* (VoIP). At the same time, the customer's data is extracted from a database on the server and the Web page loaded by the customer is determined. This information is automatically presented on the call center agent's terminal to help the agent assist the customer. Thus, questions can be answered quickly. As a consequence, customer and agent spend less time on things having nothing to do with bank services. Moreover, transactions may be done, which are not intended for the Web.

The most important benefits of a Web-enabled VoIP call center are:

- Customers get context sensitive support, e.g., for filling out forms or searching for specific content.
- Customers are not forced to repeat input, which they have already made in the Web, because it is automatically provided to the call center agent.
- Usually, calls by VoIP reduce telephone costs compared to phone calls via the public switched telephone network (PSTN). That applies not only to customers, but also to the call center, because it saves long distance toll charges that are incurred with an *800 number* [3]. Plus, there are no charges for hold time on the Internet.
- The combination of passive and active channels improves the quality of information altogether.
- If customers own only one line for surfing the Internet and making calls, they may do both at the same time by using VoIP. Therefore, VoIP removes the need for customers to disconnect their Internet connection to make a call [4].

The remainder of this paper is organized as follows: In Section 2 we specify the components of our solution. Section 3 details the implementation. Next, Section 4 gives directions for future work. In Section 5 we describe some limitations of VoIP in general. Finally, Section 6 concludes the paper with a summary.

2 Components

On principle, the required techniques for surfing and Internet telephony exist. Just an appropriate combination and integration is necessary to realize a Web-enabled VoIP call center.

The main goal of our project is to build a Web-enabled VoIP call center based on freely available software, preferably open source software. Such a solution should be suitable for small and mid-range companies, which are not able (or unwilling) to spend much money on a sophisticated system by one of the major players in the VoIP market.

On server's side, the following components are used (see Fig. 1):

- *Apache* Web server [1] for handling HTTP requests (Hypertext Transfer Protocol [19])

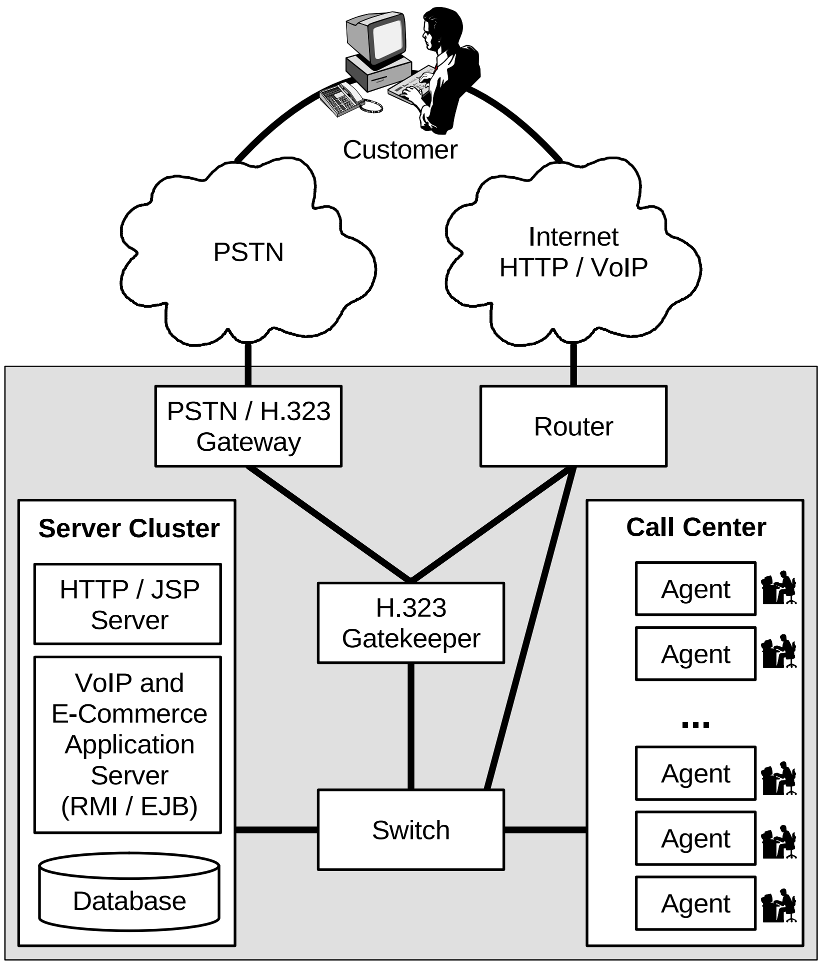


Fig. 1. Overview of a Web-enabled VoIP call center

- *Tomcat* [2] for handling *JavaServer Pages* [17]
- *Java* [14] for implementing the application server
- *OpenH323* [10] as telephony software in the call center

Tomcat is the reference implementation for JSP [2]. JSP separates the user interface from content generation enabling designers to change the overall page layout without altering the underlying dynamic content [17]. The *OpenH323* project aims to create an Open Source implementation of the ITU H.323 teleconferencing protocol that can be used by personal developers and commercial users without charge [10]. H.323 is a standard for real-time multimedia communications and conferencing over IP networks [5].

On customer's side, the requirements should be as small as possible. Thus, customers only need a multimedia PC, Internet access, a Web browser, and a H.323 compliant telephony software such as *OpenH323* [10] or *Microsoft's NetMeeting* [7], which is already part of newer versions of Windows.

While the communication with the customer's side has to deal with a myriad of possible software and hardware combinations, the equipment on the call center's side is well known. Hence, there are no obstacles to an installation of a comfortable environment. In addition to the requirements to the customer's side, the agents in the call center need a *Java Plug-in* [15] for the Web browser or the brand-new *Java Web Start* architecture [18], so applets or applications with a *Swing*-based graphical user interface [13] can be used.

A Web-enabled VoIP call center replaces a regular call center as well, which handles "normal" phone calls via the PSTN. Such calls are routed by a PSTN / H.323 gateway. A gateway translates between the traditional PSTN and packet-based data networks such as the Internet. A gatekeeper acts as the central point for all calls within its zone and provides call control services to registered endpoints. In the following, we will concentrate on VoIP calls and ignore this part concerning the PSTN.

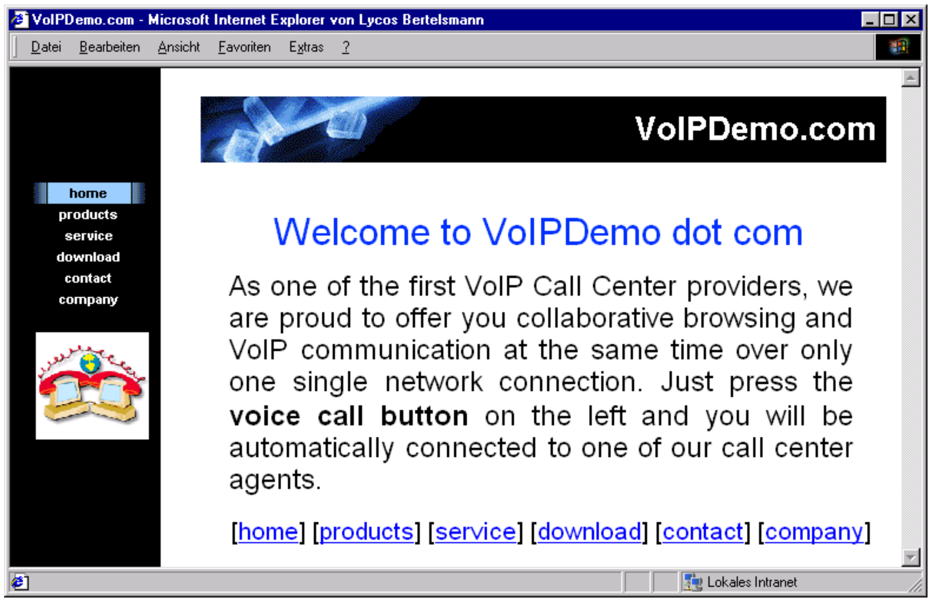
3 Implementation

The primary purpose of our prototype is to illustrate the core technological concepts. In the following, we describe a typical scenario and the implementation techniques used behind the scenes.

First of all, a customer visits the Web site by loading on of its pages with a Web browser. What he sees is an HTML page (see Fig. 2), but in fact, the Web server generated this page based on a *JavaServer Page* (JSP). This page includes a *JavaBean* that tracks the session and customer related data. Since HTTP is by design a stateless protocol, sessions administered by the Web server are required to relate different requests to the same customer.

Besides session tracking, the *JavaBean* records every URL (uniform resource locator) that the customer visits. Moreover, it stores the customer's IP address, which is part of every HTTP request.

A connection via VoIP should be established as comfortable as possible. Therefore, a voice call button is placed on every page. If the customer pushes the button, a VoIP connection between the call center and the customer has to be established. This is realized by a link to a special JSP, which includes the aforementioned *JavaBean*. The *JavaBean* connects via Java's *Remote Method Invocation* (RMI [16]) to a central server object. The server communicates with a Java applet that is part of an HTML page, which is loaded by all call center agents, because it is the login for agents (see Fig. 3). With the help of *JSObject* [9], a Java wrapper around a *JavaScript* object [8], the applet is in a position to control the call center agent's browser. This is used to open a simple dialog box that asks a selected agent, if he accepts the voice call request. In case he does,


VoIPDemo.com - Microsoft Internet Explorer von Lycos Bertelsmann
Datei Bearbeiten Ansicht Favoriten Extras ?
VoIPDemo.com
home
products
service
download
contact
company
Lokales Intranet

Welcome to VoIPDemo dot com

As one of the first VoIP Call Center providers, we are proud to offer you collaborative browsing and VoIP communication at the same time over only one single network connection. Just press the **voice call button** on the left and you will be automatically connected to one of our call center agents.

[home] [products] [service] [download] [contact] [company]

Fig. 2. HTML page with voice call button

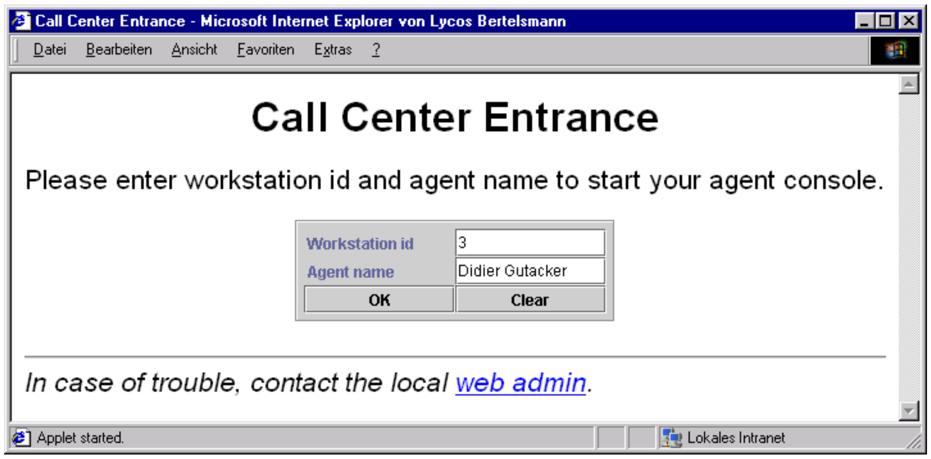
the applet starts the telephony software with the customer's IP address through a system call, which is checked by the security manager.

On the other side, the customer's telephony software has to be started unless already running. The easiest way to do this is to start it manually by the user before the voice call button is pushed. Automatic activation can be realized using a signed Java applet that starts the telephony software through a system call similar to the agent's side.

Another solution, which only works with *Microsoft's Internet Explorer* in cooperation with *NetMeeting*, is the *callto* protocol [6]. This allows links to start *NetMeeting* automatically (e.g., `<a href=callto:134.155.57.111+type=ip>`). Aside from its proprietary nature, the call would go from the customer to the call center, but we need it the other way round to avoid service malfunction and security problems. Hence, it is of no help in here, but nevertheless it is a nice feature.

If everything is fine so far, the customer is able to accept the call back of the agent and the telephony connection is established. In case no agent is available, this information is sent to the customer as an HTML page. The customer can retry later and "surf" the Internet in the meantime.

In parallel to set up the voice call connection, the server presents the customer's data to the agent (see Fig. 4). For that purpose, a Java application is started after login. If the customer has registered himself to the system his name is known. The customer's data includes visited URLs and the page that the customer is currently viewing (see Fig. 2). The voice call button of this page was


Call Center Entrance - Microsoft Internet Explorer von Lycos Bertelsmann
Datei Bearbeiten Ansicht Favoriten Extras ?

Call Center Entrance

Please enter workstation id and agent name to start your agent console.

Workstation id	3
Agent name	Didier Gutacker

In case of trouble, contact the local web admin.

Applet started.
Lokales Intranet

Fig. 3. Call center agent's login screen with applet

pushed. This page is also the starting point for collaborative browsing. The agent sees it in a separate browser window (similar to Fig. 2), which was automatically opened by the Java application.

If the customer clicks on a hyperlink, a request for the underlying page is sent to the Web server. As a result of this, the aforementioned *JavaBean* comes into play again. It informs the RMI server about the new URL. The RMI server sends the call center agent's applet the new URL, and the applet causes the browser to load this page via *JavaScript*. Now, customer and agent are both looking at the new page.

4 Future Work

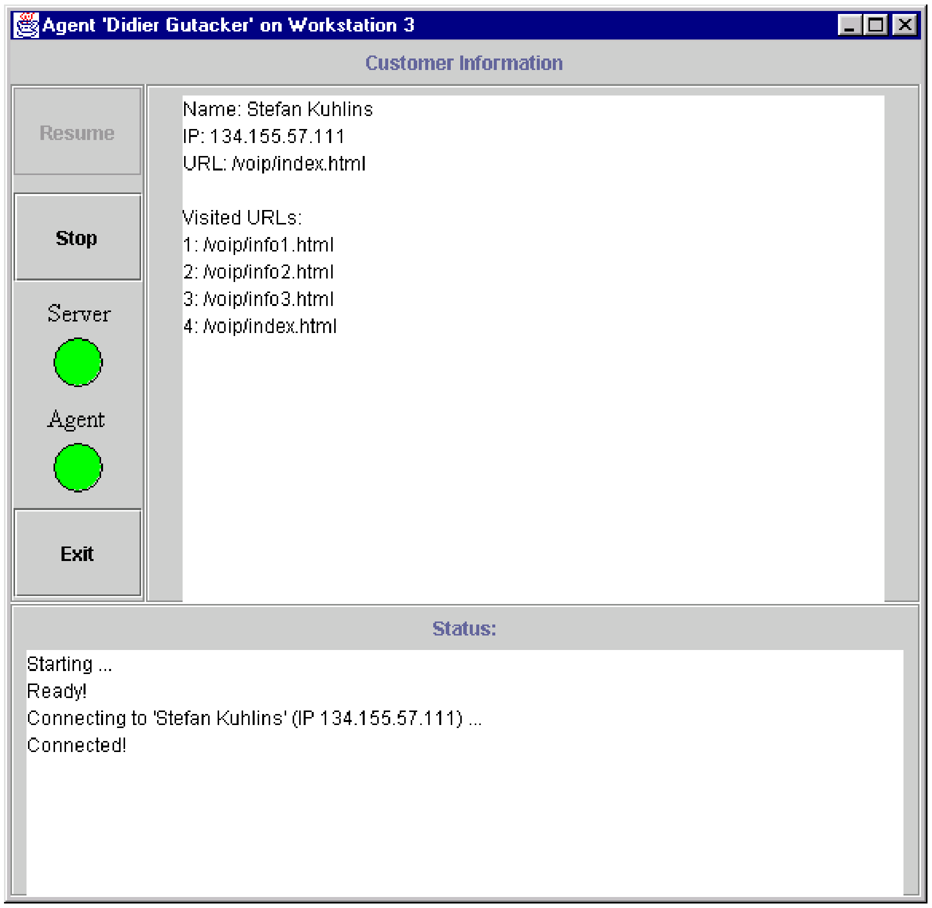
The goal of the first prototype is to provide clearer insights regarding the nature of VoIP and collaborative browsing. Hence, our initial "proof of concept" prototype works, but it is only a first step. There are many ways to improve the implementation.

If no agent accepts the call immediately, dynamic queue statistics such as position in queue and expected wait time should be provided.

For the present, the prototype supports only collaborative browsing in direction from the customer to the agent. The opposite direction can be implemented with similar techniques.

It should be possible that the agent fills in forms on behalf of the customer.

In support of a better scalability, an *Enterprise Java Beans Server* [12] should replace the RMI server. In result of this replacement, another weak point of the current prototype—the issue of persistence—would be eliminated; the central RMI server holds all data in memory, instead of using a database.



Name: Stefan Kuhlins
IP: 134.155.57.111
URL: /voip/index.html

Visited URLs:
1: /voip/info1.html
2: /voip/info2.html
3: /voip/info3.html
4: /voip/index.html

```
Starting ...
Ready!
Connecting to 'Stefan Kuhlins' (IP 134.155.57.111) ...
Connected!
```

Fig. 4. Java application with customer data

If an online shop uses the system, the customer's data presented to the agent should include the items of the "shopping cart." In this way, the agent can propose articles for cross selling.

On the customer's side, a seamless integration of telephony software into operating systems and browsers is desirable. In this case, customers are not forced to download and install telephony software themselves. Moreover, the telephony software should start automatically, similar to Microsoft's aforementioned *callto* protocol.

Provided that fast communications technologies will allow the addition of video to the connections, H.323 based video between agent and customer could be added.

5 Limitations

In the early days of IP telephony, most systems were half duplex, and all were incompatible with one another, let alone with conventional telephone systems. In the meantime, things have changed, but only when VoIP provides a level of quality that nearly equals the PSTN people will accept it.

At present, some restrictions on the customer's side concerning the technical equipment have to be considered, but in the near future, this should be no more obstacle. To achieve a suitable quality of service, customers must have an Internet connection with sufficient bandwidth. At least an ISDN line should be available. Additional equipment like a sound card, headset or speaker and microphone is needed. Today, especially headsets are not widely used.

6 Summary

Access to call center facilities via the Internet is a valuable adjunct to electronic commerce applications. Internet call center access enables a customer who has questions about a product being offered over the Internet to access customer service agents online. This combination of VoIP with point-of-service applications shows great promise for the longer term [11].

Our practical tests have shown that, apart from some minor restrictions concerning the comfort, solutions for making automated calls via VoIP and collaborative browsing are possible. Our implementation is only a first step towards a "Web-enabled voice over IP call center." More work has to be done for a portable solution with maximum comfort and speech quality.

References

1. Apache Software Foundation: Apache Web Server, `http://httpd.apache.org/`
2. Apache Software Foundation: The Jakarta Project – Tomcat, `http://jakarta.apache.org/tomcat/index.html`
3. Dialogic Corporation: IP Telephony in the Call Center, 1999, `ftp://ftp.dialogic.com/www/pdf/5007.pdf`
4. Genesys Telecommunications Laboratories, Inc.: Customer Service On The Internet – Redefining the Call Center, 1998, `http://www.telemkt.com/whitepapers/genesys-redefining.html`
5. International Telecommunication Union (ITU): Recommendation H.323, Packet-Based Multimedia Communications Systems, `http://www.itu.int/itudoc/itu-t/rec/h/h323.html`
6. Microsoft Corporation: CallTo URL Syntax, December 05, 2000, `http://msdn.microsoft.com/library/psdk/netmeet/nm3_1l4o.htm`
7. Microsoft Corporation: NetMeeting, `http://www.microsoft.com/windows/netmeeting/`
8. Mozilla Organization: JavaScript, `http://www.mozilla.org/js/`
9. Netscape Communications Corporation: Class netscape.javascript.JSObject, `http://home.netscape.com/eng/mozilla/3.0/handbook/plugins/doc/netscape.javascript.JSObject.html`

10. OpenH323 Project: http://www.openh323.org/
11. Ryan, J.: Voice over IP, The Technology Guide Series techguide.com, 1998, http://www.techguide.com/comm/voiceip.shtml
12. Sun Microsystems, Inc.: Enterprise JavaBeans Technology, http://java.sun.com/products/ejb/
13. Sun Microsystems, Inc.: Java Foundation Classes, JFC/Swing, http://java.sun.com/products/jfc/
14. Sun Microsystems, Inc.: Java Homepage, http://java.sun.com/
15. Sun Microsystems, Inc.: Java Plug-in, http://java.sun.com/products/plugin/
16. Sun Microsystems, Inc.: Java Remote Method Invocation Specification, Java 2 SDK, version 1.3.0, December 1999, ftp://ftp.java.sun.com/docs/j2se1.3/rmi-spec-1.3.pdf
17. Sun Microsystems, Inc.: JavaServer Pages, http://java.sun.com/products/jsp/
18. Sun Microsystems, Inc.: Java Web Start, http://java.sun.com/products/javawebstart/architecture.html
19. W3C: HTTP – Hypertext Transfer Protocol, March 27, 2001, http://www.w3.org/Protocols/