SOFTWARE

Is Enterprise-Level, Open Standard, Open Source VOIP A Myth?

Deke Kassabian

Not for a number of universities, where such systems are in use today in production environments.

Is open, SIP-based telephony achievable for the enterprise? Lately, I am being told by industry consultants, as well as the majority of trade press articles and vendor presentations, that it isn't—that secure and feature-rich IP-telephony can be achieved only through the proprietary extensions of the established telephony vendors, at least for now. But communications professionals at several major universities are collaborating to develop these systems, based on open source and open standards, for production use in their own enterprises. Their progress is encouraging.

Today, open source approaches to large production services are probably most appropriate for organizations that employ qualified IT staff who are comfortable building and configuring open source packages, and integrating them with their existing environment. If some staff also are software developers, who are qualified to modify or extend source code in the relevant languages, then open source choices become even more attractive.

But even organizations that lack such staff can take advantage of open source software, by availing themselves of professional services to do the installation, integration and modification work. For many open source packages, such professional service companies are widely available and can be very cost effective when compared with the full costs of commercial, closed source alternatives.

Thousands of faculty and staff members at a range of large and small colleges and universities in the U.S. already are being served successfully by open source VOIP, using open standard protocols alone, not the proprietary vendor extensions that are currently common in the industry. If these deployments continue to be successful, they'll constitute a model for enterprise VOIP that could make for a viable alternative to packaged closed source vendor solutions.

They already have developed provably scalable models to handle tens of thousands of phones, with such features as call park, transfer, ring groups, hunt groups, unified messaging and more. Shared Line Appearance will be integrated in the next year. Within a few years, more than 75,000 lines will be deployed and in full service across higher-education campuses in the U.S.

The purpose of this article is to briefly describe the aspects of open, SIP (Session Initiation Protocol)-based telephony that most completely mirror the conventional vendors' IP-PBX offerings, and those that the open source community is still working to develop. We also will explore the reasons why open source SIP-based telephony suits some organizations better than others today, and touch on a future in which phones, laptops, handhelds and other SIP user agents can take part in a range of unified communications and unified addressing, all in the context of rich presence.

But first, let's be sure we are on the same page about terms (see "Open Source And Open Standards: Definitions And Concepts"), and tackle the biggest question of all: Why would any organization want to build its own open source SIP-based telephony system?

Why Not "Just" Buy A System Or A Service?

With solid IP-telephony products available from well-established vendors, and with hosted "IP Centrex" services available in many of the service provider clouds, it may not be immediately clear why anyone would pursue an open standard, open source approach. One reason is cost, but often the bigger reasons are flexibility and control.

The flexibility of standards-based solutions not only allows for the deployment of a mix of systems that interoperate using a common dial plan, codecs, etc., but it also allows for the evolution of services. This is important when trying to accommodate different user bases on their (and your) schedule without massive forklift upgrades.

Direct control also makes it easier to integrate the VOIP solution with local middleware (such as identity management infrastructure), and with other applications. Open source and open standards lend themselves to just this kind of control.

Deke Kassabian is senior technology director with the University of Pennsylvania, specializing in communications networks, systems and applications. He can be reached at deke@upenn.edu.

Open Source And Open Standards—Definitions And Concepts

By open standards, I mean those that have been established through the recognized major standards bodies such as the IEEE, IETF, and ITU. Important standards for the purposes of this discussion include crusty old standards such as the IEEE's Ethernet (802.3 series) and LAN bridging (802.1D), the IETF's Internet Protocol (IP, RFC 791), as well as newer standards such as IETF RFC 3261 (Session Initiation Protocol), RFC 3550 (Realtime Transport Protocol) and RFC 3711 (Secure Real-time Transport Protocol), and ITU G.711 and G.729 standard codecs.

By open source, I mean software whose source code is available to the end user, with rights to modify and/or redistribute it. There are many well-known open source projects outside the telephony world. The Apache Web server and the Firefox Web browser, the Postfix mail transport system, the Cyrus and UW IMAP servers, the Thunderbird mail client, and of course, the numerous open source Linux and FreeBSD distributions are just some of the better-known examples.

Organizations that use open source systems usually cite several advantages, including the flexibility to make bug fixes directly, the confidence that comes from the ability to study the source code for security and functionality implementation details, and the ability to modify and integrate open-sourced code into other applications.

When open standards are implemented in open source, there is an excellent opportunity to achieve broad interoperability and to avoid single-vendor lock-in.□

Cost, control and flexibility are the main reasons people choose open source solutions

Beyond cost and control, there's a question of "fit." Many IP-telephony products are designed with the stereotypical corporate IT environment in mind—a monoculture of managed Microsoft desktops in a single managed Active Directory domain. But few university IT environments match this model. Instead, because many of the computers are independently owned by students, or acquired by researchers using grant dollars, their primary users often are inclined to make their own platform choices, and they generally are not inclined to turn over all their administrator rights to a central IT group.

One consequence of this is that, for many universities, some of the most common VOIP solutions, which assume a single Microsoft Active Directory for a range of directory and authentication services, don't apply well.

In any case, because you probably aren't reading it elsewhere, I'm here to tell you that we have had some excellent success in the academic community getting open-sourced, SIP-based telephony systems into production environments. Let's discuss the building of such systems, and the choices that have worked well so far.

Basic Network And Functional Architecture

As shown in Figure 1 (p. 38), the basic building blocks of our IP-telephony architecture are
(1) SIP user agents;
(2) SIP registrar/proxy/redirect servers; and
(3) PSTN gateways.
Optional elements are
(4) media and feature servers, which provide voice mail services and some other special features, and
(5) session border controllers (SBCs), which provide a measure of control for IP-telephony traffic leaving or entering our campus.

Let's look at each of these elements, starting with the user agents:

- SIP clients or user agents (UAs) can be softphones, unified communications clients, or applications which happen to implement a SIP UA in order to bring voice functionality to, say, a customer relationship management application. But the most common SIP UA today is the telephone, and that's the one I'll focus on here.

For the purposes of this article, an IP telephone with a SIP software load is the baseline user access equipment. Dozens of manufacturers make phones with modern SIP software loads. Some popular SIP phones in higher-education settings are from Avaya, Cisco, Grandstream, Linksys, Polycom, snom and others.

Once you have some SIP phones, the question becomes, where to connect them to the network? The most flexible answer would be, "anywhere on the Internet," as this would allow users to be mobile while retaining their phone numbers—and indeed, the protocols natively support this. Furthermore, most Ethernet premises networks are ready, from a facilities standpoint, to support SIP phones, meaning that they have adequate bandwidth and electrical power (either inline, or at power outlets) to every desktop.

For example, our network consists of a distributed campus routing core, with five primary core locations interconnected by 10-Gbps Ethernet. Buildings are connected by 1-Gbps Ethernet, desktops at 10 Mbps and 100 Mbps, with a few servers at 1 Gbps.

But IP phone locations are limited for a variety of practical reasons, among them bandwidth assurance and voice traffic prioritization, configuration server security, and emergency service location information.

SIP or IP trunking can be used as an alternative to local gateways

For example, our initial deployments have involved special-purpose, port-based VLANs (virtual LANs), so that we can logically isolate phones and their traffic, expedite the RTP (Real-Time Protocol) call flows, and correlate certain switch ports with certain locations reported to our Public Safety Access Point (PSAP) for 911.

More recently, we have allowed phones to be moved by their users among specially provisioned and marked ports, but we require the users (or their IT support staff) to notify us, via a Web interface, so that we can pass the updated information along to the PSAP. Eventually, we hope to dynamically recognize moves and report that data to the PSAP in near-real time.

■ SIP proxy servers are the next key building block. These servers will handle registrations of phones and signaling to allow calls to be set up. We have direct experience with two open source projects that provide very feature-rich and scalable proxies: the SIP Express Router (SER) from IPTel.org and the OpenSER project from OpenSER.org.

These SIP proxies perform well and scale easily, allowing thousands of users to be served by relatively inexpensive, low-end servers. For example, we currently use site-replicated Dell 2950 servers running Linux operating system, connected at 100 Mbps and configured with fast disk and reasonably large memory.

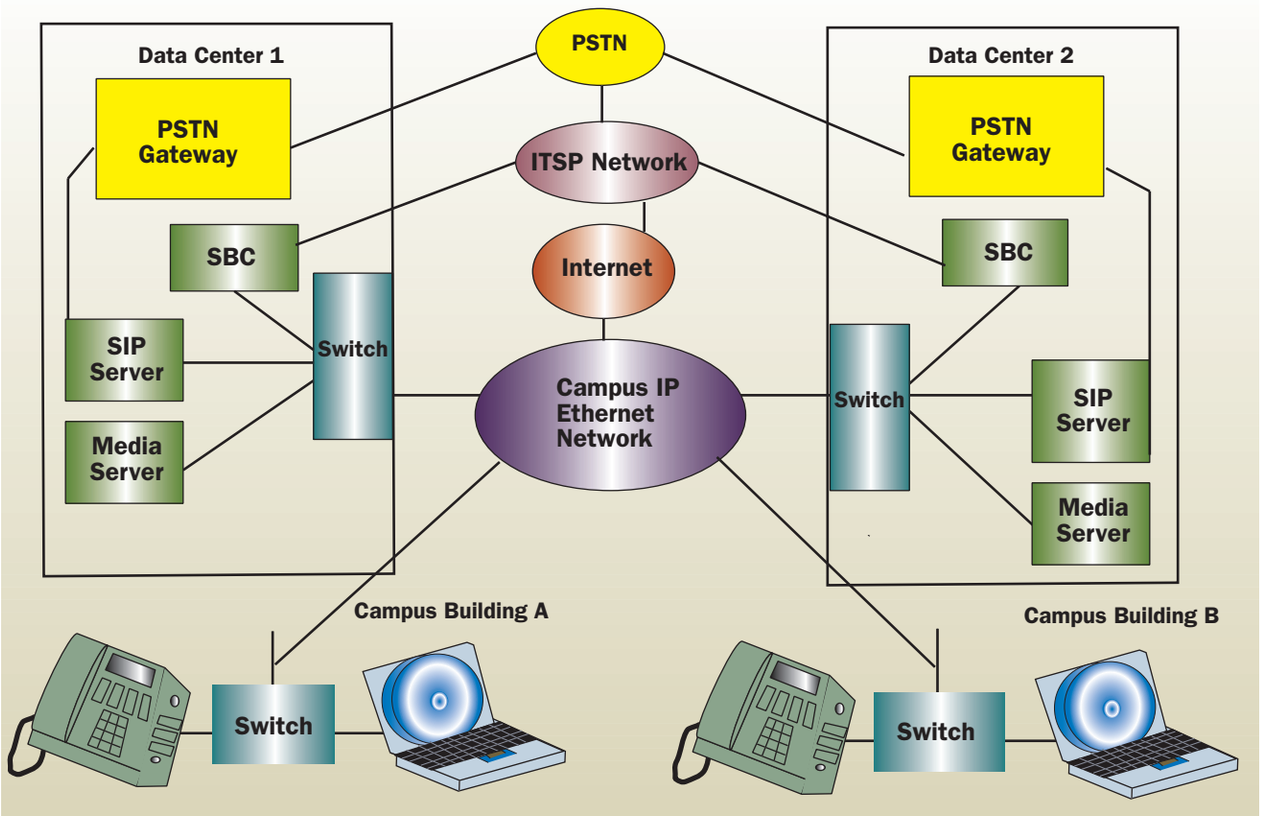
The common configuration among some of the collaborating universities is a pair of SIP proxies, on two physically separate servers, configured to handle phone registrations. Using DNS SRV records, both of these SIP proxies are available at all times to the SIP phones. The two servers are in different physical locations and on different IP subnets, to provide reasonable survivability in case of site failures or network failures.

■ Media and feature servers are used to handle things like voice mail, music on hold, conference bridges and other features. In our experience, Digium's Asterisk provides very flexible and reliable service in these areas. A single Asterisk server running on modest server hardware, as mentioned above, has been tested in performing voice mail recordings with well over 100 simultaneous incoming calls.

A pair of Asterisk servers, replicated to provide redundancy to protect against site and network failures, as described above, can be built, deployed and operated for 20–30 percent less than the cost of comparable proprietary systems.

■ Session border controllers (SBCs) in our environment are used as a control point for connecting with off-site SIP servers. These lightweight SBCs are specially configured instances of the SER or OpenSER proxy servers. Rather than perform registration services for SIP clients or user agents, they perform redirect services for other SIP

FIGURE 1 Open Source SIP-based Telephony



servers. As mentioned above, we use commodity server hardware connected at 100 Mbps.

■ Gateways connect the IP-telephony network to the public switched telephone network (PSTN). Production use implies that IP telephones can call and be called by phones that are offsite, via the PSTN. A common implementation is an IP router with one or more Ethernet networks on the IP-telephony side, and one or more ISDN PRIs on the PSTN side.

In our architecture, we operate multiple such gateways in separate campus locations. We also ensure that the multiple servers involved in call setup can each reach any and all of the gateways.

An alternative to local gateways is the use of IP trunking, sometimes called SIP trunking. Internet telephony service providers (ITSPs) provide this service, taking your SIP call traffic over the IP connection from your site, carrying it with the other VOIP traffic on their network, ultimately reaching a gateway on their network that is as close as possible to your called party. The call then uses that gateway (near the called party) for final call completion.

As with traditional Internet service providers (ISPs), it can be to an enterprise's advantage to have more than one ITSP. This can be efficient and cost effective, helping to achieve higher availability and least cost routing. In our architecture, we operate multiple local gateways and also make use of ITSP services in order to maximize our flexibility and reliability, and also to achieve a reasonable level of control over our call routing, policy and business rules.

Security Architecture

Along with the basic call flow architecture, perhaps the most important aspect to consider is security. One standard approach to VOIP security involves limiting the full use of IP-telephony to the enterprise campus, behind perimeter firewalls, with connectivity to the outside world taking place only through PSTN gateways.

This is a straightforward approach with lots of merits, but it doesn't permit us to take advantage of broader IP connectivity. Why is broad IP connectivity important? There are a number of reasons. For one thing, it allows us to use IP/SIP carrier IP networks with distributed gateways. We plan to use this as our primary means of call completion, while retaining local gateways as last-resort paths, and perhaps for any special-purpose calling not compatible with the carrier network services.

Broad IP connectivity also allows SIP users in different locations to call one another without using gateways and going through the PSTN. Instead of using gateways, SIP back-to-back-user-agents (B2BUAs) work only with SIP proxies for the SIP signaling necessary to cross the IP network administrative boundaries. Once the signaling is taken care of, the UAs communicate directly with one another, without multiple format conversions at gateways.

Finally, broad IP connectivity allows for flexible call scenarios, such as using your enterprise phone infrastructure in purpose-based, but location-independent ways, such as while traveling or at home. We believe that such uses are desirable in terms of both workflow and also for the long term success of SIP.

Our security architecture today makes use of limited access VLANs and subnets for the SIP phones, where our Dynamic Host Configuration Protocol (DHCP) servers assign address space, and our configuration servers provide the configuration and phone software loads we want the users' SIP endpoints to have. These limited-access VLANs help to keep unwanted traffic out while enabling the marking of traffic for prioritization. In addition, the subnets associated with these VLANs can have some campus edge filtering to limit traffic to and from the SIP endpoints, which can't be hardened in quite the same way that a desktop with a conventional operating system might be. Finally, we use a set of very minimal filtering for call handling on our SBCs.

Our security architecture for other VOIP components, including servers and gateways, involves time-tested approaches to hardening, patching the operating systems in a timely way, removing unnecessary services, locking down privileged accounts, enforcing strong authentication, instrumenting systems with high levels of logging and monitoring and more.

With these security mechanisms in place, and because we use publicly routable address space and very little network address translation (NAT), the traversal techniques like STUN and ICE do not play significant roles for us.

Going forward, we hope to make more use of encryption to improve on today's situation with privacy. The use of techniques like Transport Layer Security (TLS)-protected signaling as well as Secure RTP (SRTP) for the encryption of voice traffic are becoming more practical, as server software and handsets implement the necessary pieces, and as the related key exchange approaches become standardized and more widely implemented.

Together, these approaches can make IP-telephony even stronger than legacy telephony in the area of privacy. Our design approaches for open standard IP-telephony account for these possibilities, and our server and handset choices position us to take advantage of them.

Two excellent resources that have proven invaluable for us in the area of security design for enterprise telephony are the Voice over IP Security Alliance (VOIPSA) threat taxonomy (and for that matter, all the VOIPSA resources), and also the Bluebox Podcast. The former helps to look at the range of threats in an organized way, while the latter provides news on developments in the VOIP

We plan to make more use of encryption to improve on today's privacy shortcomings

Our open standard, open source VOIP implementations will scale up beautifully

security space, as well as information on some practical approaches to VOIP security.

What About Telephony Features?

With basic building blocks and a security architecture in place, it's time to use those phones. Clearly, basic calling capabilities are there and they are solid. But what about the simple and even the advanced telephony features needed by the enterprise?

Open standard IP-telephony is quite flexible. We have been able, using standard SIP loads on phones and the features implemented on open standard servers using only standard SIP signaling, to implement features beyond simple "plain old telephone service," including hold, call forward, ring groups, call park, multiple line appearances per set and more.

We have Web interfaces to manage features, and we have the ability to access voice mail using industry standard IMAP email clients. Over the short term, we will be adding a variety of call hunt and find-me/follow-me features, as part of an internal software development project.

But there are some important features we struggle to provide using only open standards. Bridged line appearance (BLA) is one of them: The standards don't currently support this functionality, so most vendor solutions achieve this through proprietary extensions. We have looked at a variety of ways to develop BLA based on emerging standards, and have concluded that there are a small number of reasonable approaches, including some promising current work in the IETF's BLISS group. Some pre-standard work with these emerging standards has resulted in alpha-code for implementing BLA on SIP phones. This work is not production-ready at the time of this writing, but it holds a great deal of promise. Within six months or so, however, we expect to deploy some BLA for production use.

As to the scale of deployment required to handle large enterprises, all indications are that our implementations of open standard, open source VOIP scale up beautifully. We have proven this with simulated loads in thousands of simultaneous calls. Incidentally, SER is included in many Linux distributions and in the Sun Solaris operating system. It also has been proven in numerous very large production deployments, including Freenet.de, the German telecom and Internet services provider, which is serving more than 1 million endpoints, and FWD (formerly FreeWorld Dialup), with half a million endpoints.

Forecast For The Future

Several smaller colleges in the U.S. and in Europe already have deployments in place based on Digium's Asterisk software PBX and on SIPfoundry IP-PBX systems. Several larger universities are collaborating on still more modular and scalable approaches in which the workload is partitioned, using Asterisk and/or SIPfoundry, but also using SER and OpenSER.

One development—a collaboration among four East Coast U.S. universities—has resulted in an approach and a collection of standard tools that could eventually be packaged as a basic "cookbook" to make deployments in other universities more straightforward.

Meanwhile, the work on open standards for VOIP and supporting protocols continues at a strong clip, as does the work on key open source IP-telephony projects. Over time, it is very reasonable to expect more mature standards, more mature code, and as a result, more deployments based on open source and open standards.

It's also likely that IP-telephony will start to migrate away from a predominantly handset-based service to a mixed mode of softphones and hardphones. The softphones will more easily enable integration with other real-time communications tools, and with enterprise presence services. This will in turn help to move telephony away from its traditional role, and toward increasing integration with other collaboration tools, until telephony becomes part of an enterprise's unified communications infrastructure.

Conclusions

Real enterprise telephony, secure and feature-rich, can be achieved today without the use of vendor proprietary extensions. Over time, the feature set will grow, and packaged open solutions will become available. This will in turn motivate more deployments.

For many enterprise IT shops, vendor solutions or carrier IP Centrex may offer the right mix of features, integration, and technical support at the right price. But a real alternative does exist in the open source, open standard world. For those who are comfortable supporting open source and building the integration with already-deployed middleware and applications, the rewards can be great□

Companies Mentioned In This Article

- Avaya (www.avaya.com)
- Blue Box (www.blueboxpodcast.com)
- Cisco (www.cisco.com)
- Digium (www.digium.com)
- Grandstream Networks (www.grandstream.com)
- iptel.org (www.iptel.org)
- Linksys (www.linksys.com)
- Microsoft (www.microsoft.com)
- OpenSER (www.openser.org)
- Polycom (www.polycom.com)
- snom (www.snom.com)
- Voice over IP Security Alliance (http://voipsa.org)