The Aculab SIP Bridge – for third party call control

White paper

Author: Ian Colville

Product Manager, Aculab

Abstract

Without doubt SIP has become the preferred standard for voice over IP (VoIP) call control and looks set to achieve widespread acceptance in both the core and the periphery of next generation networks.

Understandably many solution providers are looking to leverage the power and capabilities that SIP can bring – more efficient and adaptable use of communications hardware and software being a key driver.

Thus for solution developers seeking to enhance their media processing resource card or host media processing centric solutions with SIP signalling there are two options. They can choose a vendor integrated stack, or they can do the work themselves – that is incorporate a third party stack for the call control. Typically with the former they will get a simple, low level API and a tried, tested and supported solution. With the latter they will get as much as they can devote time to.

However in a pure IP environment, the opportunity exists to untie the relationship between media resources and signalling. With Prosody X and SIP Bridge from Aculab, developers get an integrated SIP stack with the traditional confines fundamentally removed. This new paradigm means that resources can be more effectively managed and allocated – a key goal – and third party call control becomes an option.

Now, with Prosody X, solution providers can take advantage of an extended SIP API – we've called it SIP Bridge – that coexists with the generic call control API. In aggregate, the degree of control provided by SIP Bridge is broadly equivalent to the type of third party call control afforded by a TDM PBX with a complex CTI integration. With SIP Bridge this is achieved without the use of any CTI interface, or indeed the use (or cost) of any kind of PBX.

Uniquely the Aculab SIP Bridge becomes a powerful and cost-effective way to build complex IP contact centre and IP PBX type products, with rich media and call control features, using APIs that are common across a wide range of IP and TDM protocols and formats. Despite the APIs being common, IP calls using SIP Bridge enable many call control scenarios that were simply not possible with TDM trunking and protocols.

Aculab's SIP Bridge should not be confused with the term 'SIP bridging' (see IETF RFC 3372, SIP-T), which describes a scenario in which a SIP network acts as a bridge between two segments of the PSTN.

What SIP, who SIP?

Session initiation protocol (SIP) is an end-to-end application layer signalling protocol that is used to control real-time multimedia sessions – data exchanges – between two or more participants across an IP network. The participants are identified as user agents and data exchanges can range from a two-way telephone call to a videoconference or collaborative multimedia conference.

The protocol is a proposed standard defined in RFC 3261 by the Internet Engineering Task Force (IETF) as an agile, general-purpose tool for creating, modifying, and terminating media sessions that works independently of underlying transport protocols and without dependency on the type of session that is being established. SIP does not perform the physical transportation of the media; that is typically done via RTP/UDP/IP.

Undoubtedly SIP is being adopted as the preferred standard for voice over IP (VoIP) call control and looks set to achieve widespread market acceptance in both the core and the periphery of next generation communications networks. Furthermore SIP will play a major role in the deployment of all-IP 3G networks, as defined by Third Generation Partnership Projects (3GPP).

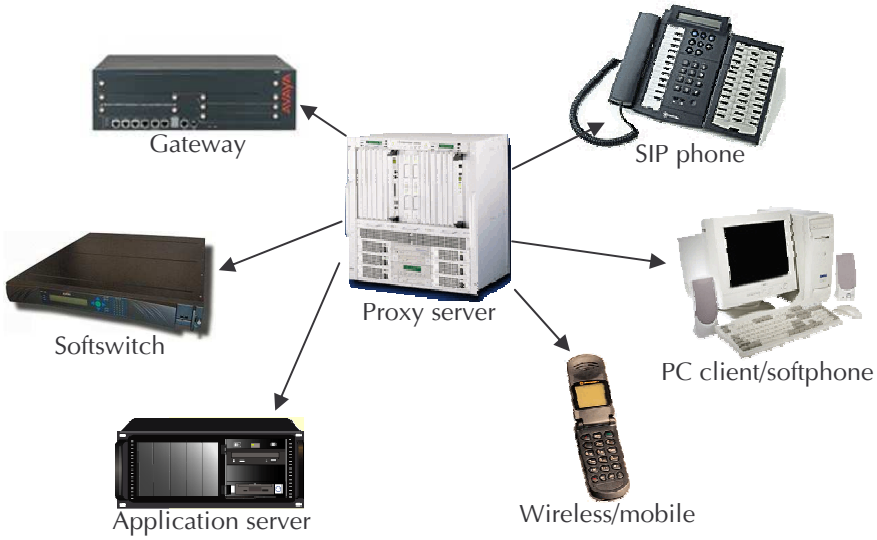
Take a SIP of this cocktail

User agent entities can reside in a SIP phone, a PC-based SIP client (softphone) or a gateway, for example. For these kinds of end-points, SIP messages provide many standard telephony features such as call hold, call transfer, call forward and conferencing set-up, as well as enabling advanced features like mobility, presence and collaboration. Additional SIP entities include proxy servers, redirect servers, back-to-back user agents (B2BUA) and presence servers. These elements are used within the network for routing, re-directing and managing calls and location information.

SIP is able to provide both session connection and session management, independent of application, allowing for its use by multiple application types. The inherent adaptability of SIP means that it covers many different possible uses including telephony, instant messaging (IM), presence, and collaboration. In addition a variety of media session types are supported, for example, audio, video, text, SMS, and file transfer are also within its capabilities to manage.

Consequently market aware developers and systems integrators looking for opportunities could target IP PBXs, softswitches, gateways, application servers, media servers, conference bridges, IVR, unified messaging systems, and contact centres as the kind of products that could be competitively enhanced by SIP's capabilities.

Diagram 1: SIP user agent entities



SIP stacks

SIP stacks and resource cards – a desirable combination

For solution developers seeking SIP functionality together with their DSP media processing resource cards, or host media processing for that matter, there are two options.

The first and most obvious is to choose a supplier, such as Aculab, that offers an integrated SIP stack. This will either be a proprietary stack, developed by the vendor, or a third party or open source stack integrated by the vendor. In such cases the stack will be provided under a cost free licence, or offered under the terms of a software licence for a fee.

Alternatively the developer can procure a third party or open source stack and do the integration work themselves; linking the stack to their application on the one level, and the vendor's media processing resources on another. The stack – usually offered as source code – will normally be purchased for a one-off cost or licensed on some basis and is incorporated via an API.

There are a number of SIP stack vendors, such as Data Connection, Flextronics and Radvision to choose from as well as open source options such as GNU oSIP. It is a case of matching your needs with the features and support available and finding a stack that comes with suitably versatile tools and APIs.

Proprietary or third party – lingering doubts

In using proprietary or third party SIP stacks, there are a number of things that a developer should reflect on, such as whether the stack is truly open i.e., it can be used with any end-point – not just an RTP end-point on the card or selected SIP phones.

A further consideration when using a third party stack is the work involved, notwithstanding the development tools and reference applications available, to bridge it into a user agent application and, more particularly, create the linkage to the resource cards at the API level. This association is needed to control the allocation of resources to a call. For those with the inclination and time available doing the work themselves won't be a problem. On the other hand, when speed is of the essence and time-to-market pressures are acute, developers may wish for a guaranteed, risk free option.

An added complication could be that the media processing resources are not readily accessible to a third party stack, perhaps because the vendor only provides one option – their own integral stack. If the vendor allows the use of a third party stack – any stack or just one particular stack – then an API gateway to the media needs to be provided.

That brings us nicely back to proprietary stacks, or card vendor integrated third party stacks, which amount to the same thing from the developer's viewpoint. What should be weighed up in this approach?

As long as the call control API provides access to the desired SIP capabilities this is a reasonable option. What is typical is that the vendor will choose to wrap the stack into their proprietary API and present SIP as just another call control option. Again, this may be perfectly acceptable, however, it will usually preclude access to low level SDP messages or SIP headers, which may then limit the degree of flexibility available to an application.

For a lot of developers this will not be a problem – a simple, low level API is exactly what they want – and they will get a tried and tested solution. Undoubtedly for others this will be a compromise and leave something to be desired in terms of the granularity of control over SIP calls. So what of the future?

SIP stack enlightening

Initial solutions based on open standard media processing resource cards would typically have the SIP stack assuming that any incoming or outgoing SIP calls will be 'gatewayed' to TDM, either for forwarding to TDM trunks, or to be handled internally by IVR in a TDM environment.

To achieve this, the SIP stack negotiated the type of codec, determined RTP end-points etc., and communicated directly with the resource card DSPs to fulfil those negotiated functions. The resource card acted as an end-point, under control of the SIP stack, which acted as a user agent, requesting or responding to a call.

Now, in a pure IP environment, in instances where calls are simply being connected between end-points, only signalling is required. Phone calls between people may not require any media resources (unless they need to be recorded, or an intelligent device needs to 'listen' into the call, for example). Only when a call is being terminated by an application, for example voicemail, are media resources always required. This means that such systems will need less media channels compared to signalling channels.

With some media processing resource card vendors, although signalling and media are separated, a signalling channel will depend on a media channel on a 1:1 basis and the media and signalling will both be terminated in the same place, as is the case for TDM calls. But it doesn't have to be this way. If these dependencies are removed – presenting the ability to more effectively manage and allocate resources – third party call control becomes an option.

Third party call control refers to the ability of an entity, such as an IP PBX, to create and manage a call in which the media actually flows between other entities. SIP facilitates this through its separation of signalling and media – the signalling can be managed by one device while the media is handled by another device.

Fundamentally giving an application the ability to handle third party call control in order to manipulate end-points is a good idea when seeking to build complex, value added solutions. Since such connections could equally be DSPs on a resource card, softphones or SIP phones – in fact any internal or external RTP end-point elsewhere on the network – the opportunities created are exciting.

Figure 1 – definitions:

First party call control	Third party call control
This takes place when the user, being one of the parties speaking, exercises control over a call's connection characteristics. In other words: dial another party; disconnect from another party; or dial commands to add a third party into a conference call.	This involves an authority other than the parties active in a call exercising its ability to change the call's connection characteristics (in the past this would have been the operator). Now the third party is more likely to be a carrier's switching system or an IP PBX.

Introducing the SIP Bridge API

Aculab offers a proprietary SIP stack that is fully integrated with Aculab's product range. It is provided as part of Aculab's software product suite under a cost free licence. When used with Aculab's generic – protocol independent – call control API it is intended for gateway or terminating (and transferring) applications such as voicemail and IVR systems. Therefore the appropriate 'gatewaying' or terminating end-points are media resources – either on an IP telephony or Prosody card, or on Aculab's host media processing product, Prosody S. Aculab also makes the Prosody media processing resources readily accessible to a third party SIP stack through an optional media gateway API.

Now, with Prosody X, solution providers can take advantage of an extended SIP API – SIP Bridge – that coexists with the generic API. The objective of the comprehensive SIP Bridge API extension is to provide a set of SIP features that cannot be represented in a protocol independent manner.

The enhancements made to the SIP API present improved scope for application handling of calls. In the generic call control API the implementation of `call_openout()` for SIP makes logical assumptions on behalf of the application in the setup of the outbound `INVITE` message. In the extended API the application is able to affect the format of the `INVITE` to a greater extent. It has been enhanced to allow the addition of custom message bodies and SIP headers and, in addition, the extended API offers application configuration of session description protocol (SDP) encoded bodies allowing developers fuller access to the power of SIP.

Uniquely SIP Bridge further increases the power of Aculab's SIP stack as the 1:1 dependence between signalling and media channels is removed – the media resource acquisition and configuration is handled outside the SIP stack – and the assumption that the media and signalling will both be terminated in the same place is broken. This allows developers to build B2BUA and third party call control products.

So third party call control using `re-INVITE` is now possible, enabling the re-direction of RTP streams between end-points. With Aculab's SIP stack a call can terminate its RTP media stream either on a media processing resource, such as a Prosody X channel, or on an RTP end-point on a separate device, such as another phone or SIP entity. This call management is further facilitated with the fine control of SDP content now available for media negotiation, including non-audio payload types, and multiple RTP streams.

Additionally, mid-call signalling (`INFO`) utilising custom headers and message bodies enables supplementary features to be implemented. This is ideal for IP PBX and contact centre type applications, where forwarding PSTN information, carrying DTMF digits, or even images and account balance information, can be useful.

Furthermore the management and allocation of resources is inherently more effective as expensive resources (jitter buffers, echo cancellation, codec support and RTP packetisation) needed to terminate RTP are no longer tied to every call.

With the removal of the binding between signalling and media channels, the effect is that the concept of a 'call' now refers specifically to a SIP session (between initial `INVITE`, and `BYE/OK`), and that this is now entirely separate from the media path. Indeed it is separate from any internal media processing resources supported by Prosody X.

Using SIP Bridge lifts the restrictions on what SIP can do. Fundamentally many more SIP calls can be handled than there are (or need be) media resource channels, greatly enhancing the resource efficiency of any system.

All under control, Sir!

From an API angle, to safely sever the relationship between signalling and media, there are two pieces of information which must be conveyed to and from the application – the SDP element (or an abstraction thereof) of an inbound or outbound SIP message, and the current 'media state' of a SIP call (the IP settings of the participants and the media payload that they are communicating).

In a typical gateway scenario, a SIP call terminates a negotiated RTP media stream (after a conversion) on the inbound side of the TDM switch. In an all-IP situation a SIP call terminates its RTP stream on the IP side of a Prosody X card media channel. From here, in either case, a recorded message or prompt may be played out. The call is then potentially (blind) transferred to another SIP (or TDM) end-point.

But with SIP Bridge it is now possible to put that call on hold (releasing the Prosody X card VoIP media channel resource along with its IP end-point) and make a new SIP call that can independently re-use that channel resource. The SIP stack is separated from the RTP capability and can manage a call without having to terminate an RTP end-point. In other words, the number of SIP calls that can be established and maintained, greatly exceeds the number of available (provisioned) media processing channels.

Additionally that initial call, whether held or not, can subsequently be directly connected via RTP media streams to another call, itself established via any SIP signalling channel, not necessarily via the same signalling channel. The key point here is that the SIP part of the calls are separately terminated on the SIP Bridge stack, and are controlled by the Aculab API in the normal manner, even when they are not associated with a current Prosody X channel, or utilising any media processing resource.

As a result of this any transferred or held call can be recovered (or `re-INVITE`d) under API control to the Prosody X card at any time for a further IVR session, for attachment to conference resources, for recording the call, for transfer to a supervisor, or for other calls. RTP media paths are made and broken as required whilst signalling is always present in a call and the application controls these actions – creating, and modifying calls via the API.

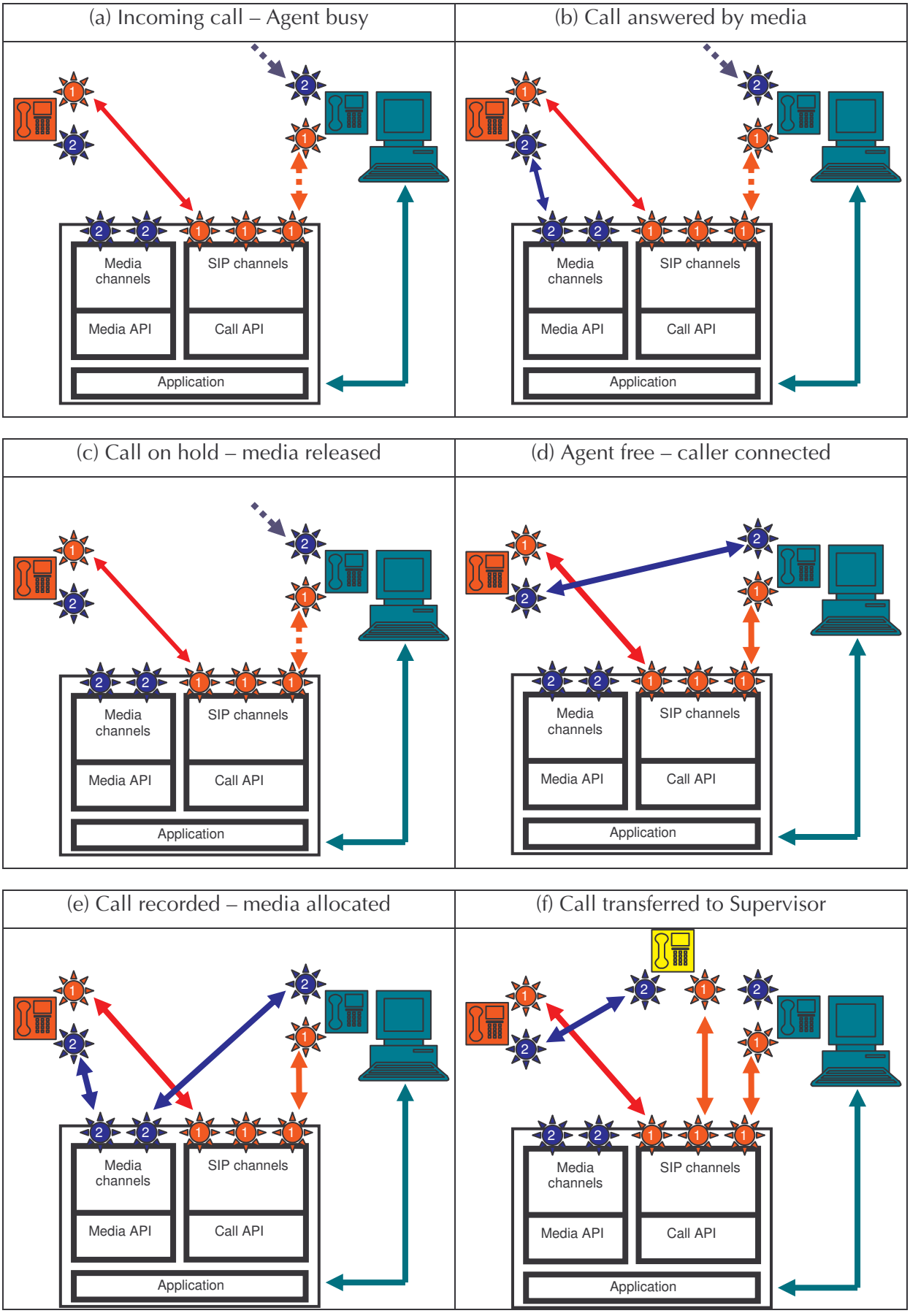
The contact centre scenario

The sequence (a) to (f) in diagram 2 illustrates a contact centre scenario with a media server providing media processing resource channels and SIP Bridge handling SIP call control channels. The '1' stars represent RTP end-points and the '2' stars represent SIP end-points.

In (a) an incoming call finds the call centre Agent busy on an existing call; the Agent is 'owned' by another channel of the application. In (b), because the Agent is busy, the call is answered by presenting a voice prompt menu playback (to determine the service required) on the media channel. So in (c) the call is placed on hold, queuing for a free Agent, and the media channel is released. Later in (d), when an Agent becomes free, the caller is connected to talk to the Agent. Optionally in (e), if the application or the Agent requires it, the conversation can be recorded by means of media channels being allocated. Finally, in (f) the caller is shown having been transferred to a Supervisor, and once again, the media channel is not needed and so is released.

The key point to note is that the caller and the Agent 'calls' are controlled by the application from beginning to end. Even when the media path has been 'transferred' (e.g., connected to an Agent) the call control hasn't. From an API standpoint, the 'calls' are the SIP sessions. The media is variable and might not be connected at any point at all during the 'call'. Media channels are only used as and when they are required and many more channels of SIP signalling might be used than media channels. The programming model is consistent, and quite simple, regardless of the service, and handling third party call control is very much like, and no more difficult than, but certainly more powerful than, simple first party call control.

Diagram 2 (a-f) – third party control in a contact centre



The third man

Today as the move to next generation IP networks gathers pace we can envisage SIP oriented developments with enhanced capabilities in addition to simply 'gatewaying' or terminating SIP calls. SIP Bridge's decomposed architecture can allow control of SIP sessions that do not terminate on media processing resources at all, and can cause external SIP devices to connect to each other, whilst retaining control of the separate parts of the call. In SIP terms, it is a B2BUA, and acts as a controller for the purposes of third party call control.

In fact, for each call that appears to go 'through' the SIP Bridge, each leg is, in SIP terms, terminated separately on the SIP Bridge. And even though the RTP media streams might well go end-to-end, and never touch the Prosody X card, they may equally well be terminated (or originated) there, for prompting, conferencing or recording purposes.

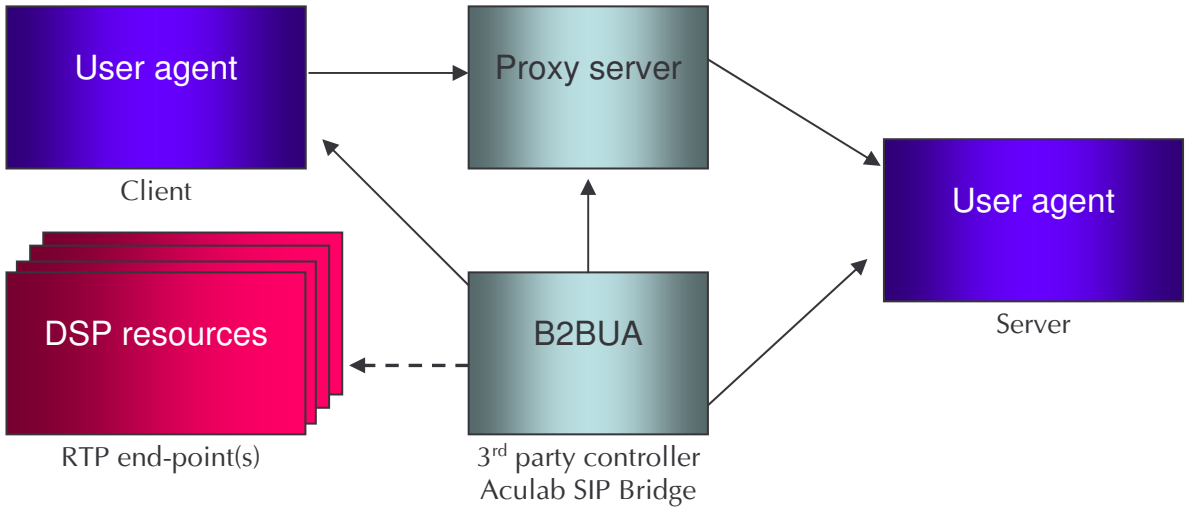
Accordingly, the Aculab SIP Bridge must be considered as a multiple set of (simple) SIP end-points, but with rich internal APIs for SIP call control, and manipulating media streams.

Overall, the degree of control provided by SIP Bridge is broadly equivalent to the type of third party call control afforded by a TDM PBX with a complex CTI integration; and this is achieved without the use of any CTI interface, or indeed the use (or cost) of any kind of PBX.

Back-to-back they faced each other

A B2BUA is a logical entity that receives a request and processes it as a user agent server (UAS). In order to determine how the request should be answered, it acts as a user agent client (UAC) and generates requests. Unlike a proxy server, it maintains dialog state and must participate in all requests sent on the dialogs it has established. A B2BUA is in effect a standards-based mechanism for achieving third party call control.

Diagram 3 – B2BUA as third party controller



Conclusion

The Aculab SIP Bridge is a powerful, and cost-effective way to build complex IP contact centre and IP PBX type products, with rich media and call control features, using APIs that are common across a wide range of IP and TDM protocols and formats, including SIP, H.323, C7 (SS7) ISUP, Q.931, Q.SIG, and all of the other national and international protocols supported by Aculab E1 and T1 products.

Despite the APIs being common, IP calls using SIP and SIP Bridge enable many call control scenarios simply not possible with TDM trunking and protocols.

Until now, implementing complex systems such as IP contact centres required the custom integration of a third party SIP stack, incorporating complex SIP proxy functionality. Developers can now build a solution that is an alternative to a proprietary TDM PBX or contact centre system, combining an appropriate application, including IVR, voicemail, unified messaging, etc., to produce rich corporate collaboration features – and all without the use of any CTI interface.

The benefits of using the Aculab SIP Bridge are available to any solution provider building IP applications, which perform functions such as; call transfers, call recording, and conferencing. Key benefits include:

- More efficient use of resources – as signalling and media are separated, media resources are only used when required
- Ease of deployment – SIP Bridge has been written, integrated and tested by Aculab so time doesn't need to be devoted to integrating a third party stack
- Allows solution providers to harness the power of SIP together with Prosody X
- Allows the development of low cost, value added solutions – which could be an alternative to proprietary IP PBX equipment – without the use of any CTI interface

About Aculab

Aculab enables developers and systems integrators to produce a variety of high performance communications solutions. Aculab's portfolio offers an exceptional mix of capabilities that are easy to integrate and bring real value – reduced costs, increased customer satisfaction and competitive advantage. A complete range of open standards building block technologies for use within telco or enterprise environments, as well as essential support services, are offered. Products include media resources, digital network access, VoIP, fax, speech processing and conferencing. Support is available to help developers through each stage of their product's life cycle including pre-sales consultancy, technical support, training and marketing.

Aculab is a full member of the SIP Forum, a principle sponsor of the SIP Center, and a contributor to the Columbia Universities' SIP implementers' mailing list.

Copyright and other notices

For more information, visit Aculab's website at www.aculab.com or email info@aculab.com

Aculab plc, Lakeside, Bramley Road, Mount Farm, Milton Keynes. Buckinghamshire. MK1 1PT. United Kingdom. Tel: +44 (0) 1908 273 800

The information in this white paper is provided by Aculab plc, an enabling technology provider, and is for informational purposes only. Nothing in this publication forms any part of any contract.

The information contained herein is based on material, which Aculab, based on its best efforts, believes to be reliable, but no representation is made as to its completeness or accuracy. Aculab make no warranties, express or implied, in this document. E&OE