

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 05, 2014

E. Ivov
Jitsi
P. Saint-Andre
Cisco Systems, Inc.
E. Marocco
Telecom Italia
October 02, 2013

CUSAX: Combined Use of the Session Initiation Protocol (SIP) and the
Extensible Messaging and Presence Protocol (XMPP)
draft-ivov-xmpp-cusax-09

Abstract

This document suggests some strategies for the combined use of the Session Initiation Protocol (SIP) and the Extensible Messaging and Presence Protocol (XMPP) both in user-oriented clients and in deployed servers. Such strategies, which mainly consist of configuration changes and minimal software modifications to existing clients and servers, aim to provide a single, full-featured, real-time communication service by using complementary subsets of features from SIP and from XMPP. Typically such subsets consist of telephony capabilities from SIP and instant messaging and presence capabilities from XMPP. This document does not define any new protocols or syntax for either SIP or XMPP, and by intent does not attempt to standardize "best current practices". Instead, it merely aims to provide practical guidance to those who are interested in the combined use of SIP and XMPP for real-time communication.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 05, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Client Bootstrap	5
3. Operation	6
3.1. Server-Side Setup	7
3.2. Service Management	7
3.3. Client-Side Discovery and Usability	8
3.4. Indicating a Relationship Between SIP and XMPP Accounts	9
3.5. Matching Incoming SIP Calls to XMPP JIDs	10
4. Multi-Party Interactions	11
5. Federation	12
6. Summary of Suggested Strategies	13
7. IANA Considerations	14
8. Security Considerations	14
9. References	15
9.1. Normative References	15
9.2. Informative References	16
Appendix A. Acknowledgements	17
Authors' Addresses	18

1. Introduction

Historically SIP [RFC3261] and XMPP [RFC6120] have often been implemented and deployed with different purposes: from its very start SIP's primary goal has been to provide a means of conducting "Internet telephone calls". XMPP on the other hand, has, from its Jabber days, been mostly used for instant messaging and presence [RFC6121], as well as related services such as groupchat rooms [XEP-0045].

For various reasons, these trends have continued through the years even after each of the protocols had been equipped to provide the features it was initially lacking:

- o In the context of the SIMPLE working group, the IETF has defined a number of protocols and protocol extensions that not only allow for SIP to be used for regular instant messaging and presence but that also provide mechanisms for related features such as multi-party chat, server-stored contact lists, and file transfer [RFC6914].
- o Similarly, the XMPP community and the XMPP Standards Foundation have worked on defining a number of XMPP Extension Protocols (XEPs) that provide XMPP implementations with the means of establishing end-to-end sessions. These extensions are often jointly referred to as Jingle [XEP-0166] and arguably their most popular use case is audio and video calling [XEP-0167].

However, although SIP has been extended for messaging and presence and XMPP has been extended for voice and video, the reality is that SIP remains the protocol of choice for telephony-like services and XMPP remains the protocol of choice for IM and presence services. As a result, a number of adopters have found themselves needing features that are not offered by any single-protocol solution, but that separately exist in SIP and XMPP implementations. The idea of seamlessly using both protocols together would hence often appeal to service providers and users. Most often, such a service would employ SIP exclusively for audio, video, and telephony services and rely on XMPP for anything else varying from chat, contact list management, and presence to whiteboarding and exchanging files. Because these services and clients involve the combined use of SIP and XMPP, we label them "CUSAX" for short.

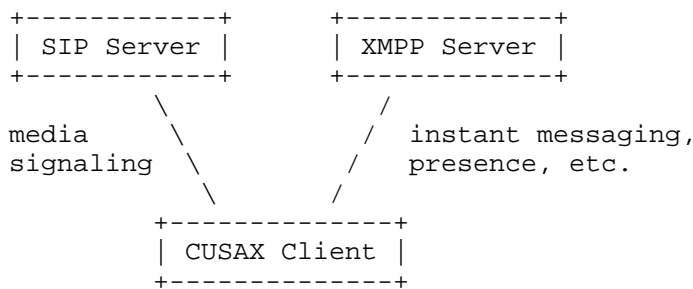


Figure 1: Division of Responsibilities

This document suggests different configuration options and minimal modifications to existing software so that clients and servers can

offer these hybrid services while providing an optimal user experience. It covers server discovery, determining a SIP Address of Record (AOR) while using XMPP, and determining an XMPP Jabber Identifier ("JID") from incoming SIP requests. Most of the text here pertains to client behavior but we also suggest certain server-side configurations and operational strategies. The document also discusses significant security considerations that can arise when offering a dual-protocol solution, and provides advice for avoiding security mismatches that would result in degraded communications security for end users.

Note that this document is focused on coexistence of SIP and XMPP functionality in end-user-oriented clients. By intent it does not define methods for protocol-level mapping between SIP and XMPP, as might be used within a server-side gateway between a SIP network and an XMPP network (a separate series of documents has been produced that defines such mappings). More generally, this document does not describe service policies for inter-domain communication (often called "federation") between service providers (e.g., how a service provider that offers a CUSAX service might communicate with a SIP-only or XMPP-only service), nor does it describe the reasons why a service provider might choose SIP or XMPP for various features.

This document concentrates on use cases where the SIP services and XMPP services are controlled by one and the same provider, since that assumption greatly simplifies both client implementation and server-side deployment (e.g., a single service provider can enforce common or coordinated policies across both the SIP and XMPP aspects of a CUSAX service, which is not possible if a SIP service is offered by one provider and an XMPP service is offered by another provider). Since this document is of an informational nature, it is not unreasonable for clients to apply some of the guidelines here even in cases where there is no established relationship between the SIP and the XMPP services (for example, it is reasonable for a client to provide a way for its users to easily start a call to a phone number or SIP URI found in a vCard or obtained from a user directory). However, the strategies to pursue in such cases are left to application developers.

This document makes a further simplifying assumption by discussing only the use of a single client, not use of and coordination among multiple endpoints controlled by the same user (e.g., user agents running simultaneously on a laptop computer, tablet, and mobile phone). Although user agents running on separate endpoints might themselves be CUSAX clients or might engage in different aspects of an interaction (e.g., a user might employ her mobile phone for audio and her tablet for video and text chat), such usage complicates the guidelines for developers of user agents and therefore is left as a matter of implementation for now.

It is important to note that this document does not attempt to standardize "best current practices" in the sense defined in the Internet Standards Process [RFC2026]. Instead, it collects together informational documentation about some strategies that might prove helpful to those who implement and deploy combined SIP-XMPP software and services. With sufficient use and appropriate modification to incorporate the lessons of experience, these strategies might someday form the basis for standardization of best current practices.

2. Client Bootstrap

One of the main problems of using two distinct protocols when providing one service is the impact on usability. Email services, for example, have long been affected by the mixed use of SMTP for outgoing mail and POP3 or IMAP for incoming mail. Although standard service discovery methods (such as the proper DNS records) make it possible for a user agent to locate the right host(s) for connect purposes, they do not provide the kind of detailed information that is needed to actually configure the user agent for use with the service. As a result, it is rather complicated for inexperienced users to configure a mail client and start using it with a new service, and as a result Internet service providers often need to provide configuration instructions for various mail clients. Client developers and communication device manufacturers on the other hand often ship with a number of so-called "wizard" interfaces that enable users to easily configure accounts with a number of popular email services. Although this may improve the situation to some extent, the user experience is still clearly sub-optimal.

While it should be possible for CUSAX users to manually configure their separate SIP and XMPP accounts (often using "wizards"), service providers offering CUSAX services to users of dual-stack SIP/XMPP clients ought to provide methods for online provisioning, typically by means of a web-based service at an HTTPS URL (naturally, single-purpose SIP services or XMPP services could offer such methods as well, but they can be especially helpful where the two aspects of the CUSAX service need to have several configuration options in common).

Although the specifics of such mechanisms are outside the scope of this document, they should make it possible for a service provider to remotely configure the clients based on minimal user input (e.g., only a user ID and password). As far as the authors are aware, no open protocol for endpoint configuration is yet available and adopted; however, application developers are encouraged to explore the potential for future progress in this space (e.g., perhaps based on technologies such as WebFinger [RFC7033]).

By default, when a CUSAX client is used in concert with SIP and XMPP accounts that have a CUSAX relationship (see Section 3.4), the client should disable audio and video calling over XMPP and disable instant messaging and presence over SIP. (It is a matter of implementation whether a CUSAX client allows a user to override these defaults in various ways, e.g., by domain, by individual contact, or by device.) The main advantage of this approach is that a client would employ the most relevant features from both SIP and XMPP when used in the context of a CUSAX service. Note that this default configuration does not apply to standalone SIP accounts or XMPP accounts, for which other settings are likely to be more appropriate (see Section 3.4 for details).

Once a client has been provisioned, it needs to independently log into the SIP account and XMPP account that make up the CUSAX "service" and then maintain both connections.

In order to improve the user experience, when reporting connection status a CUSAX client may wish to present the XMPP connection as an "instant messaging" or a "chat" account and the SIP connection as a "Voice and Video" or a "Telephony" connection. The exact naming is of course entirely up to implementers. The point is that, in cases where SIP and XMPP are components of a service offered by a single provider, such presentation could help users better understand why they are being shown two different connections for what they perceive as a single service. It could alleviate especially situations where one of these connections is disrupted while the other one is still active. Alternatively, the developers of a CUSAX client or the providers of a CUSAX service might decide to force a client to completely disconnect unless both aspects are successfully connected.

Clients may also choose to delay their XMPP connection until they have been successfully registered on SIP. This would help avoid the situation where a user appears online to her contacts but calling the user's client would fail because the user's client is still connecting to the SIP aspect of the CUSAX service.

3. Operation

Once a CUSAX client has been provisioned and authorized to connect to the corresponding SIP and XMPP services it would proceed by retrieving its XMPP roster.

The client should use XMPP for most forms of communication with the contacts from this roster, which will occur naturally because they were retrieved through XMPP. Audio/video features however, would typically be disabled in the XMPP stack, so media-related communication based on these features (e.g., direct calls, conferences, desktop streaming, etc.) would happen over SIP. The rest of this section describes deployment, discovery, usability and linking semantics that enable CUSAX clients to seamlessly use SIP for these features.

3.1. Server-Side Setup

In order for CUSAX to function properly, XMPP service administrators should make sure that at least one of the vCard [RFC6350] "tel" fields for each contact is properly populated with a SIP URI for the user's address at the SIP audio/video service provided by the CUSAX server. There are no limitations as to the form of that number. For example while it is desirable to maintain a certain consistency between SIP AORs and XMPP JIDs, that is by no means required. It is quite important however that the phone number or SIP AOR stored in the vCard be reachable through the SIP aspect of this CUSAX service. (The same considerations apply even if the directory storage format is not vCard storage over XMPP as described by [XEP-0054] or [XEP-0292].)

Administrators may also choose to include the "video" tel type defined in [RFC6350] for accounts that would be capable of handling video communication.

To ensure that the foregoing approach is always respected, service providers might consider validating the values of vCard "tel" fields before storing changes. Of course such validation would be feasible only in cases where a single provider controls both the XMPP and the SIP service since such providers would "know" (e.g., based on use of a common user database for both services) what SIP AOR corresponds to a given XMPP user.

3.2. Service Management

The task of operating and managing a standalone SIP service or XMPP service is not always easy. Combining the two into a unified service introduces additional challenges, including:

- o The necessity of opening additional ports on the client side if SIP functionality is added to an existing XMPP deployment or vice-versa.
- o The potential for important differences in security posture across SIP and XMPP (e.g., SIP servers and XMPP servers might support different TLS ciphersuites).
- o The need for, ideally, a common authentication backend and other infrastructure that is shared across the SIP and XMPP aspects of the combined service.
- o Coordinated monitoring and logging of the SIP and XMPP servers to enable the correlation of incidents and the pinpointing of problems.
- o The difficulty of troubleshooting client-side issues, e.g. if the client loses connectivity for XMPP but maintains its SIP connection.

Although separation of functionality (SIP for media, XMPP for IM and presence) can help to ease the operational burden to some extent, service providers are urged to address the foregoing challenges and similar issues when preparing to launch a CUSAX service.

Beyond the issues listed above, service providers might want to be aware of more subtle operational issues that can arise. For example, if a service provider uses different network operators for the SIP service and the XMPP service, end-to-end connectivity might be more reliable or consistent in one service than in the other service. Similar issues can arise when the media path and the signaling path go over different networks, even in standalone SIP or XMPP services. Providers of CUSAX services are advised to consider the potential for such topologies to cause operational challenges.

3.3. Client-Side Discovery and Usability

When rendering the roster for a particular XMPP account CUSAX clients should make sure that users are presented with a "Call" option for each roster entry that has a properly set "tel" field. This is the case even if calling features have been disabled for that particular XMPP account, as advised by this document. The usefulness of such a feature is not limited to CUSAX. After all, numbers are entered in vCards or stored in directories in order to be dialed and called. Hence, as long as an XMPP client has any means of conducting a call it may wish to make it possible for the user to easily dial any numbers that it learned through whatever means.

Clients that have separate triggers (e.g., buttons) for audio calls and video calls may choose to use the presence or absence of the "video" tel type defined in [RFC6350] as the basis for choosing whether to enable or disable the possibility for starting video calls (i.e., if there is no "video" tel type for a particular contact, the client could disable the "video call" button for that contact).

In addition to discovering phone numbers from vCards or user directories, clients may also check for alternative communication methods as advertised in XMPP presence broadcasts and Personal Eventing Protocol nodes as described in XEP-0152: Reachability Addresses [XEP-0152]. However, these indications are merely hints, and a receiving client ought not associate a SIP address and an XMPP address unless it has some way to verify the relationship (e.g., the vCard of the XMPP account lists the SIP address and the vCard of the SIP account lists the XMPP address, or the relationship is made explicit in a record provided by a trusted directory). Alternatively or in cases where vCard or directory data is not available, a CUSAX client could take the user's own address book as the canonical source for contact addresses.

3.4. Indicating a Relationship Between SIP and XMPP Accounts

In order to improve usability, in cases where clients are provisioned with only a single telephony-capable account they ought to initiate calls immediately upon user request without asking users to indicate an account that the call should go through. This way CUSAX users (whose only account with calling capabilities is usually the SIP part of their service) would have a better experience, since from the user's perspective calls "just work at the click of a button".

In some cases however, clients will be configured with more than the two XMPP and SIP accounts provisioned by the CUSAX provider. Users are likely to add additional stand-alone XMPP or SIP accounts (or accounts for other communications protocols), any of which might have both telephony and instant messaging capabilities. Such situations can introduce additional ambiguity since all of the telephony-capable accounts could be used for calling the numbers the client has learned from vCards or directories.

To avoid such confusion, client implementers and CUSAX service providers may choose to indicate the existence of a special relationship between the SIP and XMPP accounts of a CUSAX service. For example, let's say that Alice's service provider has opened both an XMPP account and a SIP account for her. During or after provisioning, her client could indicate that `alice@xmpp.example.com` has a CUSAX relationship to `alice@sip.example.com` (i.e., that they are two aspects of the same service). This way whenever Alice

triggers a call to a contact in her XMPP roster, the client would preferentially initiate this call through her example.com SIP account even if other possibilities exist (such as the XMPP account where the vCard was obtained or a SIP account with another provider). Similarly, the client would preferentially initiate textual chat sessions using her XMPP account.

If, on the other hand, no relationship has been configured or discovered between a SIP account and an XMPP account, and the client is aware of multiple telephony-capable accounts, it ought to present the user with the option of using XMPP Jingle as one method for engaging in audio and video interactions with a contact who has an XMPP address. This can help to ensure that a CUSAX user can complete audio and video calls with XMPP users who are not part of a CUSAX deployment.

3.5. Matching Incoming SIP Calls to XMPP JIDs

When receiving a SIP call, a CUSAX client may wish to determine the identity of the caller and a corresponding XMPP roster entry so that the receiving user could revert to chatting or other forms of communication that require XMPP. To do so, a CUSAX client could search the user's roster for an entry whose vCard has a "tel" field matching the originator of the call. In addition, in order to avoid the effort of iterating over the entire roster of the user and retrieving vCards for all of the user's contacts, the receiving client may guess at the identity of the caller based a SIP Call-Info header whose 'purpose' header field parameter has a value of "impp" as described in [RFC6993]. To enable this usage, a sending client would need to include such a Call-Info header in the SIP messages that it sends when initiating a call. An example follows.

```
Call-Info: <xmpp:alice@xmpp.example.com> ;purpose=impp
```

Note that the information from the Call-Info header should only be used as a cue: the actual AOR-to-JID binding would still need to be confirmed by the vCard of a contact in the receiving user's roster or through some other trusted means (such as an enterprise directory). If this confirmation succeeds the client would not need to search the entire roster and retrieve all vCards. Not performing the check might enable any caller (including malicious ones) to employ someone else's identity and perform various scams or Man-in-the-Middle attacks.

However, although an AOR-to-JID binding can be a helpful hint to the user, nothing in the foregoing paragraph ought to be construed as necessarily discouraging users, clients, or service providers from

accepting calls originated by entities that are not established contacts of the user (e.g., as reflected in the user's roster); that is a policy matter for the user, client, or service provider.

4. Multi-Party Interactions

CUSAX clients that support the SIP conferencing framework [RFC4353] can detect when a call they are participating in is actually a conference and can then subscribe to conference state updates as per [RFC4575]. A regular SIP user agent might also use the same conference URI for text communication with the Message Session Relay Protocol (MSRP). However, given that SIP's instant messaging capabilities would normally be disabled (or simply not supported) in CUSAX deployments, an XMPP Multi-User Chat (MUC) room [XEP-0045] associated with the conference can be announced/discovered through <service-uris> bearing the "grouptextchat" purpose [I-D.ivov-grouptextchat-purpose]. Similarly, an XMPP MUC room can advertise the SIP URI of an associated service for audio/video interactions using the 'audio-video-uri' field of the "muc#roominfo" data form [XEP-0004] to include extended information [XEP-0128] about the MUC room within XMPP service discovery [XEP-0030]; see [XEP-0045] for an example. These methods would enable a CUSAX-aware SIP conference server to advertise the existence of an associated XMPP chatroom, and for a CUSAX-aware XMPP chatroom to advertise the existence of an associated SIP conference server.

If a CUSAX client joins the MUC room associated with a particular call, it should not rely on any synchronization between the two. Both the SIP conference and the XMPP MUC room would function independently, each issuing and delivering its own state updates. Hence it is possible that that certain peers would temporarily or permanently be reachable in only one of the two conferences. This would typically be the case with single-stack clients that have only joined the SIP call or the XMPP MUC room. It is therefore important for CUSAX clients to provide a clear indication to users as to the level of involvement of the various participants: i.e., a user needs to be able to easily understand whether a certain participant can receive text messages, audio/video, or both.

At the level of the CUSAX service, it is also possible to enforce tighter integration between the XMPP MUC room and the SIP conference. Permissions, roles, kicks and bans that are granted and performed in the MUC room can easily be imitated by the conference focus/mixer into the SIP call. If, for example, a certain MUC member is muted, the conference mixer can choose to also apply the mute on the media stream corresponding to that participant. However, the details and exact level of such integration are entirely up to implementers and service providers.

The approach above describes one relatively lightweight possibility of combining SIP and XMPP multi-party interaction semantics without requiring tight integration between the two. As with the rest of this document, this approach is by no means normative. Implementations and future documents may define other methods or provide other suggestions for improving the unified communications user experience in cases of multi-user chats and conference calling.

5. Federation

In theory there are no technical reasons why federation (i.e., inter-domain communication) would require special behavior from CUSAX clients. However, it is worth noting that differences in administration policies may sometimes lead to potentially confusing user experiences.

For example, let's say atlanta.example.com observes the CUSAX policies described in this document. All XMPP users at atlanta.example.com are hence configured to have vCards that match their SIP identities. Alice is therefore used to making free, high-quality SIP calls to all the people in her roster. Alice can also make calls to the PSTN by simply dialing numbers. She may even be used to these calls being billed to her online account so she would be careful about how long they last. This is not a problem for her since she can easily distinguish between a free SIP call (one that she made by calling one her roster entries) from a paid PSTN call that she dialed as a number.

Then Alice adds xmpp:bob@biloxi.example.com. The Biloxi domain only has an XMPP service. There is no SIP server and Bob uses an XMPP-only client. However, Bob has added his mobile number to his vCard in order to make it easily accessible to his contacts. Alice's client would pick up this number and make it possible for Alice to start a call to Bob's mobile phone number.

This could be a problem because, other than the fact that Bob's address is from a different domain, Alice would have no obvious and straightforward cues telling her that this is in fact a call to the PSTN. In addition to the potentially lower audio quality, Alice may also end up incurring unexpected charges for such calls.

In order to avoid such issues, providers maintaining a CUSAX service for the users in their domain may choose to provide additional cues (e.g., a service-generated signal that triggers a user interface warning in a CUSAX client, an auditory tone, or a spoken message) indicating that a call would incur unexpected charges.

Another scenario arises when a SIP service allow communication only with intra-domain numbers; here Alice might be prevented from establishing a call with Bob's mobile phone. Providers should therefore make sure that calls to inter-domain numbers are flagged with an appropriate audio or textual warning.

6. Summary of Suggested Strategies

The following strategies are suggested for CUSAX user agents:

1. By default, prefer SIP for audio and video, and XMPP for messaging and presence.
2. Use XMPP for all forms of communication with the contacts from the XMPP roster, with the exception of features that are based on establishing real-time sessions (e.g. audio/video calls), for which SIP should be used.
3. Provide online provisioning options for providers to remotely setup SIP and XMPP accounts so that users wouldn't need to go through a multi-step configuration process.
4. Provide online provisioning options for providers to completely disable features for an account associated with a given protocol (SIP or XMPP) if the features are preferred in another protocol (XMPP or SIP).
5. Present a "Call" option for each roster entry that has a properly set "tel" field in the vCard or equivalent.
6. If the client is provisioned with only a single telephony-capable account, initiate calls immediately upon user request without asking users to indicate an account that the call should go through.
7. If no relationship has been configured or discovered between a SIP account and an XMPP account, and the client is aware of multiple telephony-capable accounts, present the user with the choice of reaching the contact through any of those accounts.
8. If known, indicate the existence of a special relationship between the SIP and XMPP accounts of a CUSAX service.
9. Optionally, present the XMPP connection as an "instant messaging" or a "chat" account and the SIP connection as a "Voice and Video" or a "Telephony" account.

10. Optionally, determine the identity of the audio/video caller and a corresponding XMPP roster entry so that the user could use textual chatting or other forms of communication that require XMPP.
11. Optionally, delay the XMPP connection until after a SIP connection has been successfully registered.
12. Optionally, check for alternative communication methods (SIP addresses advertised over XMPP, and XMPP addresses advertised over SIP).

The following strategies are suggested for CUSAX services:

1. Use online provisioning and configuration of accounts so that users won't need to setup two separate accounts for the CUSAX service.
 2. Use online provisioning so that calling features are disabled for all XMPP accounts.
 3. Ensure that at least one of the vCard "tel" fields for each XMPP user is properly populated with a SIP URI that is reachable through the SIP service.
 4. Optionally, include the "video" tel type for accounts that are capable of handling video communication.
 5. Optionally, provision clients with information indicating that specific SIP and XMPP accounts are related in a CUSAX service.
 6. Optionally, attach a "Call-Info" header with an "impp" purpose to all SIP INVITE messages, so that clients can more rapidly associate a caller with a roster entry and display a "Caller ID".
7. IANA Considerations

This document has no actions for the IANA.

8. Security Considerations

Use of the same user agent with two different accounts providing complementary features introduces the possibility of mismatches between the security profiles of those accounts or features. Two security mismatches of particular concern are:

- o The SIP aspect and XMPP aspect of a CUSAX service might offer different authentication options (e.g., digest authentication for

SIP as specified in [RFC3261] and SCRAM authentication [RFC5802] for XMPP as specified in [RFC6120]). Because SIP uses a password-based method (digest) and XMPP uses a pluggable framework for authentication via the Simple Authentication and Security Layer (SASL) technology [RFC4422], it is also possible that the XMPP connection could be authenticated using a password-free method such as client certificates with SASL EXTERNAL even though a username and password is used for the SIP connection.

- o The Transport Layer Security (TLS) [RFC5246] ciphersuites offered or negotiated on the XMPP side might be different from those on the SIP side because of implementation or configuration differences between the SIP server and the XMPP server. Even more seriously, a CUSAX client might successfully negotiate TLS when connecting to the XMPP aspect of the service but not when connecting to the SIP aspect, or vice-versa. In this situation an end user might think that the combined CUSAX session with the service is protected by TLS, even though only one aspect is protected.

Security mismatches such as these (as well as others related to end-to-end encryption of messages or media) introduce the possibility of downgrade attacks, eavesdropping, information leakage, and other security vulnerabilities. User agent developers and service providers must ensure that such mismatches are avoided as much as possible (e.g., by enforcing common and strong security configurations and policies across protocols). Specifically, if both protocols are not safeguarded by similar levels of cryptographic protection, the user must be informed of that fact and given the opportunity to bring both up to the same level.

Section 5 discusses potential issues that may arise due to a mismatch between client capabilities, such as calls being initiated with costs that are not expected by the end user. Such issues could be triggered maliciously, as well as by accident. Implementers therefore need to provide necessary cues to raise user awareness as suggested in Section 5.

Refer to the specifications for the relevant SIP and XMPP features for detailed security considerations applying to each "stack" in a CUSAX client.

9. References

9.1. Normative References

- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E.

Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, March 2011.

[RFC6121] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence", RFC 6121, March 2011.

9.2. Informative References

[I-D.ivov-groupstextchat-purpose]

Ivov, E., "A Group Text Chat Purpose for Conference and Service URIs in the Session Initiation Protocol (SIP) Event Package for Conference State ", draft-ivov-groupstextchat-purpose-03 (work in progress), June 2013.

[RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

[RFC4353] Rosenberg, J., "A Framework for Conferencing with the Session Initiation Protocol (SIP)", RFC 4353, February 2006.

[RFC4422] Melnikov, A. and K. Zeilenga, "Simple Authentication and Security Layer (SASL)", RFC 4422, June 2006.

[RFC4575] Rosenberg, J., Schulzrinne, H., and O. Levin, "A Session Initiation Protocol (SIP) Event Package for Conference State", RFC 4575, August 2006.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5802] Newman, C., Menon-Sen, A., Melnikov, A., and N. Williams, "Salted Challenge Response Authentication Mechanism (SCRAM) SASL and GSS-API Mechanisms", RFC 5802, July 2010.

[RFC6350] Perreault, S., "vCard Format Specification", RFC 6350, August 2011.

[RFC6914] Rosenberg, J., "SIMPLE Made Simple: An Overview of the IETF Specifications for Instant Messaging and Presence Using the Session Initiation Protocol (SIP)", RFC 6914, April 2013.

- [RFC6993] Saint-Andre, P., "Instant Messaging and Presence Purpose for the Call-Info Header Field in the Session Initiation Protocol (SIP)", RFC 6993, July 2013.
- [RFC7033] Jones, P., Salgueiro, G., Jones, M., and J. Smarr, "WebFinger", RFC 7033, September 2013.
- [XEP-0004] Eatmon, R., Hildebrand, J., Miller, J., Muldowney, T., and P. Saint-Andre, "Data Forms", XSF XEP 0004, August 2007.
- [XEP-0030] Hildebrand, J., Millard, P., Eatmon, R., and P. Saint-Andre, "Service Discovery", XSF XEP 0030, June 2008.
- [XEP-0045] Saint-Andre, P., "Multi-User Chat", XSF XEP 0045, February 2012.
- [XEP-0054] Saint-Andre, P., "vcard-temp", XSF XEP 0054, July 2008.
- [XEP-0128] Saint-Andre, P., "Service Discovery Extensions", XSF XEP 0128, October 2004.
- [XEP-0152] Hildebrand, J. and P. Saint-Andre, "XEP-0152: Reachability Addresses", XEP XEP-0152, September 2013.
- [XEP-0166] Ludwig, S., Beda, J., Saint-Andre, P., McQueen, R., Egan, S., and J. Hildebrand, "Jingle", XSF XEP 0166, December 2009.
- [XEP-0167] Ludwig, S., Saint-Andre, P., Egan, S., McQueen, R., and D. Cionoiu, "Jingle RTP Sessions", XSF XEP 0167, December 2009.
- [XEP-0292] Saint-Andre, P. and S. Mizzi, "vCard4 Over XMPP", XSF XEP 0292, September 2013.

Appendix A. Acknowledgements

This draft is inspired by the "SIXPAC" work of Markus Isomaki and Simo Veikkolainen. Markus also provided various suggestions for improving the document.

The authors would also like to thank the following people for their reviews and suggestions: Sebastien Couture, Dan-Christian Bogos, Richard Brady, Olivier Crete, Aaron Evans, Kevin Gallagher, Adrian Georgescu, Saul Ibarra Corretge, David Laban, Gergely Lukacsy, Murray Mar, Daniel Pocock, Travis Reitter, and Gonzalo Salgueiro.

Brian Carpenter, Ted Hardie, Paul Hoffman, and Benson Schliesser reviewed the document on behalf of the General Area Review Team, the Applications Area Directorate, the Security Directorate, and the Operations and Management Directorate, respectively.

Benoit Claise, Barry Leiba, and Pete Resnick provided helpful and substantive feedback during IESG review.

The document shepherd was Mary Barnes. The sponsoring Area Director was Gonzalo Camarillo.

Authors' Addresses

Emil Ivov
Jitsi
Strasbourg 67000
France

Phone: +33-177-624-330
Email: emcho@jitsi.org

Peter Saint-Andre
Cisco Systems, Inc.
1899 Wynkoop Street, Suite 600
Denver, CO 80202
USA

Phone: +1-303-308-3282
Email: psaintan@cisco.com

Enrico Marocco
Telecom Italia
Via G. Reiss Romoli, 274
Turin 10148
Italy

Email: enrico.marocco@telecomitalia.it

MMUSIC
Internet-Draft
Intended status: Informational
Expires: January 10, 2014

J. Marcon
R. Ejzak
Alcatel-Lucent
July 09, 2013

MSRP over WebRTC data channels
draft-marcon-msrp-over-webrtc-data-channels-00

Abstract

The Real-Time Communication in WEB-browsers (RTCWeb) working group is charged to provide protocols to support direct interactive rich communication using audio, video, and data between two peers' web-browsers. For the support of data communication, the RTCWeb working group has in particular defined the concept of bi-directional data channels over SCTP, where each data channel might be used to transport other protocols, called sub-protocols. This document specifies how the Message Session Relay Protocol (MSRP) can be instantiated as a WebRTC data channel sub-protocol, using the SDP offer/exchange to negotiate out-of-band the sub-protocol specific parameters. Two network configurations are documented: a WebRTC end-to-end configuration (connecting two MSRP over data channel endpoints), and a gateway configuration (connecting an MSRP over data channel endpoint with an MSRP over TCP endpoint).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 10, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Conventions	3
3. Terminology	3
4. WebRTC Data Channels	4
5. End-to-end configuration	5
5.1. Support for SDP-based sub-protocol negotiation	5
5.1.1. SDP syntax	5
5.1.1.1. Channel-specific setup parameters	5
5.1.1.2. Sub-protocol specific attributes	6
5.1.2. Procedures	7
5.1.2.1. Opening a data channel	7
5.1.2.2. Closing a data channel	7
5.2. Support for MSRP data channels	7
5.2.1. Overview	7
5.2.2. MSRP URI	8
5.2.3. Session negotiation	8
5.2.4. Session opening	8
5.2.5. Data sending and reporting	8
5.2.6. Session closing	9
5.3. Support for MSRP File Transfer function	9
6. Gateway configuration	9
7. Security Considerations	10
8. IANA Considerations	10
9. Acknowledgments	10
10. References	10
10.1. Normative References	10
10.2. Informative References	11
Authors' Addresses	12

1. Introduction

The Message Session Relay Protocol (MSRP) [RFC4975] is currently defined to work over TCP connections.

The RTCWeb working group has defined the concept of bi-directional data channels running on top of SCTP/DTLS. Each data channel consists of paired SCTP streams sharing the same SCTP Stream Identifier. Data channels are created by endpoint applications through the WebRTC API, and can be used to transport proprietary or well-defined protocols, which in the latter case can be signaled by the data channel "sub-protocol" parameter, conceptually similar to the WebSocket "sub-protocol". However, apart from the "sub-protocol" value transmitted to the peer, RTCWeb leaves open how endpoint applications can agree on how to instantiate a given sub-protocol on a data channel, whether it be in-band or out-of-band (or both). As an example, the SDP offer generated by the browser includes no channel-specific information.

MSRP is a protocol for transmitting a series of related instant messages in the context of a session. In addition to instant messaging, MSRP can also be used for image sharing or file transfer.

Defining the MSRP as a data channel sub-protocol has many benefits:

- o provide to WebRTC applications a proven protocol enabling instant messaging, file transfer, image sharing
- o integrate those features with other RTCWeb voice and video features
- o leverage the SDP-based negotiation already defined for MSRP
- o allows the interworking with MSRP endpoints running on a TCP connection

This document defines the use MSRP of over WebRTC data channels, where one MSRP endpoint is an MSRP WebRTC application and the other endpoint is either an MSRP WebRTC application or an MSRP TCP application.

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Terminology

This document uses the following terms:

Data channel: A bidirectional channel consisting of paired SCTP outbound and inbound streams.

In-band: transmission through the peer-to-peer SCTP association.

Out-of-band: transmission through the WebRTC signaling path, using JSEP [I-D.ietf-rtcweb-jsep] and the SDP Offer/Answer model [RFC3264].

MSRP data channel: A data channel specifically used to transport the messages of one MSRP session.

Peer: From the perspective of one of the agents in a session, its peer is the other agent. Specifically, from the perspective of the SDP offerer, the peer is the SDP answerer. From the perspective of the SDP answerer, the peer is the SDP offerer.

4. WebRTC Data Channels

This section summarizes how WebRTC data channels work in general.

A WebRTC application creates a data channel via the WebRTC Data Channel API, by providing a number of setup parameters (sub-protocol, label, reliability, order of delivery, priority).

The browser then opens in-band the data channel using the DATA CHANNEL OPEN message defined in [draft-jesup-rtcweb-data-protocol]. This message carries some of the channel-specific parameters passed by the application (sub-protocol, label, reliability, order of delivery).

In case an SCTP association is already established, the browser transmits immediately the DATA CHANNEL OPEN message to the peer, on an unused SCTP stream.

In case no SCTP association is established, the browser triggers for an SDP offer/answer exchange, and sends the DATA CHANNEL OPEN message(s) once the SCTP association is established (i.e. subsequently to the reception of the answer).

The SDP offer generated by the browser is as per [draft-ietf-mmusic-sctp-sdp]. In brief, it contains one m-line for the SCTP association on top of which data channels will run, and one attribute per protocol assigned to the SCTP ports:

```
m=application 54111 DTLS/SCTP 5000 5001 5002
c=IN IP4 79.97.215.79
a=sctpmap:5000 webrtc-datachannel 16
```

```
a=sctpmap:5001 bfcf 2
a=sctpmap:5002 t38 1
```

Note: A WebRTC browser will only create an sctpmap attribute for the webrtc-datachannel protocol, and will not create sctpmap attributes for other protocols such as bfcf or t38. This example shows the hypothetical power of the syntax to support multiplexing of SCTP associations for different protocols on the same DTLS connection.

Note: this SDP syntax does not contain any channel-specific information.

5. End-to-end configuration

This section describes the network configuration where each MSRP endpoint is a WebRTC endpoint, running MSRP over an SCTP/DTLS connection.

5.1. Support for SDP-based sub-protocol negotiation

In the default procedures described in Section 4, the channel-specific parameters are notified in-band to the peer, rather than negotiated with the peer. Also, no mechanism is defined to transmit subprotocol-specific parameters to the peer.

This section defines a means to negotiate channel-specific and subprotocol-specific parameters, using the out-of-band SDP offer/exchange.

5.1.1. SDP syntax

The SDP only contains the declaration of data channels for which an SDP-based negotiation is required, and that are either being created or already opened.

5.1.1.1. Channel-specific setup parameters

For each of these data channels, the SDP lists one attribute line providing the Stream Identifier, sub-protocol, label, reliability, order of delivery, priority.

```
a=webrtc-DataChannel:5000 stream=2;label="channel 2"; \
  subprotocol="file transfer";max_retr=3
```

NOTE: the related SDP syntax has to be imported from version 3 of [draft-ietf-mmusic-sctp-sdp].

This line MUST be replicated without changes in the SDP answer, if the answerer accepts the offered data channel.

This line MUST be replicated without changes in any subsequent offer or answer, as long as the data channel is still opened at the time of offer or answer generation.

The Sub-protocol, label, reliability and order of delivery parameters MUST be equal to those transmitted in-band in the DATA CHANNEL OPEN message. The Stream Identifier MUST be equal to the SCTP Stream Identifier on which the DATA CHANNEL OPEN message is sent.

5.1.1.2. Sub-protocol specific attributes

In the SDP, each data channel declaration MAY also be followed by other SDP attributes specific to the sub-protocol in use. Each of these attributes is represented by one new attribute line, and it includes the contents of a media-level SDP attribute already defined for use with this (sub)protocol in another IETF specification.

Each sub-protocol specific attribute such as "a=accept-types:text/plain" that would normally be used to negotiate an instance of MSRP is replaced with an attribute of the form "a=wdcsa:sctp-port:stream-id original-attribute", where wdcsa stands for "webrtc-DataChannel sub-protocol attribute", sctp-port is the sctp port number assigned for webrtc-DataChannel on the media line, stream-id is the sctp stream id assigned to this instance of MSRP, and original-attribute represents the contents of the MSRP related attribute to be included.

```
a=webrtc-DataChannel:5000 stream=2;label="channel 2"; \  
  subprotocol="MSRP";max_retr=3  
a=wdcsa:5000:2 accept-types:text/plain
```

Thus the attribute "a=wdcsa:5000:2 accept-types:text/plain" specifies that this instance of MSRP on stream id 2 accepts plain text files.

As opposed to the data channel setup parameters, these parameters are subject to offer/answer negotiation.

The same syntax applies to any other SDP attribute required for negotiation of this instance of the sub-protocol.

5.1.2. Procedures

5.1.2.1. Opening a data channel

Opening a data channel is done in-band by the DATA CHANNEL OPEN message. However when the sub-protocol requires an SDP-based negotiation, applications MUST NOT send data on this channel till both SDP negotiation and DATA CHANNEL OPEN message sending are done, which may happen in any order.

When the application creates a new data channel (requiring some sub-protocol specific negotiation), the browser follows in any case a generic behavior:

- o if no SCTP association is established, the browser triggers the SDP negotiation, and sends the DATA CHANNEL OPEN message once the answer is received and the SCTP association initialized.
- o if an SCTP association is established, the browser does not trigger any SDP negotiation but instead immediately sends a DATA CHANNEL OPEN message. The application then initiates a new offer/answer exchange

Note: in this case, as the DATA CHANNEL OPEN message is sent before the offer is created, Stream ID conflicts between offers sent to the peer, and DATA CHANNEL OPEN messages received from the peer should not occur.

The application has the task to complete the browser-generated offer (or answer) with the data channel and subprotocol specific parameters in scope of the SCTP m-line. The browser is expected to ignore those parameters when the completed offer (or answer) is applied locally.

5.1.2.2. Closing a data channel

Closing a data channel is done in-band by the SSN reset mechanism, and does not trigger a new offer/answer exchange.

5.2. Support for MSRP data channels

5.2.1. Overview

This document defines how MSRP can be used as a WebRTC sub-protocol, where the MSRP-related negotiation is done as part of the SDP-based data channel negotiation defined in Section 5.1.1.2.

In this design, the MSRP connection maps to the SCTP association and the port assigned to data channels, and each MSRP session maps to one data channel exactly.

5.2.2. MSRP URI

This document extends the MSRP URI syntax [RFC3986] by defining the new transport parameter value "dc":

```
transport = "tcp" / "dc" / 1*ALPHANUM
```

5.2.3. Session negotiation

Using the syntax `a=webrtc-DataChannel:<port> <param=value>`, the SDP declaration of a given MSRP data channel can include at least all the following well-known parameters:

- o defined in [RFC4975]: "path", "accept-types", "accept-wrapped-types", "max-size"
- o defined in [RFC4566]: "sendonly", "recvonly", "inactive", and "sendrecv"
- o defined in [RFC6135]: "setup"
- o defined in [RFC6714]: "msrp-cema"
- o defined in [RFC5547]: all the parameters related to MSRP file transfer

5.2.4. Session opening

The MSRP session is normally opened by the active MSRP endpoint, which sends an MSRP SEND message (empty or not) to the other MSRP endpoint. The active MSRP endpoint does not use the path attribute to open a transport connection to its peer. Instead, the active MSRP endpoint uses the DataChannel established for this MSRP session by the procedures in Section 5.1. The cema attribute is implicitly associated with every MSRP session using data channel transport.

5.2.5. Data sending and reporting

5.2.6. Session closing

Either endpoint can close the MSRP session by closing the underlying data channel. Closing an MSRP session should not trigger an SDP negotiation.

5.3. Support for MSRP File Transfer function

[RFC5547] defines an end-to-end file transfer method based on MSRP and the SDP offer/answer mechanism. This file transfer method is also usable by MSRP WebRTC endpoints, with the following considerations:

- o As an MSRP session maps to one data channel, a file transfer session maps also to one data channel.
- o SDP attributes specified in [RFC5547] for a file transfer m-line are embedded as subprotocol-specific attributes as defined in Section 5.1.1.2.
- o Each file chunk is transmitted over one SCTP user message.
- o Once the file transfer is complete, the same data channel MAY be reused for another file transfer.
- o Following the aborting of a file transfer, the SDP can be updated by adding the "inactive" attribute to the list of subprotocol-specific attributes associated with the corresponding data channel.

6. Gateway configuration

This section describes the network configuration where one endpoint runs MSRP over a WebRTC SCTP/DTLS connection, the other MSRP endpoint runs MSRP over one or more TLS/TCP connections, and the two endpoints interwork via an MSRP gateway.

Specifically, a gateway can be configured to interwork an MSRP session using a data channel with a peer that does not support data channel transport in one of two ways. In one model, the gateway performs as a MSRP B2BUA to interwork all the procedures as necessary between the endpoints. No further specification is needed for this model.

Alternately, the gateway can use CEFA procedures to provide transport level interworking between MSRP endpoints using different transport protocols as follows.

When the gateway performs transport level interworking between MSRP endpoints, all of the procedures in section Section 5.1 apply to each peer, with the following additions:

- o The endpoint establishing an MSRP session using data channel transport shall not request inclusion of any relays, although it may interoperate with a peer that signals the use of relays.
- o The gateway receiving an SDP offer that includes a request to negotiate an MSRP session on a data channel can provide transport level interworking in the same manner as a CEMA SBC by forwarding TCP or TLS transport parameters in a new m line with the appropriate attributes within the forwarded SDP offer.
- o Similarly, a gateway receiving an SDP offer to negotiate an MSRP session using TCP or TLS transport with an endpoint that only supports data channel transport for MSRP can provide transport level interworking in the same manner as a CEMA SBC by establishing a new data channel for the MSRP session with the target endpoint.

7. Security Considerations

To be completed.

8. IANA Considerations

To be completed.

9. Acknowledgments

The authors wish to thank... for their invaluable comments.

10. References

10.1. Normative References

- [I-D.ietf-rtcweb-jsep]
Uberti, J. and C. Jennings, "Javascript Session Establishment Protocol", draft-ietf-rtcweb-jsep-02 (work in progress), October 2012.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

- [RFC4975] Campbell, B., Mahy, R., and C. Jennings, "The Message Session Relay Protocol (MSRP)", RFC 4975, September 2007.
- [RFC4976] Jennings, C., Mahy, R., and A. Roach, "Relay Extensions for the Message Sessions Relay Protocol (MSRP)", RFC 4976, September 2007.
- [RFC5547] Garcia-Martin, M., Isomaki, M., Camarillo, G., Loreto, S., and P. Kyzivat, "A Session Description Protocol (SDP) Offer/Answer Mechanism to Enable File Transfer", RFC 5547, May 2009.
- [RFC6135] Holmberg, C. and S. Blau, "An Alternative Connection Model for the Message Session Relay Protocol (MSRP)", RFC 6135, February 2011.
- [RFC6714] Holmberg, C., Blau, S., and E. Burger, "Connection Establishment for Media Anchoring (CEMA) for the Message Session Relay Protocol (MSRP)", RFC 6714, August 2012.

10.2. Informative References

- [I-D.ietf-rtcweb-data-channel] Jesup, R., Loreto, S., and M. Tuexen, "RTCWeb Data Channels", draft-ietf-rtcweb-data-channel-04 (work in progress), February 2013.
- [I-D.jesup-rtcweb-data-protocol] Jesup, R., Loreto, S., and M. Tuexen, "WebRTC Data Channel Protocol", draft-jesup-rtcweb-data-protocol-04 (work in progress), February 2013.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.
- [WebRtcAPI] Bergkvist, A., Burnett, D., Narayanan, A., and C. Jennings, "WebRTC 1.0: Real-time Communication Between Browsers", World Wide Web Consortium WD WD-webrtc-20120821, August 2012, <<http://www.w3.org/TR/2012/WD-webrtc-20120821>>.

Authors' Addresses

Jerome Marcon
Alcatel-Lucent
Route de Villejust
Nozay 91620
France

Email: jerome.marcon@alcatel-lucent.com

Richard Ejzak
Alcatel-Lucent
1960 Lucent Lane
Naperville, Illinois 60563-1594
US

Phone: +1 630 979 7036

Email: richard.ejzak@alcatel-lucent.com