Building a House of Cards: On the Intrinsic Challenges of Evolving Communication Standards

Jean-Charles Grégoire
INRS-EMT, CANADA
email:gregoire@emt.inrs.ca

***Abstract*—A critical look at a recent communications standard exposes how matters of feature interaction remain pervasive, not necessarily at a horizontal level, but also through layers of the architectures. This situation arises from the emergence of new standards or popular services based on earlier infrastructures, generic support middleware as well as emerging technologies. Several illustrations of such problems are presented and discussed in this paper, as illustrations of problems to try to avoid in practice. We show how the industrial practices remain lacking but also how some of the difficulties around the emergence of feature interactions are deeply linked to the standardization process.**

Keywords–Feature Interaction; RCS; SIP; OMA CPM; OMA SIMPLE IM; IMS.

I. INTRODUCTION

Much has been written over the years on the problem of feature interaction (FI) and its numerous guises [1], [2], [3], [4], [5], [6]. Different communities, such as requirements engineering and formal methods have come together to present various perspectives on issues pertaining to the specification and implementation of telecommunication services, and further expanding such investigations beyond telephony to other domains, even besides telecommunications. Tools, architectures, methods and insights has been proposed, with varying, yet demonstrated degree of usefulness. At least two questions remain at this stage: what practical impact can we observe from this work and is there still "something" that we have missed.

The fairly recent 5.1 release of the Rich Communications System (RCS) standard [7] has been an opportunity for us to look at these questions. In this paper, we look at a number of issues we perceive as problems related with this specification, and later discuss how a feature interaction perspective may help with such problems, or where more effort is warranted. We begin with giving some background on messaging protocols used in cellular communications. We will then follow with the description of a number of issues of an FI nature raised by the current specification and expand into a more general discussion.

Because of the nature of this special issue, we expect that the reader is familiar with the Session Initiation Protocol (SIP) and the work done on that foundation by different bodies. The introduction only highlights the contributors but not the technology itself. Among many possibilities, references [8], [9] can be used if such information is desired.

The paper is structured as follows. The next Section gives some background on the standards of interest. Section 3 presents and illustrates a number of problems. Section 4 is a general discussion and Section 5 concludes this paper.

II. BACKGROUND

In the age of the Internet, different bodies contribute to the standardization process. While the IETF has created the Session Initiation Protocol and retains the control over its evolution, it has become the foundation of several service infrastructures, notably the IP Multimedia Subsystem (IMS), originally developed by the 3rd Generation Partnership Project (3GPP) for wireless services. Over the years, other standardization bodies have become interested in IMS which built into joint work with 3GPP2, the European Telecommunications Standards Institute (ETSI) and CableLabs. Practically, this has led to the extension of the use of IMS in 3GPP to various network access technologies, and their evolution. While 3GPP was concerned with the middleware, it was not interested in pursuing work on applications beyond audio/video (A/V) services. Other bodies, possibly closer to the ear of operators, have looked into other services, such as the Open Mobile Alliance (OMA) for wireless. Over time OMA focus has moved from being platform neutral to acknowledging IMS as an enabling platform. OMA has produced two standards for personal communications based on SIP, among other protocols: Instant Messaging (IM)[11] and Converged IP Messaging (CPM) [12]. The GSM Alliance (GSMA) has in turn reused and extended some of that work to create the current (5th) release of RCS. As companion to the standard, we find a number of *endorsement documents* for a variety of messaging communication standards which describe to what extent their functions are supported (e.g OMA SIMPLE IM 2.0, OMA CPM 2.0).

Why such a complex picture? While a division of responsibilities may appear clear between IETF and 3GPP, matters become more complicated when competition over forums, markets and cultures emerge. Tensions between different perspectives abound while efforts are constantly made to bind them together in a convergent view–pick your favourite analogy of the One Ring or the Holy Grail–to have as large a market as possible. GSMA, for example, takes upon the mandate to look at standards proposed by major agencies, extracts from complex architectures or a large selection of protocols a set of mechanisms sufficient to support a set of services of interest to its members, in a streamlining process, which in essence does not create anything new except of specific focus for service enabling technology, with different profiles.

This picture explains how the standardization process remains complex, even though quite often the same companies

 ISBN: 978-1-61208-330-8

will be involved with different aspects of the work. Since the devil remains in the details, we see that the traditional issue of going through piles of documents written in prose ornamented with diagrams has not changed. A quick glance at the RCS specification from GSMA reveals a typical document structure with general concepts, feature specifications and use cases. In traditional fashion, we find text describing interactions between the different features of the service, even for the most trivial cases, to. viz.:

> 3.2.2 Interaction with other RCS features
> There are no interactions between the RCS 5.1 Standalone Messaging service and other RCS services.

(Rich Communication Suite 5.1[7] p. 138.)

The questions we raise are the identification of interactions arising from this piecemeal construction of a standard, constrained by the reuse of existing components designed in different fora because of a number of constraints, not the least commercial ones, but also historical, as our last example shall illustrate.

III. Problems

In this section, we illustrate a number of issues of an FI nature with the 5.1 revision of the RCS standard. This is done informally, based on quotes from the document itself and a discussion thereof.

A. Problem 1

Recall that a REGISTER message is a pre-requisite to SIP operations, to bind the user to a proxy (or P-CSCF for IMS) and allow further end-to-end communications; it is essentially the first operation to be performed by the user to allow access to services. An INVITE will initiate a communication. Figure 1 illustrates the process.

In RCS, there are three different ways to send a message: through a SIP MESSAGE message, by initiating a session using the MSRP [13] protocol, or directly within the SIP INVITE message, with a CPIM body. Again, these alternatives stem from the consideration of different communication scenarios in different standards, conversational or standalone messages, and also message size—in the standards, we see different modes of operation for standalone messages: pager mode or large message mode. Furthermore, non-delivered messages are stored in a server and delivered at the next registration. The need to be able to hold messages requires the presence of a server able to temporarily store messages in the path. The server is informed of the registration through a notification and automatically sends the pending messages.

Now let us consider this segment of the RCS specification.

> **2.4.5 Registration frequency optimization**
> *An RCS client shall not send more register requests than what is needed to maintain the registration state in the network. When the IP connectivity is lost and restored with the same IP address, the RCS client shall:*
> - *Only send a register refresh upon retrieval of IP connectivity if the duration for sending a register-refresh since the last REGISTER request has been exceeded, and,*
> - *Only send an initial register upon retrieval of IP connectivity if the registration has expired.*

(RCS 5.1 Advanced Communications Specification [7], p. 55.)

There are several issues here. Again, going back to figure 1, we see that the access link can be of various natures, including WiFi. Depending on the technology used, the loss of the communication link may or may not be detected automatically and notified to IMS. In some way, detection will boil down to the use of a form of timeout but the duration of that timeout is key: it goes from milliseconds in some cases to seconds in others. The matter is worst when timeout detection is done through the loss of the TCP carrier.

We intuitively see that this situation leads to a race condition, where a user can be disconnected and reconnected while a TCP session is still established and this can lead to out-of-order messages delivery if new messages are sent while a TCP connection still holding messages was ended, not unlike what we observe with email when messages cannot be delivered and are retransmitted at a later time. This could occur because of unstable access, typically WiFi, and some messages being stored until they can be delivered.

B. Problem 2

As a corollary of the previous problem, we might wonder what the issue is with the TCP protocol that standard bodies seem reluctant to use it. As we have seen, rather than relying the MSRP protocol for instant communications, we find a number of alternative behaviours.

At stake there can be the compatibility between an Internet-like messaging (SIP-independent) behaviour and an SMS-like behaviour. In a typical Internet-like, IM behaviour, a session is established with a server and we receive notifications when a message is received for us, typically through polling mechanisms. TCP is the underlying transport mechanism and guarantees the stability of the session. Alternatively, an SMS can be received or sent independently of a service session. This latter behaviour has led to the creation of short messages in OMA standards: a message can be carried in an invite and a session—complete with MSRP transport—will be established when a message is sent back.

This behaviour actually alleviates some issues. For one, *if the target of the message is connected through multiple terminals, which one will actually be used for the exchange?* Only when an answer is sent can this be safely assessed, and a unique TCP session established with a full SIP INVITE exchange. Of course, it could be argued that a typical SIP session could instead be used. But, if the user has registered multiple terminals, and one or more are auto-answered enabled, this could lead to the set-up of multiple connections, possibly with the split of the TCP session at a message relay in the call, and increased costs in resource usage and bookkeeping. Practically, different markets have chosen different solutions and a global standard must support them all.

 ISBN: 978-1-61208-330-8

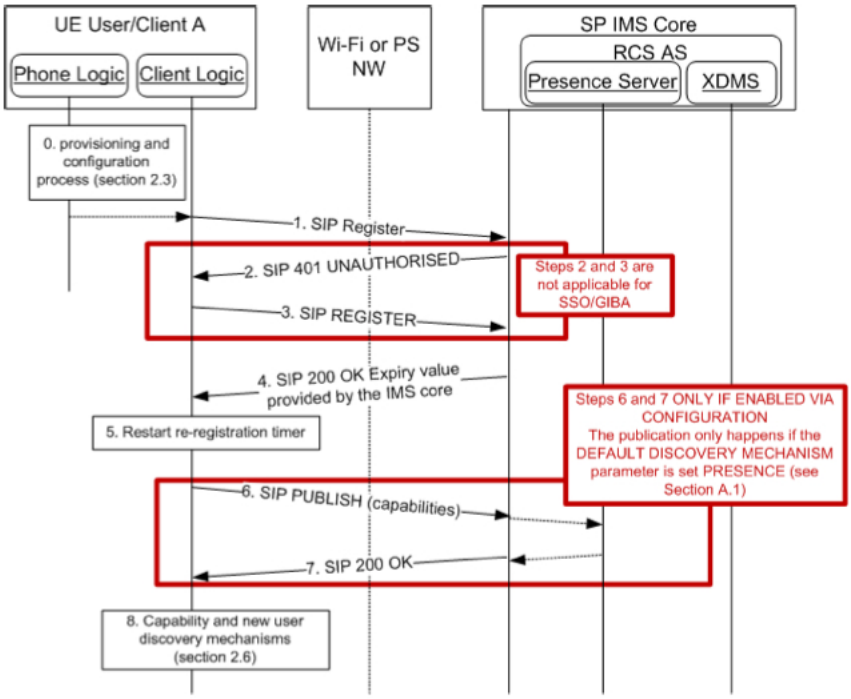

Figure 1. The SIP registration process, from [7].

C. Problem 3

Another corollary of the existence of various forms of messages is the intersection of the signalling path and the media path. Since messages are carried in signalling messages or on MSRP (TCP) sessions and since messages must be stored, if necessary, in a message store and possibly be forwarded to different destinations, there are nodes which must be both on the signalling path and on the media path.

The following quotation illustrates how these matters are acknowledged in standards. In practice, the CPM Feature Tag is used to route the SIP messages to the proper functional components.

> The protocols used for the CPM-PF1 interface are SIP, SDP, MSRP, and RTP/RTCP. SIP is used for CPM Session signalling, for CPM File Transfer signalling and for discrete Pager Mode CPM Standalone Message transfer. SDP is used to describe the set of Media Streams, codecs, and other Media related parameters supported during CPM Session set up and for describing file characteristics during CPM File Transfer initiation. MSRP is used for the transfer of Large Message Mode CPM Standalone Messages, for the exchange of CPM Chat Messages, both small and large, and for the Media transfer of a CPM File Transfer. RTP is used for continuous Media transport and RTCP supports for the exchange of information needed to control RTP sessions.
>
> NOTE: The exact network path used for the actual Media transfers (i.e., MSRP and RTP/RTCP protocols) will be negotiated via the SIP signalling part of this interface. For example, it is possible that direct client-to-client Media transfers are negotiated, or a direct Media transfer between a client and an Interworking Function. The signalling part of the CPM-PF1 interface is dependent on an underlying SIP/IP core infrastructure.
>
> (OMA CPM standard, [12] p. 30.)

For the sake of completeness, let us just add that the choice of MSRP as a support for standalone message transfer (large message mode) depends solely on the size of the message to be sent. Above 1300 bytes, an MSRP session is established and disconnected once the transfer has been accomplished.

D. Problem 4

Another acknowledged challenge has been the proliferation of the means to identify users and the means to access them. The two following quotations from the RCS specification clearly illustrate this. In the first case, we see the link between access technology and the user identity. What is interesting here is that we could assume that such problems are resolved by IMS, which should provide a unique means of identification. With different access technologies and authentication requirements, we end up with a proliferation of such information, and possible inconsistencies. In the second case we see the necessity of being able to use an identity users are still very familiar with, such as a telephone number (tel URI), in a typical sign of legacy constraint.

> Both a SIP and a tel URI may be configured for a user with following clarifications:
>
> - The configured values should not be used in the non-REGISTER transactions; instead the client uses one of the SIP or tel URIs provided in the P-Associated-URI header field

returned in the 200 OK to the SIP REGISTER request as described in [18].
- The user's own tel URI and/or SIP URI identities are configured through the Public_User_Identity parameters defined in [18].
- The public identity used for IMS registration is built according to the procedure defined in [18].
- When the device has either ISIM or USIM present and the RCS client has access to the ISIM or USIM, it does not rely on the SIP URI and tel URI configuration parameters.
- If the device has neither ISIM nor USIM present or is not able to access to it, a SIP URI must be configured. This URI is used for REGISTER transactions.
- Configuration of the tel URI is optional.

(RCS 5.1 Advanced Communications Specification, Version 1.0, [7] p. 332.)

> For device incoming SIP requests, the address(es) of the contact are, depending on the type of request, provided as a URI in the body of a request or contained in the P-Asserted-Identity and/or the From headers. If the P-Asserted-Identity header is present, the From header will be ignored. The only exception to this rule is when a request for Chat or Standalone Messaging includes a Referred-By header (it is initiated by Messaging Server for example in a store and forward use case as described in 3.3.4.1.4), thereby the Referred-By header should be used to retrieve the originating user instead. The receiving client will try to extract the contact's phone number out of the following types of URIs:
>
> - tel URIs (telephone URIs, for example tel:+1234578901, or tel:2345678901;phone-context=*phonecontextvalue*)
> - SIP URIs with a "user=phone" parameter, the contact's phone number will be provided in the user part (for example sip:+1234578901@operator.com;user=phone or sip:1234578901;phone-context=*phonecontextvalue* @operator.com;user=phone)
>
> Once the MSISDN is extracted it will be matched against the phone number of the contacts stored in the Address Book. If the received URI is a SIP URI but does not contain the "user=phone" parameter, the incoming identity should be checked against the SIP and tel URI address of the contacts in the address book instead. If more than one P-Asserted-Identity is received in the message, all identities shall be processed until a matched contact is found.

(RCS 5.1 Advanced Communications Specification, Version 1.0 [7], p. 57.)

E. Problem 5

As terminal technology evolves, traditional, basic assumptions on capabilities and capability negotiation need to be reassessed. It used to be that terminals had simple capabilities and a support service, for example for MMS, could resolve conversion issues simply based on the identification of the terminal and its profile. Nowadays, smartphones will support not only different codecs but also different formats and profiles. Current IETF work [14] reflects this trend and defines extensions to SDP to support new multimedia capabilities but one may wonder if all communication features can be supported in such a mode of negotiation, or what manufacturers do while waiting for the IETF to update its standards accordingly. For example, Apple's FaceTime supports switching video transmission between portrait and landscape orientation, and will notify the receiver side of the proper orientation, which will automatically trigger the appropriate view change for the receiver. This quite useful and rather trivial extension was not reflected in standard communication specifications until RCS 5.1 and has since then pushed into 3GPP.

F. Problem 6

If we consider capability exchange in RCS, that is, which features the terminal or terminal application supports, we see the combination of two approaches, one which assumes that a presence server is used in the network, which follows the OMA Presence specification, and one where terminals will exchange capability end to end. We should note that the use of SIP OPTIONS for such end to end discovery of communication features conforms with the IETF (SIP) RFC 3261, although the purpose is slightly different. The presence server will hold the information of features supported by the multiple terminals a user may have active at some time. A communication attempt can be made based on the features identified based on the query of the server, and will be forked only to the terminals supporting those features, again as per RFC 3841[15].

However, if one resorts to using SIP OPTIONS capability exchange, i.e., in the absence of a Presence server, the target device of the user cannot be selected a priori and the request message has to be forked to all terminals by the terminating CSCF (acting as a SIP proxy server, following the trapeze model.) The first terminal answering will establish itself as the terminating end, but it may not support the features required for the call, and the full set of capabilities available will not be returned, which may result in a communication attempt not being made.

To avoid such a situation, a new application server has to be introduced, in this case the Options AS. ([7], p. 59.) We must note two things about this server. First, it is specified only implicitly in the document, as it is supposed to make sure all the SIP OPTIONS based call flow behave as required. Second, its presence in a network is only required if the Presence server cannot be used. Still we can see why some manufacturers/operators would have preferred one approach vs. the other. Keeping the decision closer to the end terminal allows adaptation to dynamic access conditions, which is harder to achieve when such information must be updated on a remote—here presence—server.

G. Problem 7

Although feature tags are used in accordance with IETF requirements, their interpretation differs slightly as RCS uses them to indicate to the network and other devices the set of

 ISBN: 978-1-61208-330-8

communication methods used by the device, whereas they are meant to help route calls to suitable terminals in RFC 3841.

While the end effect may end up being the same for the user, there are interactions at work here between the information users (and their applications) want, and the needs of the network, specifically in terms of accounting and ultimately billing. Whereas the user application would be happy to indicate that it supports RCS communications, an operator would appreciate having a break down per specific features, e.g., file transfer or video transfer, for content-specific billing. Again, for historical reasons, RCS supports feature tags of both natures, more detailed and more specific.

IV. Discussion

We have presented matters of conflicting and evolving requirements, consensus building, changing context, lack of proper architectural constructs. Some are typical Feature Interaction matters, others might not be considered as such. Overall, such issues are not new, but we see them occurring in new standards and this leads to wonder about the impact of research on feature interaction on standards work.

A. On Standards

An important concern we have attempted to expose here is *evolution*. Evolution of standards, of course, but also evolution of technology. In some cases, we could feel as if the rug had been pulled from under our feet as an issue which appeared resolved is reopened, because new uses are added, or the underlying technology changes. Formally, this can be captured as a matter of machine-closed-ness [8], [9]. For specifications to be machine-closed means that no liveness property imposes restriction on finite behaviours of the system. Once the implementation changes, such a condition may no longer be satisfied and elements of the proof of implementation be broken; if we integrated two bodies of technology into one to support different legacies, we must handle possible assumption mismatch. How we, in practice, detect such conditions in an evolving environment largely remains an open issue, even if this is hardly a new problem, as it can be related to the 1996 Ariane 5 flight 501 failure; see for example [10].

Evolution in standards, context and usages are not coordinated. The terminal changes, and user interfaces will take advantage of it, while application protocols, standardized separately, will evolve separately and introduce limitations, or create issues where none existed before. Different markets will make different decisions on the evolution of service offerings, on matter as diverse as the path to obsolescence for SMS and its transition to IP-based texting. This in turns has an impact on specifications such as RCS. In practice, we have only seen such coordination succeed in situations where a single party controls most of the application and the infrastructure, as is becoming common in cloud-based environments such as offered by Google or Facebook. Interoperability, then, is not a concern. These applications, globally designated as “over the top” by operators, restrict them to a role of carriers without added value and are rather not viewed too fondly by the latter.

B. On Race Conditions

An area, however, where we could do much better is in the explicit capture of the semantics of connectivity and their report, especially failure semantics. It was quite surprising to realize how standards are still missing a clear way to define and report connectivity to applications, consistently across both signalling and media path. That we still find ourselves confronted with matters of race conditions and reordering in these days and age is quite amazing. Sadly, the trend will be to try to patch the problem, and not to go to its core, as we keep on building the house of cards.

This state of affairs also warrants a deeper look. At stake here is the end-to-end conception of communications services, as supported by IMS, and as opposed to a centralized model of control. Most popular communication services we use nowadays over the Internet are based on a model of cloud-based, client server-like centralized control, where is it possible to coordinate sender and receiver through a unique relay. In such circumstances, the effects of temporary disconnection can be easily managed, simply because the client will have the responsibility of querying the server for any new messages. In a straightforward peer-to-peer model, it is also possible to avoid such issues if we operate under a single domain/application (e.g., Skype) and the client can take the responsibility of holding on to messages and queueing them in proper delivery order until their delivery has been confirmed.

Race condition situations arise as domains are split and each domain takes responsibility for its part of the transaction: sending or receiving. On the receiver side, another entity must be introduced to temporarily store messages and therefore adds a further communication path to the receiver. The SIP call model, through its forking mechanism, easily supports placing a server on a call which will accept incoming communications if the user devices cannot and such addition is trivial. The challenge is then to deliver messages which may come directly from the sender and from storage in the right order: the store will need to be informed of the renewed connectivity before it can transmit the message, while the sender can send new messages directly to the receiver and they may arrive out of order. This race condition cannot be resolved unless the client polls the server first after re-establishing its connectivity, but this is not part of the SIP model and puts more demands on the applications. Also, in the IMS model, the server could be systematically put on the signalling path, but to function properly would need to completely intercept the call, i.e., be a back to back user agent (B2BUA), which would break the semantics of the call.

To summarize, we see that race conditions become a side effect of a forcing some features on top of a signalling model which is not fully adequate. Strict adherence to an end-to-end model without intermediate storage would resolve this issue, but then put more complexity in the client. But this leads to other philosophical debates.

We should mention a related approach [19], which we have described in earlier work, that would end the call on a form of user avatar, a client virtualized in a cloud at the edge of the operator’s domain (the edge cloud). It would act as be a stable point for communications while a simple, streamlined GUI protocol would run on the access link, as a form of compromise

 ISBN: 978-1-61208-330-8

between the multi-domain and centralized solutions.

V. CONCLUSION

We have presented a number of problems with a contemporary telecommunications standard, to illustrate how, beyond the progress we have made in requirements engineering and formal analysis we still have work to do as a community to improve the industrial state of the art.

We have illustrated how many of these issues are not really new nor ground breaking in nature. Solutions for them do exist or, in other cases, the nature of the issue can be identified and diagnosed before it makes its way into standards, or worse into implementations. Still, while the cure to the issues may be clear, we still need to better understand their cause and it is our hope that this paper can serve as a salutary lesson in that respect.

REFERENCES

[1] M. Calder, M. Kolberg, E.H. Magill, and S. Reiff-Marganiec, "Feature Interaction: A Critical Review and Considered Forecast", Computer Networks, Vol. 41, Jan. 2003, pp. 115–141.

[2] K.J. Turner and E.H. Magill, Eds. "Feature Interaction in Communications and Software Systems", Computer Networks, Special Issue on Feature Interaction, Vol. 57, Aug. 2013, pp. 2395–2464.

[3] M. Jackson and P. Zave, "Distributed Feature Composition: A Virtual Architecture for Telecommunications Services", Software Engineering, IEEE Transactions on, Vol. 24, Oct. 1998, pp. 831–847.

[4] M. Kolberg and E.H. Magill, "Managing feature interactions between distributed SIP call control services", Computer Networks: Special Issue on Feature Interaction, Vol. 51, Feb. 2007, pp. 536–557.

[5] A. Nhlabatsi, R. Laney, and B. Nuseibeh, "Feature interaction: the security threat from within software systems", Progress in Informatics, Special issue: The future of software engineering for security and privacy, Mar. 2008, pp, 75–89.

[6] A. Gouya and N. Crespi, "Detection and resolution of feature interactions in IP multimedia subsystem", Intl. Journal of Network Management, Vol. 19, Jul/Aug. 2009, pp. 315–337.

[7] GSM Association, Rich Communication Suite 5.1 Advanced Communications Services and Client Specification, Version 1.0, 13 August 2012.

[8] M. Poikselka and G. Mayer, "The IMS : IP Multimedia Concepts and Services", J. Wiley, 3rd ed., 2009.

[9] R. Noldus et al., "IMS Application Developer's Handbook: Creating and Deploying Innovative IMS Applications", Academic Press, 2011.

[10] J. L. LIONS, "ARIANE 5: Flight 501 Failure Report", the Inquiry Board, Paris, 19 July 1996, available from http://www.ima.umn.edu/ arnold/disasters/ariane5rep.html, last accessed February 14th, 2014.

[11] OMA SIMPLE IM V2.0, Open Mobile Alliance, Release date (Candidate Version), 2012-07-31.

[12] Converged IP Messaging Architecture V2.0, Open Mobile Alliance, Release Date (Candidate Version) 2013-06-11.

[13] B. Campbell, R. Mahy, and C. Jennings, The Message Session Relay Protocol (MSRP), RFC 4975, IETF, 2007.

[14] J. Rosenberg, H. Schulzrinne, and P. Kyzivat, Indicating User Agent Capabilities in the Session Initiation Protocol (SIP), RFC 3840, IETF, August 2004.

[15] J. Rosenberg, H. Schulzrinne, and P. Kyzivat, Caller Preferences for the Session Initiation Protocol (SIP), RFC 3841, IETF, August 2004.

[16] L. Lamport, The Temporal Logic of Actions, ACM Transactions on Programming Languages and Systems (TOPLAS), Vol. 16, May 1994, pp. 872–923.

[17] M. Abadi and L. Lamport, The existence of refinement mappings, Theoretical Computer Science, May 1991, pp. 253–284.

[18] Third Generation Partner Project (3GPP), IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP), TS 24.229, Release 11, 2012.

[19] S. Islam and J.-Ch. Grégoire, Giving users an edge: A flexible Cloud model and its application for multimedia, Future Generation Computer Systems, June 2012, pp. 823–832.

 ISBN: 978-1-61208-330-8