Push using SIP

Candidate Version 1.0 – 01 Apr 2010

Open Mobile Alliance

OMA-TS-SIP_Push-V1_0-20100401-C

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.

Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.   **[OMA-Template-Spec-20100101-I]**

Contents

Figures

Tables

1. Scope

This specification defines the protocol for delivery of content to a mobile terminal via SIP [RFC3261] and MSRP [RFC4975], referred to as SIP Push. The protocol specified in this document is an application layer protocol that interacts with a SIP/IP Core network, such as IMS [IMSArch] to fulfil the Push using SIP service.

The scope of this specification is to specify mechanisms for the following functionality within the architecture identified in [PushSIPArch]

2. References

2.1 Normative References

[3GPP TS 23.228]	"IP Multimedia Subsystem (IMS); Stage 2", 3GPP TS 23.228
[3GPP TS 33.203]	"Access Security for IP-based services", 3GPP TS 33.203
[3GPP TS 33.210]	"Network domain security; IP network layer security", 3GPP TS 33.210
[3GPP TS 24.229]	"Internet Protocol (IP) multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3", 3GPP TS 24.229
[3GPP2 X.S0013-002-A]	"All-IP Core Network Multimedia Domain: IP Multimedia Subsystem - Stage 2", Revision A, Version 2.0, 3GPP2, 2004
[3GPP2 X.S0013-004-A]	"All-IP Core Network Multimedia Domain: IP Multimedia Call Control Protocol Based on SIP and SDP Stage 3", Revision A, Version 2.0, 3GPP2, 2004
[3GPP2 S.R0086-0]	"IMS Security Framework", 2004
draft-drage-sipping-service-identification	"A Session Initiation Protocol (SIP) Extension for the Identification of Services". NOTE: The referenced IETF draft is a work in progress.
[draft-ietf-sip-gruu]	Obtaining and Using Globally Routable User Agent (UA) URIs (GRUU) in the Session Initiation Protocol NOTE: The referenced IETF draft is a work in progress.
[draft-ietf-sipping-asserted-id]	Updates to Asserted Identity in the Session Initiation Protocol (SIP); draft-ietf-sipping-update-pai-02.txt NOTE: The referenced IETF draft is a work in progress.
[IOPPROC]	"OMA Interoperability Policy and Process", Version 1.1, Open Mobile Alliance™, OMA-IOP-Process-V1_1, URL:http://www.openmobilealliance.org/
[OMA-UAProf]	"User Agent Profile", OMA-TS-UAProf-V2_0-20060206-A. URL: http://www.openmobilealliance.org/
[OMNA]	"OMA Naming Authority". Open Mobile Alliance™. http://www.openmobilealliance.org/tech/omna
[RFC5547]	"A Session Description Protocol (SDP) Offer/Answer Mechanism to Enable File Transfer", Garcia-Martin, M., Ed., Isomaki, M, Camarillo, G, Loreto, S, and Kyzivat, P, May 2009, URL:http://www.ietf.org/rfc/rfc5547.txt
[RFC4975]	"The Message Session Relay Protocol (MSRP)", Campbell, B., Ed., Mahy, R., Ed., and C. Jennings, Ed.,, September 2007, URL:http://www.ietf.org/rfc/rfc4975.txt
[RFC4976]	"Relay Extensions for the Message Session Relay Protocol (MSRP)", Jennings, C., Mahy, R., and A. Roach,, September 2007, URL:http://www.ietf.org/rfc/rfc4976.txt
[RFC2119]	"Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997, URL:http://www.ietf.org/rfc/rfc2119.txt
[RFC2183]	"Communicating Presentation Information in Internet Messages: The Content-Disposition Header Field." URL:http//www.ietf.org/rfc/rfc2183.txt
[RFC2234]	"Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell. November 1997, URL:http://www.ietf.org/rfc/rfc2234.txt
[RFC2506]	"Media Feature Tag Registration Procedure". Holtman et al. M1999. URL:http://www.ietf.org/rfc/rfc2506.txt
[RFC2616]	"Hypertext Transfer Protocol -- HTTP/1.1", Fielding et al, June 1999, URL:http://www.ietf.org/rfc/rfc2616.txt
[RFC3261]	"SIP: Session Initiation Protocol". J. Rosenberg et al. June 2002. URL:http://www.ietf.org/rfc/rfc3261.txt

[RFC3264]	"An Offer/Answer Model with the Session Description Protocol (SDP)". Rosenberg et al. URL:http://www.ietf.org/rfc/rfc3264.txt
[RFC3325]	"Private Extensions to the Session Initiation Protocol", C. Jennings, J. Peterson, M. Watson, Nov. 2002, URL:http://www.ietf.org/rfc/rfc3325.txt
[RFC3428]	"Session Initiation Protocol (SIP) Extension for Instant Messaging". B. Campbell et al. December 2002. URL:http://www.ietf.org/rfc/rfc3428.txt
[RFC3680]	"A Session Initiation Protocol (SIP) Event Package for Registrations", Rosenberg, J.,, March 2004, URL:http://www.ietf.org/rfc/rfc3680.txt
[RFC3840]	"Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)".J. Rosenberg, H.Schulzrinne, P.Kyzivat, Aug. 2004. URL:http://www.ietf.org/rfc/rfc3840.txt
[RFC3841]	"Caller Preferences for the Session Initiation Protocol (SIP)".J. Rosenberg, H.Schulzrinne, P.Kyzivat, Aug. 2004. URL:http://www.ietf.org/rfc/rfc3841.txt
[RFC4145]	"TCP-Based Media Transport in the Session Description Protocol (SDP)", September 2005, URL:http://www.ietf.org/rfc/rfc4145.txt
[RFC4483]	"A Mechanism for Content Indirection in Session Initiation Protocol (SIP) Messages", E. Burger, Ed., May 2006, URL:http://www.ietf.org/rfc/rfc4483.txt
[RFC4566]	"SDP: Session Description Protocol", July 2006, URL:http://www.ietf.org/rfc/rfc4566.txt

2.2   Informative References

[IMSArch]	"Utilization of IMS capabilities Architecture", OMA-AD-IMS-V1_0-20050204-C URL:http://www.openmobilealliance.org/
[PushMsg]	"Push Message Specification". WAP Forum™. WAP-251-PushMessage. URL: http://www.openmobilealliance.org/
[PushSIPArch]	"OMA-AD-SIP_Push_AD-V1_0-20071203-D ", Open Mobile Alliance™. URL:http//www.openmobilealliance.org/

3. Terminology and Conventions

3.1 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].

All sections and appendixes, except "Scope" and "Introduction", are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

Application	An implementation of a related set of functions that perform useful work, often enabling one or more services. It may consist of software and/or hardware elements.
Capabilities	Platform, protocol, or configuration characteristics that a system supports.
Content	A digital work e.g. a ringing tone, a screen saver, etc.
Device	Device is a network entity that is capable of sending and/or receiving packets of information and has a unique device address. A device can act as either a client or a server within a given context or across multiple contexts. For example, a device can service a number of clients (as a server) while being a client to another server.
Push Access Protocol	A protocol used for conveying content that should be pushed to a client, and push related control information, between a Push Initiator and a Push Proxy/Gateway.
Push Content	Content, metadata and application level control information that has a shared interpretation by both Push Sender Agents and Push Receiver Agents.
Push Framework	Is the entire push system. The push framework encompasses the protocols, service interfaces, and software entities that provide the means to push data to user agents on a client
Push Initiator	The entity that originates Push Content and submits it to the push framework for delivery to a user agent on a client
Push OTA Protocol	A protocol used for conveying content between a Push Proxy/Gateway and a certain user agent on a client.
Push Proxy Gateway	A proxy gateway that provides push proxy services.
Push Receiver Agent	Push Receiver Agent is a logical entity that uses the SIP Push procedure to receive Push Content, and generate a response to the Push Sender Agent request.
Push Resource Identifier	The identifier of a resource, i.e. a component, function, enabler, or application that can send, receive, or process requests (see reference to Section 7.1)
Push Sender Agent	Push Sender Agent is a logical entity that creates a push request, and then uses the SIP Push procedure to send Push Content.
User Agent	Any software or device that acts on behalf of a user, interacting with other entities and processing resources.
User Plane	The User Plane includes the media (MSRP) and media control signalling between the Push Sender Agent and the Push Receiver Agent

3.3 Abbreviations

HTTP	Hypertext Transfer Protocol
IANA	Internet Assigned Numbers Authority
IP	Internet Protocol
MIME	Multipurpose Internet Mail Extensions
MMD	Multi-Media Domain

MMS	Multimedia Messaging Service
MSRP	The Message Session Relay Protocol
OMA	Open Mobile Alliance
OMNA	OMA Naming Authority
OTA	Over The Air
PAP	Push Access Protocol
RFC	Request For Comments
SDP	A Session Description Protocol
SIP	Session Initiation Protocol
TCP	Transmission Control Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WAP	Wireless Application Protocol
XML	Extensible Mark-up Language

4. Introduction

Push-based service enablers in OMA define the delivery of content to a mobile device utilizing push methods. The SIP Push architecture [PushSIPArch] defines the architectural context of such enablers in SIP-based environments. This specification defines the functions required of the Push Sender and Receiver Agents as defined in [PushSIPArch].
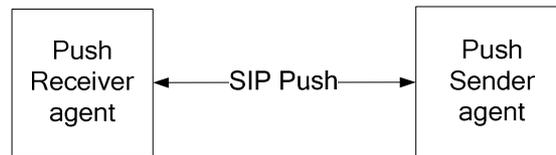
Figure 1: Push Sender and Receiver Agent

In general terms, the functionality defined in this specification will be referred to as "SIP Push". To implement this specification, it is necessary that the Push Sender and Receiver Agents interface with a SIP/IP Core network [PushSIPArch]. An example of SIP/IP Core network is the 3GPP IMS [3GPP TS 23.228] and 3GPP2 MMD [3GPP2 X.S0013-002-A] networks.

This specification will also make reference to specific SIP RFCs to further clarify the use of SIP for push service in the context of particular SIP/IP Core network definitions.

This specification encompasses the following:

1. SIP Push Protocol definition, including the push methods: Pager-Mode, and session mode (SIP INVITE and MSRP procedure).

2. Registration of Push Receiver Agents with Push Sender Agents.

3. Push Resource Identifier.

4. Authorization for push service.

5. Supporting multi-terminals belonging to the same user in SIP Push service.

4.1 End to End Push Service

Within the scope of this specification, the support for end-to-end push services focuses on the relationship between the Push Sender Agent and the Push Receiver Agent. While not limiting the potential types of end-to-end services that can leverage push, there are three types of push objectives that have been considered in the creation of this specification:

- Generic Push: Push is available towards user's device, asynchronously, whilst the user is registered with the SIP/IP Core network (e.g. used to broadcast content to a large community of users, and for generic "content-to-person" applications).

- "Trusted" Push: for pushing high value, trusted and certified information (e.g. device management/configurations, policies, based on terminal characteristics or user profile and related to application/service/user configuration).

- Selective Push: push can be used to provide content or application-related information in a personalized way, depending on user profile, user preferences or explicit interests, and device capabilities.

To provide options and methods best supporting those types of push, within the context of SIP, multiple mechanisms are defined for SIP Push. The SIP-based content delivery methods leveraged in this specification are based on Pager-Mode, and session-mode models.

Pager-Mode through SIP can be provided through the SIP MESSAGE method ([RFC3428]), which may be more appropriate in some contexts. The SIP MESSAGE method [RFC3428] is an extension to SIP that allows the transfer of messages to the

client. Since the SIP MESSAGE request is an extension to SIP, it inherits all the request routing and security features of that protocol. The SIP MESSAGE request carries the content or content reference (for content indirection) in the form of MIME body parts. Because SIP MESSAGE is a SIP signalling message, the ability to embed content is also limited by the maximum size of SIP signalling messages. The SIP MESSAGE requests do not themselves initiate a SIP dialog; under normal usage each SIP MESSAGE is a stands alone transaction, much like pager messages. Thus, each SIP MESSAGE request is independent and no session states are stored in the system. The Push Content is carried in a SIP message body.

Session-mode (also referred to as the SIP INVITE and MSRP procedure avoids the limitations on embedded content imposed by SIP MESSAGE. This limitation is especially acute if the Push Content is multimedia in nature. To allow arbitrarily large messages, the content is carried by MSRP [RFC4975]. A SIP session is established between the interested parties (Push Sender Agent and Push Receiver Agent) with MSRP as the media component. The SIP session can be used to transmit exactly one large message or a number of large messages either parallel or sequentially. The file selection mechanism allows for the Push Receiver Agent to understand ahead of delivery on what is going to be transferred as a number of additional attributes are supported [[RFC5547]].This gives the Push Receiver Agent a possibility to take decisions ahead of delivery and by that save network and terminal capacity.

Push Sender Agent is responsible for the delivery method to use based upon the requirements of the specific Push service being provided.

In multi-terminal scenario it is possible to deliver Push Content to all or subset of the terminals associated with the same user. When Push Content is delivered to a subset of the terminals associated with the same user GRUU [draft-ietf-sip-gruu] is used to identify each terminal. GRUU is obtained through registration procedure. If the SIP OPTIONS method is used for discovery of device capability, GRUU is required to keep track of the unique terminal.

4.2   Future OMA Push specification

This specification addresses SIP Push as a reference enabler, in which the requirements of specific end-to-end Push services are not addressed. Future work will address specific OMA enabler use of SIP Push in support of Push services. For example, a new version of the OMA Push enabler may be developed, to take advantage of SIP transport for legacy Push-based and Push-dependent services. In that case, the existing OMA Push architecture entities (Push Proxy Gateway and Push Client) will likely take on the roles of the SIP Push entities (Push Sender Agent and Push Receiver Agent, respectively). The current roles of the OMA Push entities will further be adapted to use of SIP Push as the Push transport protocol, e.g.

- For the Push Proxy Gateway, its primary role as an adapter between Push requests on the network side (via the Push Access Protocol (PAP)) and Push delivery to the Push Client will be extended to include SIP Push as an over-the-air protocol (in addition to the current Push Over-the-Air (Push-OTA) protocol). This may include updates to PAP to expose specific SIP Push related aspects, e.g. use of SIP URIs for addressing the target, or enhancements to Push Quality of Service options for selection of the SIP Push methods to be used.

- For the Push Client, its primary role as a Push message receiver and router inside terminals for Push applications will be extended to include SIP Push as an over-the-air protocol (in addition to Push-OTA). This may include the ability to register and subscribe to push services on behalf of the Push Application clients present in the terminal.

5. Capability Delivery and Resource Negotiation

This function provides the capability for a Push Receiver Agent to request the push resources capabilities from a Push Sender Agent and deliver its device profile to Push Sender Agents. An enabler using this capability will also be notified that a new Push Receiver Agent is available and requesting push resources.

5.1 Procedures at the Push Receiver Agent

Upon completion of SIP registration with the SIP/IP Core network or if changes to the supported push resources occur while registered the Push Receiver Agent:

1. SHALL generate a SIP OPTIONS request according to the rules and procedures of [RFC3261];

2. SHALL set the Request-URI of the SIP OPTIONS request to the URI of the Push Sender Agent;

3. MAY include in the P-Preferred-Identity header the URI of the Push Receiver Agent according to the rules and procedures of [RFC3325];

4. SHALL if the Push Receiver Agent obtained a public GRUU during registration include in the Contact header the GRUU value according to the rules and procedures of [draft–ietf-sip-gruu];

NOTE: Public GRUU is the recommended unless there are for enabler specific requirements to support a temporary GRUU.

5. SHALL include in the Contact header the Push Resource Identifier feature tag containing the values of all the push resources that are associated with the URI in the Contact header as defined in Section 7.1;

6. SHALL include an Accept-Contact header containing the Push Resource Identifier feature tag containing the values of all the requested push resources to be supported by the Push Sender Agent as defined in Section 7.1;

7. SHALL if the Push Receiver Agent obtained a public GRUU during registration according to the rules and procedures of [draft–ietf-sip-gruu] include a Supported header containing the option tag gruu;

8. SHALL include an Allow header containing all the SIP methods that the Push Receiver Agent supports;

9. SHOULD include a User-Agent header containing the model, vendor, and version of the Push Receiver Agent according to the rules and procedures of [RFC3261;

10. SHALL, if a User Agent Profile as defined by [OMA-UAProf] is supported, include a Content-Type header according to the rules and procedures of [RFC4483] containing:

    a. the MIME type message/external-body;

    b. "URL" in the ACCESS_TYPE parameter;

    c. the expiration parameter according to the rules and procedures of [RFC4483];

    d. the HTTP URL of the [OMA-UAProf] document in the URL parameter.

11. SHALL, if a User Agent Profile as defined by [OMA-UAProf] is supported, include a body according to the rules and procedures of [RFC4483] containing:

    a. application/rdf+xml in the Content-Type;

    b. "attachment" in the Content-Disposition, A Content-ID according to the rules and procedures of [RFC4483].

NOTE: If the User Agent Profile as defined by [OMA-UAProf] is not supported by the Push Receiver Agent, the Push Sender Agent can only obtain information about the Push Receiver Agent from the User-Agent header.

12. SHALL send the SIP OPTIONS request towards the SIP/IP Core network according to the rules and procedures f the SIP/IP Core network.

Upon receiving the SIP 200 "OK" response to the SIP OPTIONS request the Push Receiver Agent:

1. SHALL store the Push Capabilities of the Push Sender Agent that were included in the Accept-Contact header according to the rules and procedures of [RFC3841;

2. SHALL store the SIP methods supported by the Push Sender Agent that were included in the Allow header.

5.2   Procedures at the Push Sender Agent

Upon receiving a SIP OPTIONS request the Push Sender Agent:

1. SHALL verify that a P-Asserted-Identity header is present in the SIP OPTIONS request and that the URI in P-Asserted-Identity header is trusted. If the authorization check fails, the Push Sender Agent SHALL return a SIP 403 "Forbidden" response according to the rules and procedures of [RFC3261];

2. SHALL, if the SIP OPTIONS request contains a Content-Type header containing the MIME type message/external-body referring to a Content-Type Dev-Cap, and if the [OMA-UAProf] document is not already cached,retrieve the [OMA-UAProf] document of the Push Receiver Agent using the HTTP URL contained in the URL parameter of the Content-Type header;

3. SHALL if a GRUU was included in the Contact header in the SIP OPTIONS request, store the GRUU of the Push Receiver Agent and use that GRUU in the Request-URI of all SIP requests destined for that Push Receiver Agent;

4. SHALL generate a SIP 200 "OK" response according to the rules and procedures of [RFC3261];

5. SHALL include in the Contact header a Push Resource Identifier feature tag containing the supported push resource values, that were in the list of the push resource values in the Push Resource Identifier feature tag in the Contact header of the SIP OPTIONS request from the Push Receiver Agent, as defined in Section 7.1;

6. SHOULD include an Accept header containing the MIME types supported by the Push Sender Agent including, message/external-body and application/rdf+xml, and any other MIME types supported by the Push Sender Agent as part of the service it provides, according to the rules and procedures of [RFC3261];

7. SHALL include an Allow header containing all the SIP methods that the Push Sender Agent supports; and;

8. SHALL send the SIP 200 "OK" response towards the SIP/IP Core network according to the rules and procedures of the SIP/IP Core network.

6. Registration

6.1   Procedures at Push Receiver Agent

When the Push Receiver Agent register, re-register and de-register to the SIP/IP Core network, it SHALL do so according to rules and procedures of [RFC3261] with the clarifications in the following subsection.

When the Push Receiver Agent registers, re-register, and de-register, the Push Receiver Agent:

1. SHALL generate a SIP REGISTER request according to the rules and procedures of [RFC3261];

2. SHALL include in the Contact header the Push Resource Identifier feature tag +g.oma.pusheventapp, containing the value of all supported push resources (see Section 7.1);

3. SHALL include a require header with the option tag "pref" according to rules and procedures of [RFC3840];

NOTE: The Require header ensures that, per [RFC3480], the caller preferences indicated in the Contact header are stored by the SIP/IP Core network and made available to application servers, e.g. the Push Sender Agent.

4. SHALL request a GRUU value during the registration process, if supporting the Capability Delivery and Resource Negotiation in a multi terminal environment;

5. MAY request a GRUU value during the registration process if the Capability Delivery and Resource Negotiation is not supported in a multi terminal environment;

6. SHALL include in the Contact header the +sip.instance parameter according to rules and procedures of [draft-ietf-sip-gruu];

NOTE: Upon successful registration, the SIP/IP Core network returns the GRUU values (temporary and public GRUU values). These GRUU values can then be used by the Push Receiver Agent in non-REGISTER requests.

7. SHALL include a Supported header with the value "gruu";

8. SHALL send the SIP REGISTER request to the SIP/IP Core network according to the rules and procedures of the SIP/IP Core network;

9. On receiving a SIP 200 "OK" response to the SIP REGISTER request where the Contact header field contains a "pub-gruu" and "temp-gruu" value, the Push Receiver Agent SHALL retrieve and store the "pub-gruu" and "temp-gruu" from the Contact headers to be usage according to rules and procedures of [draft-ietf-sip-gruu].

When the Push Receiver Agent needs to de-register one more Push Resources but remain SIP/IP Core network registered, the Push Receiver Agent:

1. SHALL generate a SIP REGISTER request according to the rules and procedures of [RFC3261];

2. SHALL include in the Contact header the Push Resources Identifier feature tag +g.oma.pusheventapp, containing the value of all supported push resources that need to remain registered;

NOTE: If no push resources need to remain registered the Push Resources Identifier feature tag +g.oma.pusheventapp is not included in the Contact header of the SIP REGISTER request.

3. SHALL include a require header with the option tag "pref" according to rules and procedures of [RFC3840];

4. SHALL send the SIP REGISTER request to the SIP/IP Core network according to the rules and procedures of the SIP/IP Core network.

When the Push Receiver Agent needs to de-register from SIP/IP Core network, the Push Receiver Agent:

1. SHALL send a SIP REGISTER request according to the rules and procedures of [RFC3261];

2. SHALL include an Expire header with the value set to 0;

3. SHALL send the SIP REGISTER request to the SIP/IP Core network according to the rules and procedures of the SIP/IP Core network.

7. Resources and Application Addressing

7.1 Push Resource Identifier

A Push Resource Identifier SHALL identifies the resources (e.g. enabler-specific application) of implementing enablers. The resources MAY be targeted using SIP Push as a content or notification of transport mechanism.

Push Resource Identifiers are represented as a media feature tag, as specified in Section 7.2.

7.1.1 Feature Tag Format

The syntax of the Push Resource Identifier feature tag is name=value.

The Push Resource Identifier SHALL be encoded into a feature tag where:

- The name of the media feature tag [RFC3840] is g.oma.pusheventapp, an object identifier in the global tree assigned by IANA according to [RFC2506]. When the feature tag is included in a header, it is prefixed with "+".

- The value of the media feature tag is the "event-app-id" presented as a quoted-string (which may be a comma-separated list) that defines the resource(s).

  The syntax of the quoted-string is "event-app-id*("," event-app-id)", where "event-app-id" = 1*(%x21 / %x23-2B / %x2D-7E).

For example, in the context of a Push service based upon the OMA Push enabler (assuming it has been enhanced as a SIP Push implementing enabler, referred to here as OMA Push 2.x), the Push Resource Identifier could be used to dispatch Push events to applications that use the Push enabler, as supported by the OMA Push architecture. If there is an MMS application that leverages the OMA Push 2.x enablers for MMS notifications, the "event-app-id" can be the full URN or only the Namespace Specific String of the MMS Push Application ID defined by [OMNA], i.e. x-wap-application:mms.ua or only mms.ua. The feature tag in this case would be g.oma.pusheventapp ="x-wap-application:mms.ua" or g.oma.pusheventapp =" mms.ua".

7.2 Push Resource Identifier Usage

7.2.1 REGISTER

The Push Resource Identifier of each supported push application MAY be included as a feature tag in the Contact header of the REGISTER message.

7.2.2 SIP MESSAGE

In delivering push messages through the SIP MESSAGE method the Push Sender Agent MAY add the Push Resource Identifiers as a feature tag in the Accept-Contact header according to rules and procedures in Section 8.1.

When receiving a SIP MESSAGE request, the Push Receiver Agent SHALL uses the Push Resource Identifier to identify the targeted push resource.

An example usage (sample flow) of the SIP MESSAGE method with feature tag is shown in Appendix B.2.

7.2.3 SIP INVITE

The Push Sender Agent MAY add the Push Resource Identifiers as a feature tag in the Accept-Contact header of the SIP INVITE message as specified in the Section 8.2.

8. Push Operations

This section describes three delivery mechanisms for SIP Push. It is the choice of the individual Push Sender Agent to implement and select the appropriate Push method, e.g. based upon the service requirements. Such service-specific method selection criteria are considered outside the scope of SIP Push.

8.1   SIP MESSAGE Method (Pager- Mode)

8.1.1   Procedures at the Push Receiver Agent

Upon receiving an incoming SIP MESSAGE request according to [RFC3428] the Push Receiver Agent:

1. SHALL check if a Push Resource Identifier is present in the Accept-Contact header as defined in Section 7.1. If a Push Resource Identifier feature tag is not present or the value is not recognised, the Push Receiver Agent SHALL return a SIP 403 "Forbidden" response;

2. SHALL verify that a P-Asserted-Identity header is present and that the URI in P-Asserted-Identity header is trusted. If the authorization check fails, the Push Receiver Agent SHALL return a SIP 403 "Forbidden" response;

3. SHALL generate a SIP 200 "OK" according to [RFC3428];

4. SHALL retrieve the Push Content at the indicated location, if the content is indirectly referenced in the SIP MESSAGE request as specified in [RFC4483];

5. SHALL provide the Push Content to the targeted push application.

8.1.2   Procedures at the Push Sender Agent

When generating a SIP MESSAGE request according to [RFC3428] the Push Sender Agent:

1. SHALL include in the Accept-Contact header the Push Resource Identifier feature tag name and the value as defined in Section 7.1;

2. SHALL set the Request-URI of the SIP MESSAGE request to either:

    a. the public user identity of the intended recipient if the request is to be sent to all Push Receiver Agents of the intended recipient, or

    b. the public or temporary GRUU of the intended recipient, according to the rules and procedures of [draft-ietf-sip-gruu] if the request is to be sent to a specific Push Receiver Agent of the intended recipient;

3. SHALL use content indirection as specified in [RFC4483] if the entire SIP MESSAGE request exceeds 1300 bytes as defined in [RFC3428] or if the implementing enabler requires use of content interaction;

NOTE: The process for arranging the availability and creating the content reference to the content is an application or implementing-enabler specific issue. For example:

    1. The application may deliver Push content to the Push Sender Agent, which caches the content and uses the cached location as the indirect reference.

    2. The application may request that the Push Sender Agent issue a Push content availability notification, but retain the content at the application server, and thus give the Push Sender Agent a URI reference to the content, which the Push Sender Agent forwards to the Push Receiver Agent in the SIP MESSAGE.

A Push Sender Agent which implements a Push service based upon SIP Push may offer either or both of the indirection approaches (1 or 2) above to applications, and may offer the application the option of choosing the method. An application server which directly implements the Push Sender Agent functionality may use either of the above approaches (1 or 2) to reference the content.

4. MAY include in the P-Asserted-Identity header the URI of the Push Sender Agent according to rules and procedures of [draft-ietf-sipping-asserted-id];

5. SHALL include the Push Content in the body of the SIP MESSAGE request;

6. SHALL send the SIP MESSAGE request towards the SIP/IP Core network according to the procedures of the SIP/IP Core network.

NOTE: May need to validate compatibility of the Push Receiver Agent with the content per the requirements of the implementing enabler.

8.2 SIP INVITE and MSRP Procedure (Session-Mode)

When a Push Sender Agent wishes to send a large content without using content indirection, the Push Sender Agent initiates a MSRP session as described in Section 8.2.2.

8.2.1 Procedures at the Push Receiver Agent

8.2.1.1 Push Receiver Agent Invited to a MSRP Session

When the Push Receiver Agent receives a SIP INVITE request to set up a MSRP session, the Push Receiver Agent:

1. SHALL check if the accept-type attribute of the SDP media ("m=") line in the SIP INVITE request are supported by the Push Receiver Agent and if not, reject the request with a SIP 488 "Not Acceptable Here" response. The response MAY include an SDP body which contains the list of the accept-type attributes supported by the Push Receiver Agent according to rules and procedures of [RFC3261] and [RFC4975]. Otherwise, continue with the rest of the steps;

2. SHALL check if a Push Resource Identifier is present in the Accept-Contact header as defined in Section 7.1. If a Push Resource Identifier feature tag is not present or the value is not recognised, the Push Receiver Agent SHALL return a SIP 403 "Forbidden" response;

3. SHALL verify that a P-Asserted-Identity header is present and the URI in P-Asserted-Identity header is trusted. If the authorization check fails, the Push Receiver Agent SHALL return a SIP 403 "Forbidden" response;

4. SHALL accept from the Push Sender Agent's list of files, the files that a Push Receiver Agent is willing to receive when the "file-selector" attribute as specified in [RFC5547] is included in the SDP contained in the SIP INVITE request;

When generating a SIP 200 "OK" response to the SIP INVITE request, the Push Receiver Agent:

1. SHALL generate a SIP 200 "OK" response according to the rules and procedures of [RFC3261];

2. SHALL include an SDP answer according to rules and procedures of [RFC3264], [RFC4566], and [RFC4975]. The Push Receiver Agent:

    a. SHALL set the SDP directional media attribute to a=recvonly;

    b. MAY indicate the maximum size message they wish to receive using the max-size attribute ("a=") line according to rules and procedures of [RFC4975];

    c. SHALL include the "a=setup" attribute with the value "active" according to rules and procedures of [RFC4145].

3. SHALL send the SIP 200 "OK" response towards the Push Sender Agent according to rules and procedures of the SIP/IP Core network;

4. SHALL prepare to receive MSRP SEND request as described in [RFC4975].

8.2.1.2 MSRP session release

Upon reception of a SIP BYE request, the Push Receiver Agent:

1. SHALL generate a SIP 200 "OK" response according to rules and procedures of [RFC3261];

2. SHALL send the SIP 200 "OK" response according to rules and procedures of SIP/IP Core network;

3. SHALL release User Plane resources associated with the MSRP session.

8.2.1.3 MSRP Relays

Push Receiver Agents MAY support the use of intermediaries for MSRP sessions, as described in [RFC4976].

8.2.2 Procedures at the Push Sender Agent

8.2.2.1 Establishment of a SIP session for MSRP

When the establishment of an MSRP session is needed, the Push Sender Agent:

1. SHALL generate an initial SIP INVITE request as according to rules and procedures of [RFC3261];

2. SHALL include in the Accept-Contact header the Push Resource Identifier feature tag name and the value as defined in Section 7.1;

3. SHALL set the Request-URI of the SIP MESSAGE request to either

    a. the public user identity of the intended recipient if the request is to be sent to all Push Receiver Agents of the intended recipient, or

    b. the public or temporary GRUU of the intended recipient, according to the rules and procedures of [draft-ietf-sip-gruu] if the request is to be sent to a specific Push Receiver Agent of the intended recipient;

4. MAY include in the P-Asserted-Identity header the URI of the Push Sender Agent according to rules and procedures of [RFC3325];

5. SHALL include an SDP offer according to rules and procedures of [RFC3264], [RFC4566] and [RFC4975]. The Push Sender Agent:

    a. SHALL set the SDP directional media attribute to a=sendonly;

    b. MAY add the media attribute to the SDP settings as specified in [[RFC5547]];

    c. SHALL add multiple media ("m=") lines if more than one file is to be transferred as described in [RFC4566];

    d. SHOULD include the "a=setup" attribute with the value "passive" according to [RFC4145].

6. SHALL send the SIP INVITE request towards the Push Receiver Agent according to rules and procedures of the SIP/IP Core network.

NOTE: May need to validate compatibility of the Push Receiver Agent with the content per the requirements of the implementing enabler.

On receiving a SIP 200 "OK" response to the SIP INVITE request the Push Sender Agent:

1. SHALL interact with the User Plane as specified in 8.2.3.

8.2.2.2 SIP session cancellation

When the Push Sender Agent wants to cancel the SIP session initiation, and the Push Sender Agent has not yet received a final SIP response for the SIP INVITE request, Push Sender Agent SHALL send a SIP CANCEL request according to rules and procedures of [RFC3261].

8.2.2.3 SIP session release

When the last MSRP session is released the Push Sender Agent:

1. SHALL generate a SIP BYE request according to the rules and procedures of [RFC3261];

2. SHALL send the SIP BYE request according to rules and procedures of SIP/IP Core network.

8.2.2.4 MSRP Relays

Push Sender Agents MAY support the use of intermediaries for MSRP sessions, as described in [RFC4976].

8.2.3 User Plane

8.2.3.1 General

Media parameters and media formats for the MSRP Session are to be negotiated in SDP with an offer and answer as specified in [RFC3264] and [RFC3261].

8.2.3.2 MSRP session

The Push Receiver Agent and Push Sender Agent SHALL follow the rules and procedures of [RFC4975] and [RFC5547], with the following exception:

1. The Push Receiver Agent SHALL establish the TCP connection used for the MSRP towards the IP address included in the SDP connection ("c=") line and the TCP port included in the SDP media ("m=") line of the MSRP SDP description as specified in the [RFC4566] in the SDP body of the SIP INVITE request that the Push Sender Agent sent.

NOTE: In this case the Push Sender Agent does not use the MSRP path attribute to establish the TCP connection.

NOTE: MSRP REPORT gives the ability for delivery confirmation.

9. Security

SIP Push Receiver Agents and SIP Push Sender Agents SHALL rely on security features and mechanisms provided by the underlying SIP/IP Core network, to e.g. secure the service environment and authenticate users.

For Push Receiver Agents implemented on trusted or untrusted network, authentication and secure communication channels SHOULD use intra-network security procedures provided by the SIP/IP Core network.

For applicability to specific SIP/IP Core network environments please refer to Appendix E. The following sub-sections provide detailed security requirements for the SIP Push enabler.

9.1 Trust Model

The SIP Push enabler SHOULD rely upon the trust model for SIP signalling provided by the SIP/IP Core network.

9.2 SIP Signalling Security

9.2.1 Source Origin Authentication

Push Receiver Agents and Push Sender Agents SHALL use the security mechanisms provided by the SIP/IP Core network to ensure message source authentication. See Appendix E as examples of SIP/IP Core network implementations.

When a SIP Push Sender Agent is within a trusted network, the Push Sender Agent MAY assert the authenticity of the URI of the Push Sender Agent by including the P-Asserted-Identity header when the Push Sender Agent is acting as the originating user agent as specified in [RFC3325].

9.3 User Plane security

SIP Push implementing enablers SHALL ensure that user plane security is addressed through the options available for the user plane transport protocols, e.g. MSRP or HTTP (in the case of content redirection).

9.4 Minimizing Congestion in SIP/IP Core

A misbehaving PSA may flood the SIP/IP Core network signalling network with push requests. It is assumed that the SIP/IP Core network implementation follows the SIP [RFC3261] guidelines and may apply additional measures for congestion control.

A Push Sender Agent SHOULD minimizes ineffective Push attempts, e.g. avoid sending push requests when it knows that the Push Receiver Agent is unavailable.

Appendix A.   Change History   (Informative)

A.1   Approved Version History

Reference	Date	Description
n/a	n/a	No prior version –or- No previous version within OMA

A.2   Draft/Candidate Version 1.0 History

Document Identifier	Date	Sections	Description
Draft Versions OMA-SIP-Push-V0_1	21 Jul 2005	-	From input paper OMA-PUSH-2005-0032; OMA-PUSH-2005-0030; and emails
OMA-SIP-Push-V0_2	16 Sep 2005	6.2 and 6.3	OMA-PUSH-2005-0043R02-PushUsingSIP-SenderReceiver-Description
OMA-SIP-Push-V0_3	19 Oct 2005	All	Incorporated OMA-PUSH-2005-0037R02 OMA-PUSH-2005-0039R04 OMA-PUSH-2005-0046R01 OMA-PUSH-2005-0047 OMA-PUSH-2005-0049 OMA-PUSH-2005-0051 OMA-PUSH-2005-0053 OMA-PUSH-2005-0054
OMA-SIP-Push-V0_4	17 Apr 2006	All	Editorial clean up, and generate a clean version of the document.
OMA-SIP-Push-V0_4	17 May 2006	All	OMA-PUSH-2006-0004R02 OMA-PUSH-2006-0010R02 OMA-PUSH-2006-0011R02 OMA-PUSH-2006-0012R02 OMA-PUSH-2006-0013R02 OMA-PUSH-2006-0016R01
OMA-SIP-Push-V0_6	23 Jun 2006		OMA-PUSH-2006-0022R01 OMA-PUSH-2005-0055R04
OMA-SIP-Push-V0_7	12 Apr 2007		Based on OMA-CD-2006-074-Minutes_21Aug2006_Push_Beijing_Meeting OMA-CD-2006-0014R03 Based on OMA-CD-2006-0101-MINUTES_21Sept2006Push_CC OMA-CD-2006-0039R01 OMA-CD-2006-0064R01 OMA-CD-2007-0085-INP_SIP_Push_IC0034_Comment_Tracking - implemented: A001, A002, A003, A007, A008, A012, A013
OMA-SIP-Push-V0_8	6 Jun 2007		Editorial: Move section 6.1 and 6.2 to section 6. Global technical change to reflect +g.oma.icsi.push';+ g.oma.iari.push.XXX based on contribution -0125R04 OMA-CD-2007-0125R04-CR_SIP_push_service_and_application_addressing
OMA-SIP-Push-V0_9	4 Jul 2007		OMA-CD-2007-0124R04-CR_Introducing_INVITE_MSRP_as_push_method OMA-CD-PUSH-2007-0012-CR_SIP_Push_ClientCapabilitiesWithPresence
OMA-SIP-Push-V0_10	10 Sep 2007		OMA-CD-PUSH-2007-0008R02-CR_SIP_Push_Client_Capabilities OMA-CD-PUSH-2007-0032R02-CR_Update_of_SUBSCRIBE_NOTIFY_Method OMA-CD-PUSH-2007-0024R04-CR_Definition_of_additional_profile_to_the_sipping_config_framework_package
OMA-TS-SIP_Push-V1_0	28 Sep 2007	All	Version fixed from V0_10 to V1_0 Editorial fixes: - Cover page with correct file name and versioning - History moved to App A and fixed as per template - 2007 template and styles - fixed cross-references

Document Identifier	Date	Sections	Description
	14 Nov 2007	All	OMA-CD-PUSH-2007-0047-CR_AppID_inconsistency
			OMA-CD-PUSH-2007-0061R01-INP_MSRP_Relay
			OMA-CD-PUSH-2007-0063R01-INP_SIP_PUSH_Content_Indirection
			OMA-CD-PUSH-2007-0024R08-CR_Definition_of_additional_profile_to_the_sipping_config_framework_package
			OMA-CD-PUSH-2007-0058R01-INP_AI2007_0001
			OMA-CD-PUSH-2007-0064-CR_Device_reference_removal
			OMA-CD-PUSH-2007-0054R03-CR_An_update_to_support_for_CSI_and_ARI_in_SIP_Push
			OMA-CD-PUSH-2007-0068-CR_OMA_TS_SIP_Push_V1_0_20070928_D_left_to_change
			OMA-CD-PUSH-2007-0065R02-CR_SCR_in_SIP_Push
	05 Dec 2007	All	OMA-CD-PUSH-2007-0085R01-INP_OMA_TS_SIP_Push_V1_0_20071204_D
			OMA-CD-PUSH-2007-0077-CR_SIP_Push_TS_Edits.doc (implemented Change #1 and #3 only).
	08 Jan 2008	All	OMA-CD-PUSH-2007-0060R01-INP_SIP_PUSH_TS_edits
			OMA-CD-PUSH-2007-0088R01-CR_Section_9_Rewording
			OMA-CD-PUSH-2007-0067R02-CR_ICSI_IARI_usage_example
			OMA-CD-PUSH-2007-0082R01-CR_New_figure_for_ICSI_and_IARI_usage
	07 Feb 2008		OMA-CD-PUSH-2007-0091R01-CR_Security_chapter_10
			OMA-CD-PUSH-2008-0001R02-CR_Example_Consistency
			OMA-CD-PUSH-2008-0006-CR_Resolution_AI1012_1
			OMA-CD-PUSH-2008-0005R01-CR_Includes_changes_agreed_to_by_the_group_based_on_the_comments_from_CableLabs.
			OMA-CD-PUSH-2008-0010R02-CR_Update_of_the_REFER
	15 Feb 2008		OMA-CD-PUSH-2008-0008R02-CR_CR_SIPPUSH_NewAppendix
			OMA-CD-PUSH-2007-0079-R5-CR_Multi_terminals_situation_in_SIP_Push_latest
	03 Mar2008		Same as 15 February, 2008 (clean version)
	14 Mar 2008		Editorial fixes:
			2008 template and styles
			History Box fixed
	11 Apr 2008	All	Document updated as per CONRR comments:
			B007: fix occurances of [OMNA]
			B071: fix occurrences of NOTE:
			B086: fix double commas, semi-columns, double periods.
	09 June 2008		OMA-CD-PUSH-2008-0033R04-CR_Config_Framework_Usage
			OMA-CD-PUSH-2008-0063-CR_TS_update_removing_Config_framework
	17 June 2008		OMA-CD-PUSH-2008-0047R01-CR_Push_Resource_Identifier_Update
			OMA-CD-PUSH-2008-0061R01-CR_Agreed_Syntax_in_Chapter_7_Message_method
			OMA-CD-PUSH-2008-0062-CR_Agreed_Syntax_in_Chapter_7_INVITE_method.
			OMA-CD-PUSH-2008-0035-CR_Feature_Tag_Registration
			OMA-CD-PUSH-2008-0036R01-CR_HTTP_usage
	24 June 2008		Global remove and replace "Application Resource" with "Push Resource"
			Correct all of the references in the SCR table.
	03 July 2008		OMA-CD-PUSH-2008-0058R02-CR_NAT_Issues
			OMA-CD-PUSH-2008-0068R01-CR_SIP_488_response.
			OMA-CD-PUSH-2008-0070R01-CR_Remove_Session_Identity
			OMA-CD-PUSH-2008-0072-CR_Changes_agreed_to_TS_for_CONRR_B238
			OMA-CD-PUSH-2008-0074R01-CR_section_10_in_TS
			OMA-CD-PUSH-2008-0080-CR_TS_Section_1_CommentB005.
			OMA-CD-PUSH-2008-0085R01-CR_CONRR_B113_Content_Indirection
			OMA-CD-PUSH-2008-0086R01-CR_CONRR_B239_Push_Service

Document Identifier	Date	Sections	Description
	13 Aug 2008		OMA-CD-PUSH-2008-0084-CR_TS_Section9.1_CommentB236.
			OMA-CD-PUSH-2008-0089-CR_CONRR_B242_IARI_and_PRI
			OMA-CD-PUSH-2008-0081R02-CR_TS_Section6_CommentB230.
			OMA-CD-PUSH-2008-0091R05-CR_Option_Usage
			OMA-CD-PUSH-2008-0098R01-CR_Updates_to_GRUU_
			OMA-CD-PUSH-2008-0105-CR_Purpose_of_GRUU.
			OMA-CD-PUSH-2008-0095R03-CR_CONRR_Appendix_B_Flows_Part_2
			OMA-CD-PUSH-2008-0096R02-CR_CONRR_Appendix_B_Flows_Part_3
	15 Aug. 2008		CONRR comment #B044, B327
	29 Aug 2008		OMA-CD-PUSH-2008-0117R01-CR_Comments_to_CR_0098R01
			OMA-CD-PUSH-2008-0123R01-CR_TS_section_9
			OMA-CD-PUSH-2008-0124R02-CR_Static_Conformance_Requirements_update
			OMA-CD-PUSH-2008-0131R01-CR_Status_of_PRA.
			OMA-CD-PUSH-2008-0125R01-CR_DoS_Attacks
			OMA-CD-PUSH-2008-0128R03-CR_TS_clarification_appB
			OMA-CD-PUSH-2008-0130R02-CR_example_flow_using_SIP_OPTIONS
			OMA-CD-PUSH-2008-0092R04-CR_CONRR_Appendix_B_Flows_Part_1
			OMA-CD-PUSH-2008-0116R01-CR_Clarification_TS_section_4.1
			OMA-CD-PUSH-2008-0119-CR_CR_CONRR_B_CR088_5
Candidate Version: OMA-TS-SIP_Push-V1_0	02 Dec 2008	All	Editorial fixes: cross-references added
			Status changed to Candidate by TP:
			OMA-TP-2008-0439-INP_SIP_Push_V1_0_ERP_for_Candidate_Approval
Draft Versions OMA-TS-SIP-Push-V1_0	10 Mar 2010	All	OMA-CD-PUSH-2009-0079-CR_SIP_Push_TS_Fix_ICSI
			OMA-CD-PUSH-2009-0100-CR_Cleanup_to_SIP_PUSH_TS
			Updated 2010 template
Candidate Version: OMA-TS-SIP_Push-V1_0	01 Apr 2010	All	Status changed to Candidate by TP:
			OMA-TP-2010-0171-INP_SIP_PUSH_V1_0_ERP_for_Notification

Appendix B.   Sample flows   (Informative)

Note: The following flows and message examples are intended to be illustrative only. In particular:

- They do not show all parameters, or are necessarily complete/correct in every parameter.

- They are not meant to reflect any specific deployment scenario, e.g. re the "Via:" header, SIP may be deployed over UDP or TCP, and in the wireless case will typically be deployed over UDP.

- They are informative, and are not meant to represent normative requirements, either explicitly or implicitly. The normative requirements if any are addressed by the normative sections of this specification, or the referenced specifications that address the specific flow/message aspect.

- The examples include an IMS Communication Service Identifier but not an IMS Application Reference Identifier (IARI) since in these examples it is assumed that the target application is the default application for the communication service, thus the IARI does not need to be included.

Note: Message aspects of particular importance to SIP Push are shown in bold.

B.1   Registration

B.1.1   Third Party Registration in IMS Networks

This flow shows the initial registration or re-registration of a Push Receiver Agent with a SIP/IP Core network, and 3rd-party registration with the Push Sender Agent, as supported in IMS.
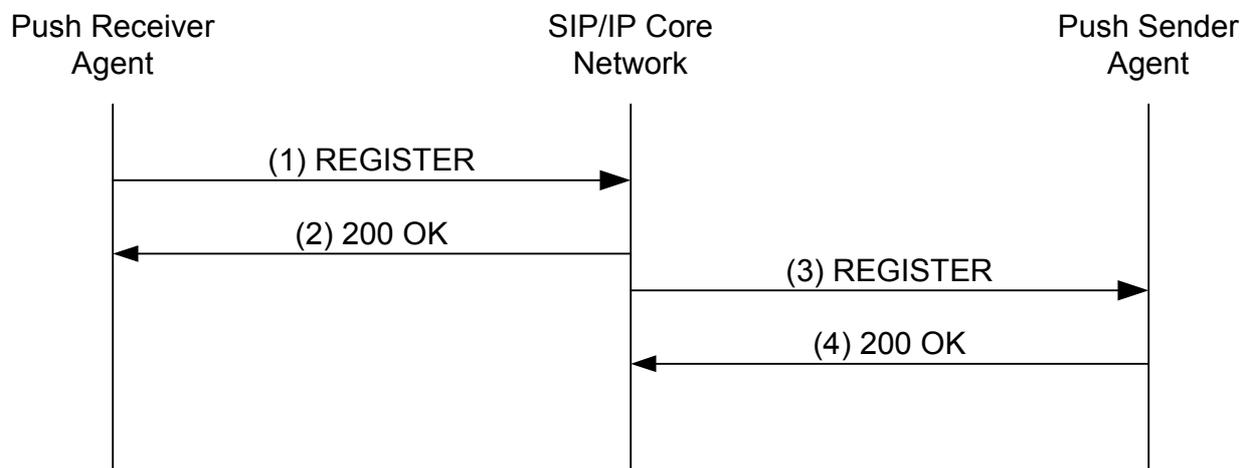
Figure 2: Third Party Registration in IMS Networks

1.   Upon connecting to the SIP/IP Core Network, the Push Receiver Agent will perform the registration procedure as specified in 3GPP [3GPP TS 23.228] and 3GPP2 [3GPP2 X.S0013-002-A], respectively. For de-registration, the same procedures will apply accordingly.

REGISTER from the Push Receiver Agent

```
REGISTER sip:registrar.domain.com SIP/2.0
Via: SIP/2.0/UDP userpc.domain.com:5060;branch=z9hG4bKnashds7
Max-Forwards: 70
```

```
To: Bob <sip:user@domain.com>
From: Bob <sip:user@domain.com>;tag=456248
Call-ID: 843817637684230@998sdasdh09
CSeq: 1826 REGISTER
Require: pref
Supported: gruu
Contact: <sip:user@192.0.2.4>;+sip.instance="<urn:uuid:f81d4fae-7dec-11d0-a765-
00a0c91e6bf6>" +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.omapush",
+g.oma.pusheventapp="wml.ua,mms.ua,wv.ua,loc.ua,syncml.dm"
Expires: 7200
Content-Length: 0
```

2. Upon successful registration procedure (assuming the Push Receiver Agent is authorized to use the service), the SIP/IP Core acknowledges the user registration.

    200 OK as sent in response to the Push Receiver Agent

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP userpc.domain.com:5060;branch=z9hG4bKnashds7;received=1.2.3.4
To: Bob <sip:user@domain.com>;tag=2493k59kd
From: Bob <sip:user@domain.com>;tag=456248
Call-ID: 843817637684230@998sdasdh09
CSeq: 1826 REGISTER
Contact: <sip:user@192.0.2.4> ;pub-gruu="sip:user@domain.com;gr=urn:uuid:f81d4fae-7dec-
11d0-a765-00a0c91e6bf6";temp-gruu="sip:tgruu.7hs==jd7vnzga5w7fajsc7-
ajd6fabz0f8g5@domain.com;gr";+sip.instance="<urn:uuid:f81d4fae-7dec-11d0-a765-
00a0c91e6bf6>",+g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-
service.ims.icsi.omapush",+g.oma.pusheventapp="wml.ua,mms.ua,wv.ua,loc.ua,syncml.dm";expir
es=3600
Expires: 7200
Content-Length: 0
```

3. After processing of initial filter criteria information, the SIP/IP Core Network generates a third-party REGISTER request and sends it to the Push Sender Agent, as an interested application server identified via the initial filter criteria configuration. Key parameters of the third party REGISTER request include the public user identity in the To header, the address of the SIP registrar (e.g. S-CSCF) in the From header, and the expiration time. The same procedures apply for de-registration. Note that the 3rd-party REGISTER does not include the contact details of the Push Receiver Agent, thus cannot be used by itself to discover those details. Other solutions may be used to resolve this problem: (a) use of the registration event notification (see the next flow); (b) receiving this information in SIP OPTIONS (Appendix B.2).

    Third Party REGISTER to the Push Sender Agent as application server

```
REGISTER sip:psa.domain.com SIP/2.0
Via: SIP/2.0/TCP registrar.domain.com;branch=z9hG4bK123dsghds
Max-Forwards: 70
To: Bob <sip:user@domain.com>;tag=456248
From: <sip:registrar.domain.com>
Call-ID: 843817637684230@998sdasdh09
CSeq: 1826 REGISTER
Contact: <sip:registrar@domain.com>
Expires: 7200
Content-Length: 0
```

4. The Push Sender Agent acknowledges the REGISTER with SIP 200 OK.

   200 OK as sent in response to the SIP/IP Core

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP psa.domain.com:5060;branch=z9hG4bKnashds7;received=1.2.3.4
To: Bob <sip:user@domain.com>;tag=456248
From: <sip:registrar.domain.com>
Call-ID: 843817637684230@998sdasdh09
CSeq: 1826 REGISTER
Contact: <sip:psa.domain.com>
Expires: 7200
Content-Length: 0
```

B.1.2   Registration Event Notification in IMS Networks

This flow shows the initial registration of a Push Receiver Agent with a SIP/IP Core network, Third Party Registration and delivery of the registration event to the Push Sender Agent.
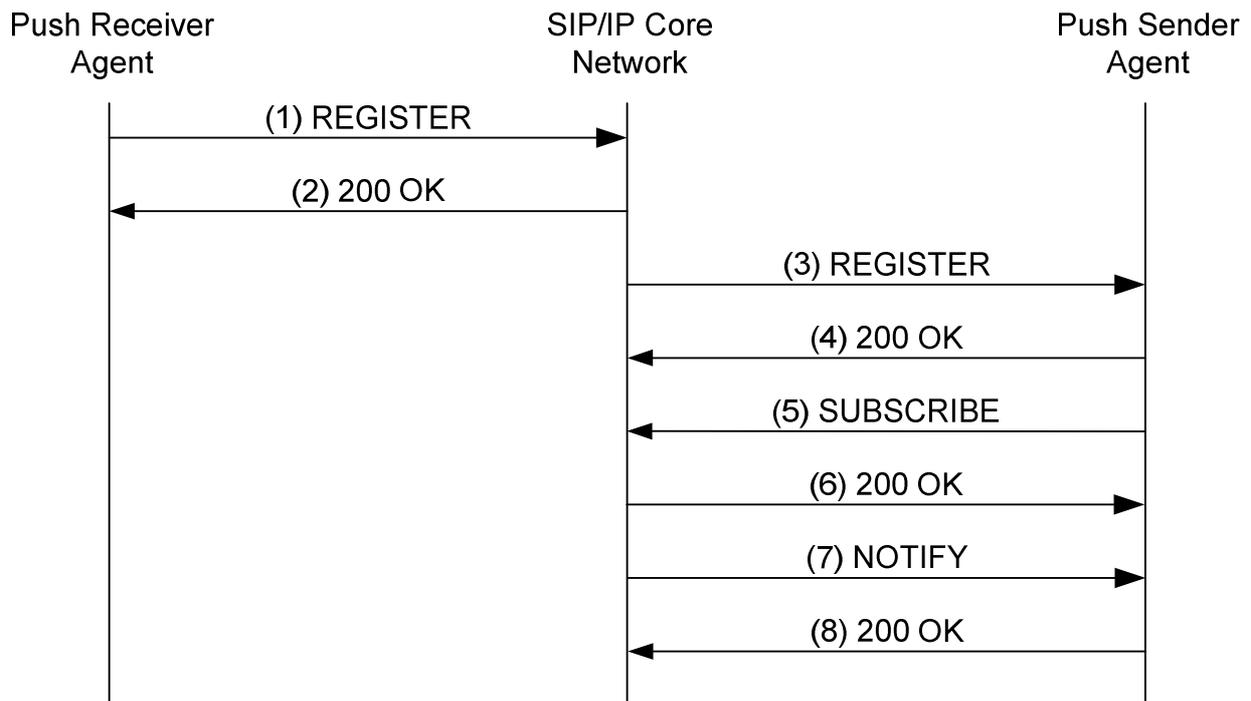
Figure 3: Registration Event Notification in IMS Networks

1. Same as "Third Party Registration in IMS Networks" step 1.

2. Same as "Third Party Registration in IMS Networks" step 2.

3. Same as "Third Party Registration in IMS Networks" step 3.

4. Same as "Third Party Registration in IMS Networks" step 4.

5. The Push Sender Agent subscribes to the reg-event package for user@domain.

SUBSCRIBE as sent to the SIP/IP Core

```
SUBSCRIBE sip:user@domain.com SIP/2.0
Via: SIP/2.0/UDP psa.domain.com;branch=z9hG4bKnashds7
P-Asserted-Identity: sip:psa.domain.com
To: <sip:registrar.domain.com>
From: <sip:psa.domain.com;tag=27182>
Call-ID: gbjg0b@psa.domain.com
CSeq: 9887 SUBSCRIBE
Contact: <sip:psa.domain.com>
Event: reg
Max-Forwards: 70
Accept: application/reginfo+xml
```

6. The SIP registrar establishes a subscription to the reg-event package for user@domain.com and sends a 200 OK response to the SUBSCRIBE.

200 OK as sent to the Push Sender Agent

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP psa.domain.com;branch=z9hG4bKnashds7
To: <sip:registrar.domain.com;tag=262281>
From: <sip:psa.domain.com;tag=27182>
CSeq: 9887 SUBSCRIBE
Contact: <sip:registrar.domain.com>
Content-Length: 0
Expires: 3600
```

7. The SIP registrar finds an active subscription to the reg-event package for the Push Sender Agent, and issues a NOTIFY with the registration information.

NOTIFY to the Push Sender Agent as application server

```
NOTIFY sip:psa.domain.com SIP/2.0
Via: SIP/2.0/UDP registrar.domain.com;branch=z9hG4bKnashds7
To: <sip:registrar.domain.com;tag=262281>
From: <sip:psa.domain.com;tag=27182>
Call-Id: gbjg0b@psa.domain.com
CSeq: 633 NOTIFY
Subscription-State: active;expires=3600
Event: reg
Max-Forwards: 70
Content-Type: application/reginfo+xml
Contact: <sip:registrar.domain.com>
Content-Length: ...

<?xml version="1.0"?>
  <reginfo xmlns="urn:ietf:params:xml:ns:reginfo"
           xmlns:gr="urn:ietf:params:xml:ns:gruuinfo"
           version="1" state="full">
    <registration aor="sip:user@domain.com" id="a7" state="active">
        <contact id="92" state="active" event="registered"
                 duration-registered="1" expires="3599"
```

```
                callid="843817637684230@998sdasdh09" cseq="1826">
        <uri>sip:user@domain.com</uri>
        <unknown-param name="+sip.instance">
           "&lt;urn:uuid: f81d4fae-7dec-11d0-a765-00a0c91e6bf6&gt;"
        </unknown-param>
        <gr:pub-gruu uri=" sip:@domain.com;gr=urn:
  uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6"/>
        <gr:temp-gruu uri="sip:tgruu.7hs==jd7vnzga5w7fajsc7-ajd6fabz0f8g5@domain.com;gr"
first-cseq="1826"/>
        <unknown-param name="+g.oma.pusheventapp">"wml.ua,mms.ua,wv.ua,loc.ua,syncml.dm"
        </unknown-param>
     </contact>
   </registration>
 </reginfo>
```

8. The Push Sender Agent sends a 200 OK response to the NOTIFY

   200 OK as sent to the SIP/IP Core

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP registrar.domain.com;branch=z9hG4bKnashds7
To: <sip:registrar.domain.com;tag=262281>
From: <sip:psa.domain.com;tag=27182>
CSeq: 633 NOTIFY
Contact: <sip:psa.domain.com>
Content-Length: 0
```

B.1.3   Registration Event Notification in non-IMS Networks

This flow shows the initial registration or re-registration of a Push Receiver Agent with a SIP/IP Core network, and delivery of the registration event to the Push Sender Agent.
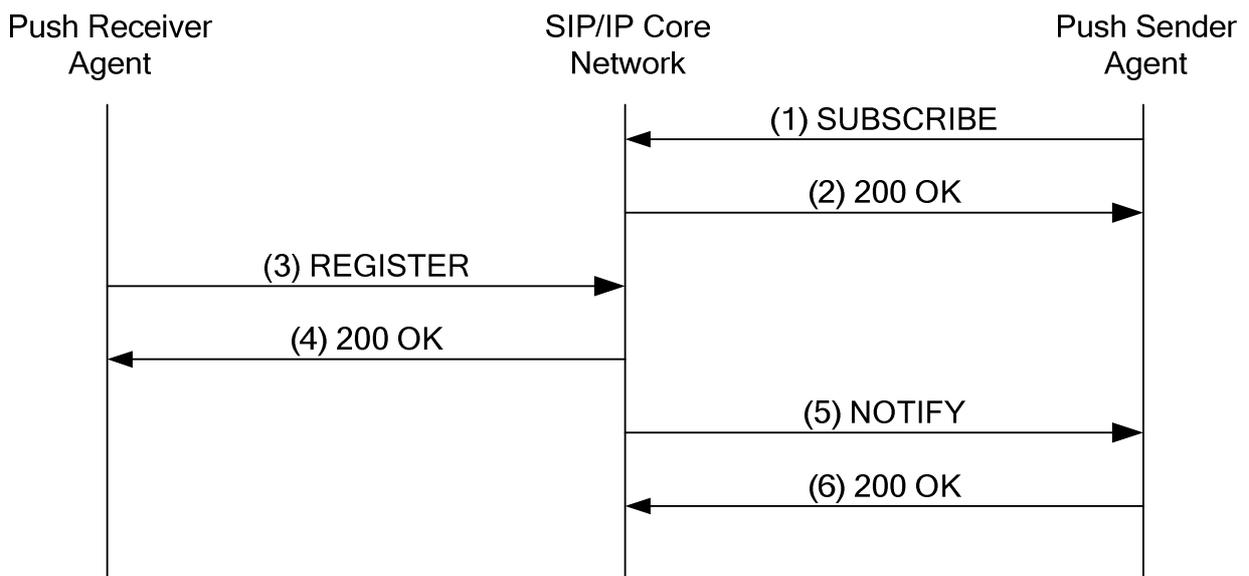
Figure 4: Registration Event Notification in non-IMS Networks

1. Same as "Registration Event Notification in IMS Networks" step 5.

2. Same as "Registration Event Notification in IMS Networks" step 6.

3. Same as "Third Party Registration in IMS Networks" step 1, except that the IMS ICSI is not included.

4. Same as "Third Party Registration in IMS Networks" step 2. , except that the IMS ICSI is not included.

5. Same as "Registration Event Notification in IMS Networks" step 7, except that the IMS ICSI is not included.

6. Same as "Registration Event Notification in IMS Networks" step 8, except that the IMS ICSI is not included.

B.2 Capability Delivery and Resource Negotiation

This flow shows that after registration to SIP/IP Core Push Receiver Agent provides its capability to Push Sender Agent and requests the push resource that Push Sender Agent supports using SIP OPTIONS request. GRUU is included in the Contact header in the SIP OPTIONS request. Push Sender Agent using this capability is notified that the Push Receiver Agent is available and requesting push resources.
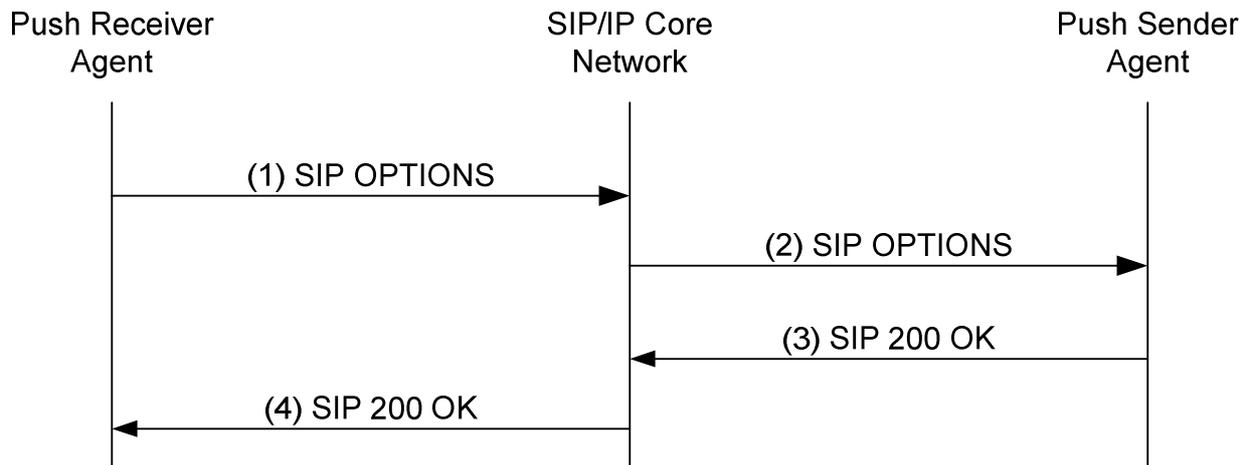
Figure 5: Flow for Capability Delivery and Resource Negotiation via SIP OPTIONS

1. A Push Receiver Agent sends a SIP OPTIONS request to the SIP/IP Core network. The GRUU of Push Receiver Agent received during registration to SIP/IP Core is included in the Contact header.

   ```
   OPTIONS sip:psa@domain.com SIP/2.0
   Via: SIP/2.0/UDP pra1.domain.com;branch=z9hG4bK123dsghds
   Max-Forwards: 70
   From: "Bob" <sip:user@domain.com;tag=456248
   To: <sip:psa.domain.com>
   P-Preferred-Identity: sip:user@domain.com
   P-Preferred-Service: urn:urn-7:3gpp-service.ims.icsi.omapush
   Contact: <sip:user@domain.com>;gr=urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6>
   ;+g.oma.pusheventapp="dcd.ua,loc.ua,wml.ua,syncml.dm"
   Accept-Contact: *;+g.oma.pusheventapp="dcd.ua,loc.ua,wml.ua,syncml.dm"
   Supported: gruu
   Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, MESSAGE
   User-Agent: softphone Beta1.5
   Content-Type: message/external-body; access-type="URL"; expiration="Sat, 5 July 2008
   09:00:00 GMT"; URL="http://DcdProfile.vendor.com/uaprof/N6230ir200.xml"

   Content-Type: application/rdf+xml
   Content-Disposition: attachment
   ```

2. The SIP/IP Core network forwards the SIP OPTIONS to the Push Sender Agent.

   ```
   OPTIONS sip:psa@domain.com SIP/2.0
   Via: SIP/2.0/TCP proxy.domain.com;branch=z9hG4bK776sgdkse
   Via: SIP/2.0/UDP pra1.domain.com;branch=z9hG4bK123dsghds
   ```

```
Max-Forwards: 69
From: "Bob" <sip:user@domain.com;tag=456248
To: <sip:psa.domain.com>
P-Asserted-Identity: sip:user@domain.com
P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.omapush
Contact: <sip:user@domain.com>;gr=urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6>
;+g.oma.pusheventapp="dcd.ua,loc.ua,wml.ua,syncml.dm"
Accept-Contact: *;+g.oma.pusheventapp="dcd.ua,loc.ua,wml.ua,syncml.dm"
Supported: gruu
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, MESSAGE
User-Agent: softphone Beta1.5
Content-Type: message/external-body; access-type="URL"; expiration="Sat, 5 July 2008
09:00:00 GMT"; URL="http://DcdProfile.vendor.com/uaprof/N6230ir200.xml"

Content-Type: application/rdf+xml
Content-Disposition: attachment
```

3. The Push Sender Agent retrieves capability of the Push Receiver Agent including GRUU and finds out Push Receiver Agent is available to the SIP/IP Core. The Push Sender Agent acknowledges with a SIP 200 "OK" to the SIP/IP Core network.

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP proxy.domain.com;branch=z9hG4bK776sgdkse
Via: SIP/2.0/UDP pra1.domain.com;branch=z9hG4bK123dsghds
P-Asserted-Identity: sip:psa@domain.com
From: <sip:psa.domain.com>;tag=49583
To: "Bob" <sip:user@domain.com>;tag=456248
Contact: <sip:psa@domain.com>;+g.oma.pusheventapp="dcd.ua"
Supported: gruu
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, MESSAGE
Accept: message/external-body, application/rdf+xml
Call-ID: 723459815768908@998sdasdh09
CSeq: 1 OPTIONS
Content-Length: 0
```

4. The SIP/IP Core network forwards the SIP "OK" to the Push Receiver Agent.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pra1.domain.com;branch=z9hG4bK123dsghds
P-Asserted-Identity: sip:psa@domain.com
From: <sip:psa.domain.com>;tag=49583
To: "Bob" <sip:user@domain.com>;tag=456248
Contact: <sip:psa@domain.com>;+g.oma.pusheventapp="dcd.ua"
Supported: gruu
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, MESSAGE
Accept: message/external-body, application/rdf+xml
Call-ID: 723459815768908@998sdasdh09
CSeq: 1 OPTIONS
Content-Length: 0
```

B.3   SIP MESSAGE sample flow

This flow shows the delivery of Push Content via the SIP MESSAGE method. The example shows the delivery of Push Content to a terminal MMS client, by a hypothetical OMA Push enabler which has been enhanced for support of SIP Push.
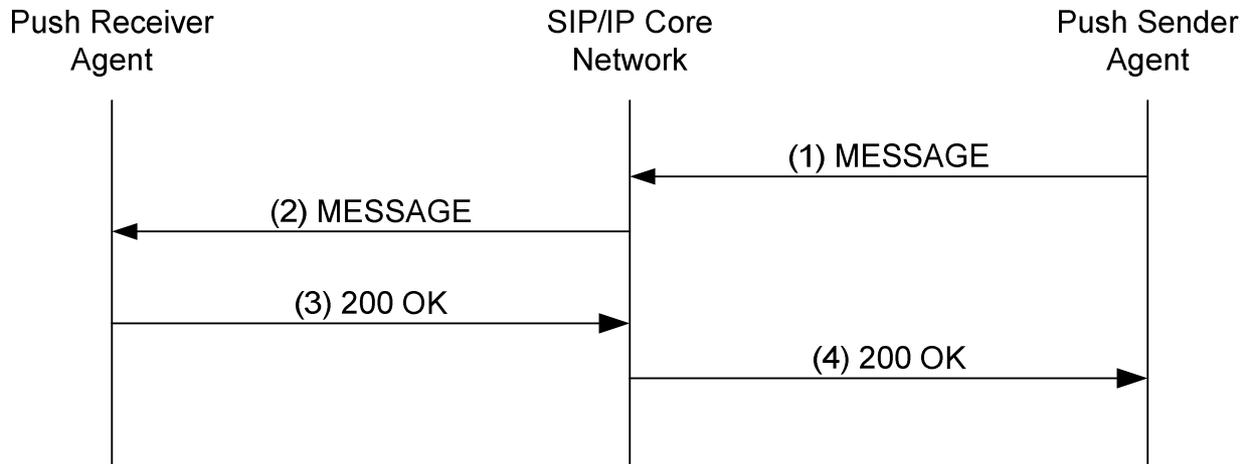
Figure 6: Push Content Delivery via SIP MESSAGE

1.   A Push Sender Agent sends a SIP MESSAGE request to the Push Receiver Agent. The GRUU URI parameter ("gr=...") is included if a GRUU was assigned during registration. The Call-ID is set per the Call-ID assigned by the Push Receiver Agent when it registered. The requirements of the implementing enabler/application define the values of the Accept-Contact feature tags for the ICSI (`g.3gpp.icsi-ref`) and Push Resource Identifier (`g.oma.pusheventapp`), and the Content-Type. Note: the P-Asserted-Identity is included in SIP MESSAGE request because the Push Sender Agent is assumed to be trusted by the SIP/IP Core network.

SIP MESSAGE from the Push Sender Agent

```
MESSAGE sip:user@domain.com;gr=urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6 SIP/2.0

Via: SIP/2.0/TCP psa.domain.com;branch=z9hG4bK776sgdkse

Max-Forwards: 70

P-Asserted-Identity: sip:psa@domain.com

From: <sip:psa.domain.com>;tag=49583

To: "Bob" <sip:user@domain.com;tag=456248 ;gr= urn:uuid:f81d4fae-7dec-11d0-a765-
00a0c91e6bf6>

P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.omapush

Supported: gruu

Accept-Contact: *; +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.omapush",
+g.oma.pusheventapp="mms.ua"

Call-ID: 723459815768908@998sdasdh09

CSeq: 1 MESSAGE

Content-Type: application/vnd.oma.push

Content-Length: ..
```

(Push content…)

2. The SIP MESSAGE request is forwarded to the Push Receiver Agent by the SIP/IP Core network. If a GRUU was included, the message is delivered to the specified terminal; otherwise the forking procedures of the SIP/IP Core network determine the target terminal.

    SIP MESSAGE as forwarded to the Push Receiver Agent

    ```
    MESSAGE sip:user@domain.com;gr=urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6 SIP/2.0
    Via: SIP/2.0/UDP proxy.domain.com;branch= z9hG4bK123dsghds
    Via: SIP/2.0/TCP psa.domain.com;branch=z9hG4bK776sgdkse;received=192.0.2.4
    Max-Forwards: 69
    P-Asserted-Identity: sip:psa@domain.com
    P-Called-Party-ID: sip:user@domain.com;gr=urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6"
    From: <sip:psa.domain.com>;tag=49583
    To: Bob <sip:user@domain.com;tag=456248 ;gr=urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6>
    Supported: gruu
    Accept-Contact: *; +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.omapush";
    +g.oma.pusheventapp="mms.ua"
    Call-ID: 723459815768908@998sdasdh09
    CSeq: 1 MESSAGE
    Content-Type: application/vnd.oma.push
    Content-Length: ..

    (Push content…)
    ```

3. The Push Receiver Agent returns a SIP 200 "OK" to the SIP/IP Core network.

    SIP 200 OK as sent to the SIP/IP Core

    ```
    SIP/2.0 200 OK
    Via: SIP/2.0/UDP proxy.domain.com;branch=z9hG4bK123dsghds;received=192.0.2.1
    Via: SIP/2.0/TCP psa.domain.com;branch=z9hG4bK776sgdkse;received=192.0.2.4
    From: <sip:psa.domain.com>;tag=49583
    To: Bob <sip:user@domain.com>;tag=456248
    Supported: gruu
    Call-ID: 723459815768908@998sdasdh09
    CSeq: 1 MESSAGE
    Content-Length: 0
    ```

4. The SIP/IP Core network forwards the SIP 200 "OK" to the Push Sender Agent

    SIP 200 OK as forwarded to the Push Sender Agent

    ```
    SIP/2.0 200 OK
    Via: SIP/2.0/TCP psa.domain.com;branch=z9hG4bK776sgdkse;received=192.0.2.4
    P-Asserted-Identity: user@domain.com
    From: <sip:psa.domain.com>;tag=49583
    To: Bob <sip:user@domain.com>;tag=456248
    Supported: gruu
    Call-ID: 723459815768908@998sdasdh09
    CSeq: 1 MESSAGE
    Content-Length: 0
    ```

B.4 SIP INVITE / MSRP Sample Flow

This flow shows the delivery of Push Content via the SIP INVITE method, using MSRP. The example shows the delivery of Push Content to a terminal browser, by a hypothetical OMA Push enabler which has been enhanced for support of SIP Push.
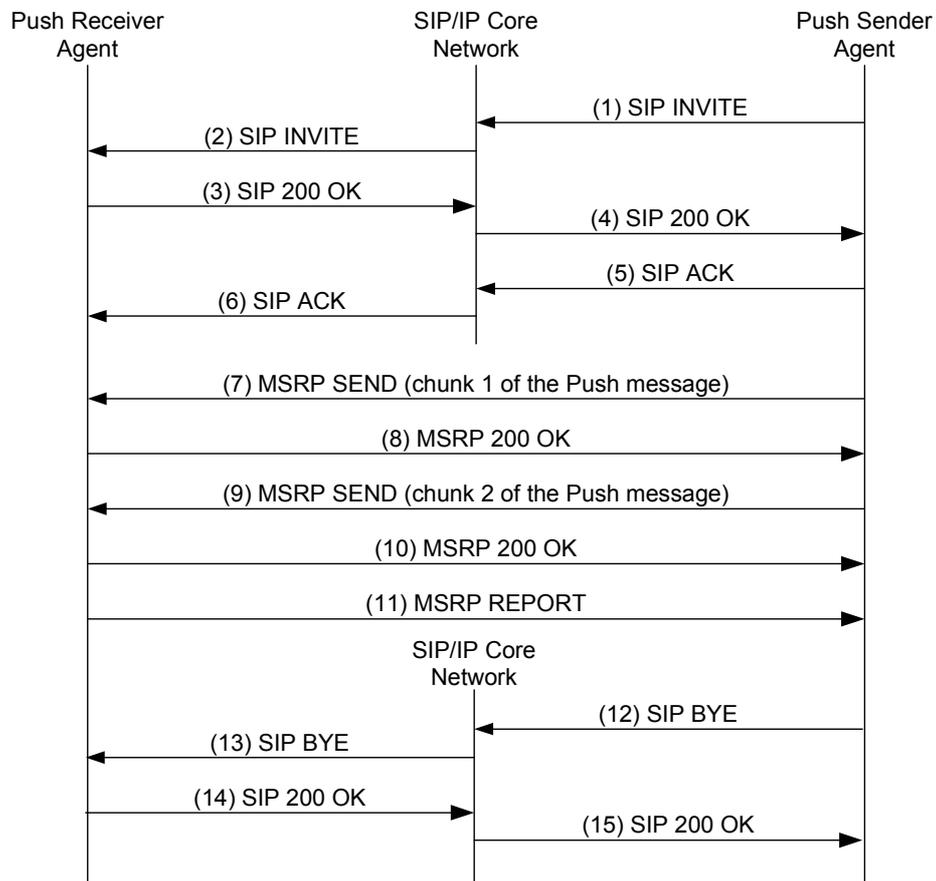
Figure 7: Push Content Delivery via SIP INVITE/MSRP

In this flow, one or more messages may be sent to the Push Receiver Agent in an MSRP session.

1. The Push Sender Agent sends the SIP INVITE to the Push Receiver Agent. The GRUU URI parameter ("gr=…") is included if a GRUU was assigned during registration. The Call-ID is set per the Call-ID assigned by the Push Receiver Agent when it registered. The requirements of the implementing enabler/application define the values of the Accept-Contact feature tags for the ICSI (`g.3gpp.icsi-ref`) and Push Resource Identifier (`g.oma.pusheventapp`). Note: the "s=" line of the SDP body is set as recommended in [RFC3264].

    SIP INVITE as sent to the SIP/IP Core

```
INVITE sip:user@domain.com;gr=urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6 SIP/2.0
Via: SIP/2.0/TCP psa@domain.com;branch=z9hG4bK776sgdkse
Max-Forwards: 70
P-Asserted-Identity: <sip:psa@domain.com>
To: Bob <sip:user@domain.com>
From: <sip:psa.domain.com>;tag=49583
P-Asserted-Service: urn:urn-7:3gpp-service.ims.icsi.omapush
Supported: gruu
```

```
Contact: <sip:psa.domain.com>;+g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-
service.ims.icsi.omapush"; +g.oma.pusheventapp="wml.ua"
Accept-Contact: *; +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.omapush";
+g.oma.pusheventapp="wml.ua"
Call-ID: 390458126334871@998sdasdh09
CSeq: 198456 INVITE
Content-Type: application/sdp
Content-Length: ..

v=0
o=psa 2890844526 2890844526 IN IP4 domain.com
s=-
c=IN IP4 psa.domain.com
t=0 0
m=message 7654 TCP/MSRP *
a=setup: passive
a=sendonly
a=accept-types: application/vnd.oma.push
a=path: msrp://psa.domain.com:7654/a3cd20b8ae7f;tcp
a=file-transfer-id: a3cd20b8ae7f
a=file-selector: name:"pushid-a3cd20b8ae7f" type:application/vnd.oma.push
 size:4092 hash:sha-1:72:24:5F:E8:65:3D:DA:F3:71:36:2F:86:D4:71:91:3E:E4:A2:CE:2E
```

2. The SIP/IP Core network forwards the SIP INVITE to the Push Receiver Agent.

    SIP INVITE as forwarded to the Push Receiver Agent
```
INVITE sip:user@domain.com;gr=urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6 SIP/2.0
Via: SIP/2.0/TCP proxy.domain.com;branch= z9hG4bK123dsghds
Via: SIP/2.0/TCP psa.domain.com;branch=z9hG4bK776sgdkse
Max-Forwards: 69
P-Asserted-Identity: <sip:psa@domain.com>
P-Called-Party-ID: sip:user@domain.com;gr=urn:uuid:f81d4fae-7dec-11d0-a765-
00a0c91e6bf6"
To: Bob <sip:user@domain.com>
From: <sip:psa.domain.com>;tag=49583
Supported: gruu
Contact: <sip:psa.domain.com>;+g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-
service.ims.icsi.omapush"; +g.oma.pusheventapp="wml.ua"
Accept-Contact: *; +g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.omapush";
+g.oma.pusheventapp="wml.ua"
Call-ID: 390458126334871@998sdasdh09
CSeq: 198456 INVITE
Content-Type: application/sdp
Content-Length: ..

v=0
o=psa 2890844526 2890844526 IN IP4 domain.com
s=-
c=IN IP4 psa.domain.com
t=0 0
m=message 7654 TCP/MSRP *
```

```
a=setup: passive
a=sendonly
a=accept-types: application/vnd.oma.push
a=path: msrp://psa.domain.com:7654/a3cd20b8ae7f;tcp
a=file-transfer-id: a3cd20b8ae7f
a=file-selector: name:"pushid-a3cd20b8ae7f" type:application/vnd.oma.push
 size:4092 hash:sha-1:72:24:5F:E8:65:3D:DA:F3:71:36:2F:86:D4:71:91:3E:E4:A2:CE:2E
```

3. The Push Receiver Agent analyses the SDP parameters and returns the agreed parameters by sending a SIP 200 "OK" response to the SIP/IP Core network. If the Push Receiver Agent for some reason does not want to accept the Push content, it sets the port number of the media ("m=") line associated with the file to zero.

SIP 200 OK <u>as sent to the SIP/IP Core</u>

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP proxy.domain.com;branch=z9hG4bK123dsghds;received=192.0.2.1
Via: SIP/2.0/TCP psa@domain.com;branch=z9hG4bK776sgdkse;received=192.0.2.4
To: Bob <sip:user@domain.com;tag=456248
From: <sip:psa.domain.com>;tag=49583
Supported: gruu
Call-ID: 390458126334871@998sdasdh09
CSeq: 198456 INVITE
Contact: <sip:user@domain.com;gr=urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6>;
+g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.omapush";
+g.oma.pusheventapp="wml.ua"
Content-Type: application/sdp
Content-Length: ..


v=0
o=bob 2890844656 2890844656 IN IP4 domain.com
s=-
c=IN IP4 userpc.domain.com
t=0 0
m=message 8888 TCP/MSRP *
a=setup: active
a=recvonly
a=accept-types: application/vnd.oma.push
a=path: msrp://userpc.domain.com:8888/9di4ea;tcp
a=file-transfer-id: a3cd20b8ae7f
a=file-selector: name:"pushid-a3cd20b8ae7f" type:application/vnd.oma.push
 size:4092 hash:sha-1:72:24:5F:E8:65:3D:DA:F3:71:36:2F:86:D4:71:91:3E:E4:A2:CE:2E
```

4. The SIP/IP Core network forwards the SIP 200 "OK" response to the Push Sender Agent.

SIP 200 OK <u>as forwarded to the Push Sender Agent</u>

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP psa@domain.com;branch=z9hG4bK776sgdkse
P-Asserted-Identity: user@domain.com
To: Bob <sip:user@domain.com>;tag=456248
Supported: gruu
From: <sip:psa.domain.com>;tag=49583
Call-ID: 390458126334871@998sdasdh09
CSeq: 198456 INVITE
```

```
Contact: <sip:user@domain.com;gr=urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6>;
+g.3gpp.icsi-ref="urn%3Aurn-7%3A3gpp-service.ims.icsi.omapush";
+g.oma.pusheventapp="wml.ua"
```
```
Content-Type: application/sdp
Content-Length: ..


v=0
o=bob 2890844656 2890844656 IN IP4 domain.com
c=IN IP4 userpc.domain.com
t=0 0
m=message 8888 TCP/MSRP *
a=setup: active
a=recvonly
a=accept-types: application/vnd.oma.push
a=path: msrp://userpc.domain.com:8888/9di4ea;tcp
a=file-transfer-id: a3cd20b8ae7f
a=file-selector: name:"pushid-a3cd20b8ae7f" type:application/vnd.oma.push
 size:4092 hash:sha-1:72:24:5F:E8:65:3D:DA:F3:71:36:2F:86:D4:71:91:3E:E4:A2:CE:2E
```

5. The Push Sender Agent acknowledges the SIP 200 "OK" response with a SIP ACK request sent to the SIP/IP Core. network

    SIP ACK <u>as sent to the SIP/IP Core</u>

    ```
    ACK sip:user@domain.com;gr=urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6 SIP/2.0
    Via: SIP/2.0/TCP psa.domain.com;branch=z9hG4bK776sgdkse
    Max-Forwards: 70
    To: Bob <sip:user@domain.com>;tag=456248
    From: <sip:psa.domain.com>;tag=49583
    Call-ID: 390458126334871@998sdasdh09
    CSeq: 198456 ACK
    Content-Length: 0
    ```

6. The SIP/IP Core network forwards the SIP ACK request to the Push Sender Agent.

    SIP ACK <u>as forwarded to the Push Receiver Agent</u>

    ```
    ACK sip:user@192.0.2.4  SIP/2.0
    Via: SIP/2.0/TCP proxy.domain.com;branch= z9hG4bK123dsghds
    Via: SIP/2.0/TCP psa.domain.com;branch=z9hG4bK776sgdkse
    Max-Forwards: 69
    To: Bob <sip:user@domain.com>;tag=456248
    From: <sip:psa.domain.com>;tag=49583
    Call-ID: 390458126334871@998sdasdh09
    CSeq: 198456 ACK
    Content-Length: 0
    ```

7. In this example, the Push Content is large enough to require chunking over multiple MSRP SEND requests, and a delivery success report is requested by Push Sender Agent. The Push Sender Agent opens a TCP connection to the Push Receiver Agent, and then sends the first chunk of data in a MSRP SEND request to the Push Receiver Agent using the MSRP session, including the Success Report header.

    MSRP SEND <u>as sent to the Push Receiver Agent</u>

    ```
    MSRP d93kswow SEND
    ```

```
To-Path: msrp://userpc.domain.com:8888/9di4ea;tcp
From-Path: msrp://psa.domain.com:7654/a3cd20b8ae7f;tcp
Message-ID: 12339sdqwer
Byte-Range: 1-2048/4385
Success-Report: yes
Content-Type: application/vnd.oma.push


(first chunk of Push content)
-------d93kswow+
```

8. The Push Receiver Agent responds with an MSRP 200 "OK" to the first MSRP SEND request to the Push Sender Agent using the MSRP session. Note that MSRP supports pipelining of SEND requests, i.e. the sender does not have to wait for a 200 "OK" to send the next chunk. This example shows the case in which pipelining is not used.

MSRP 200 OK as sent to the Push Sender Agent

```
MSRP d93kswow 200 OK
To-Path: msrp://psa.domain.com:7654/a3cd20b8ae7f;tcp
From-Path: msrp://userpc.domain.com:8888/9di4ea;tcp
-------d93kswow$
```

9. The Push Sender Agent sends the next chunk of data in a MSRP SEND request to the Push Receiver Agent using the MSRP session.

MSRP SEND as sent to the Push Receiver Agent

```
MSRP op2nc9a SEND
To-Path: msrp://userpc.domain.com:8888/9di4ea;tcp
From-Path: msrp://psa.domain.com:7654/a3cd20b8ae7f;tcp
Message-ID: 12339sdqwer
Byte-Range: 2049-4385/4385
Success-Report: yes
Content-Type: application/vnd.oma.push


(last chunk of Push content)
------- op2nc9a$
```

10. The Push Receiver Agent responds with an MSRP 200 "OK" on the second MSRP SEND request to the Push Sender Agent using the MSRP session.

MSRP 200 OK as sent to the Push Sender Agent

```
MSRP op2nc9a 200 OK
To-Path: msrp://psa.domain.com:7654/a3cd20b8ae7f;tcp
From-Path: msrp://userpc.domain.com:8888/9di4ea;tcp
------- op2nc9a$
```

NOTE: Steps 9 and 10 will be repeated for the remaining chunks required to deliver the entire message. In this example, two chunks were required.

11. As requested by the Push Sender Agent, the Push Receiver Agent sends a MSRP REPORT conveying the status of the received message.

MSRP REPORT as sent to the Push Sender Agent

```
MSRP dkei38sd REPORT
To-Path: msrp://psa.domain.com:7654/a3cd20b8ae7f;tcp
From-Path: msrp://userpc.domain.com:8888/9di4ea;tcp
```

```
Message-ID: 12339sdqwer
Byte-Range: 1-4385/4385
Status: 000 200 OK
------- dkei38sd$
```

NOTE: In this example, a single file selector was used to delivery a single Push Content item. If multiple file selectors had been included in the SDP offer, steps 7 through 11 may be repeated for delivery of the additional Push Content items.

12. When the Push Sender Agent and the Push Receiver Agent do not need the MSRP session, the Push Sender Agent sends a SIP BYE to SIP/IP Core network to disconnect the MSRP session with the Push Receiver Agent.

SIP BYE as sent to the SIP/IP Core

```
BYE sip:user@domain.com gr=urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6" SIP/2.0
Via: SIP/2.0/TCP psa.domain.com;branch=z9hG4bK776sgdkse
Max-Forwards: 70
To: Bob <sip:user@domain.com>;tag=456248
From: <sip:psa.domain.com>;tag=49583>
Call-ID: 390458126334871@998sdasdh09
CSeq: 198457 BYE
Content-Length: 0
```

13. The SIP/IP Core network forwards the SIP BYE request to the Push Receiver Agent.

SIP BYE as forwarded to the Push Receiver Agent

```
BYE sip:user@192.0.2.4  SIP/2.0
Via: SIP/2.0/TCP proxy.domain.com;branch= z9hG4bK123dsghds
Via: SIP/2.0/TCP psa.domain.com;branch=z9hG4bK776sgdkse
Max-Forwards: 69
To: Bob <sip:user@domain.com>;tag=456248
From: <sip:psa.domain.com>;tag=49583
Call-ID: 390458126334871@998sdasdh09
CSeq: 198457 BYE
Content-Length: 0
```

14. The Push Sender Agent responds to the Push Receiver Agent with SIP 200 "OK".

SIP 200 OK as sent to the SIP/IP Core

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP proxy.domain.com;branch=z9hG4bK123dsghds;received=192.0.2.1
Via: SIP/2.0/TCP psa.domain.com;branch=z9hG4bK776sgdkse
To: Bob <sip:user@domain.com>;tag=456248
From: <sip:psa.domain.com>;tag=49583
Call-ID: 390458126334871@998sdasdh09
CSeq: 2 BYE
Content-Length: 0
```

15. The SIP/IP Core network forwards the SIP 200 "OK" to the Push Sender Agent.

SIP 200 OK as forwarded to the Push Sender Agent

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP psa.domain.com;branch=z9hG4bK776sgdkse
```

```
To: Bob <sip:user@domain.com>;tag=456248
From: <sip:psa.domain.com>;tag=49583
Call-ID: 390458126334871@998sdasdh09
CSeq: 2 BYE
Content-Length: 0
```

Appendix C.   Response Code Interpretation   (Informative)

When the Push Receiver Agent receives a SIP request, it will generate a SIP response. The following Table 1 shows a mapping between SIP response codes to the SIP Push response scenarios. These response scenarios represent an interpretation of SIP response codes and guidance for applications. Unless stated otherwise in this document, response is interpreted according to the rules in the SIP specification [RFC3261], and [RFC3428.

Scenario	SIP Push Method	SIP Response code	Description
1	SIP MESSAGE	- 200 OK	Push request accepted
2	SIP MESSAGE	- 400 bad request - 500 server internal error - 503 server unavailable - 603 decline	Push request rejected without specific causes, retries allowed
3	SIP MESSAGE	- 403 Forbidden - 604 does not exist anywhere	Push request rejected without specific causes, no retries
4	SIP MESSAGE	- 408 request timeout	Push request rejected because the push message cannot be delivered to the intended destination
5	SIP MESSAGE	- 500 Server internal error.	Push request rejected because the push message is discarded due to resource shortage
6	SIP MESSAGE	- 415 Unsupported media type	Push request rejected, because the content type cannot be processed

Table 1: Response code interpretation

Appendix D.   Interoperability with ICSI and IARI   (Informative)

D.1   Introduction

Enablers using this specification to enable push services may be deployed within 3GPP IMS networks. In this case the implementing enabler is considered a communication service which is identified through an IMS Communication Service Identifier (ICSI) within the IMS network.

More than one application of the same type may run over the same communication service. The IMS application reference identifier (IARI) is used to address the application instance. For example it might be possible to use two MMS applications on the same device.

An enabler utilizing SIP Push may itself have different resources (see Section 7.1) which needs to be addressed. For example in a hypothetical enabler "OMA Push 2.x", an enhancement of the OMA Push enabler as a SIP Push implementing enabler, could provide Push message delivery service to a variety of OMA enabler clients, e.g. browsers, MMS clients, IM clients, device management clients, as currently supported by the OMA Push enabler. To use the services of the OMA Push 2.x enabler, other enablers and applications are assigned a Push Application ID through OMNA. The OMA Push 2.x enabler may use the SIP Push-defined Push Resource Identifier to identify the target application (equivalent in purpose to the Push Application ID), with the IARI identifying a particular instance of a OMA Push 2.x Client operating on the device, and the ICSI identifying the OMA Push service itself.

The diagram below illustrates the addressing of applications via the Push Resource Identifier.
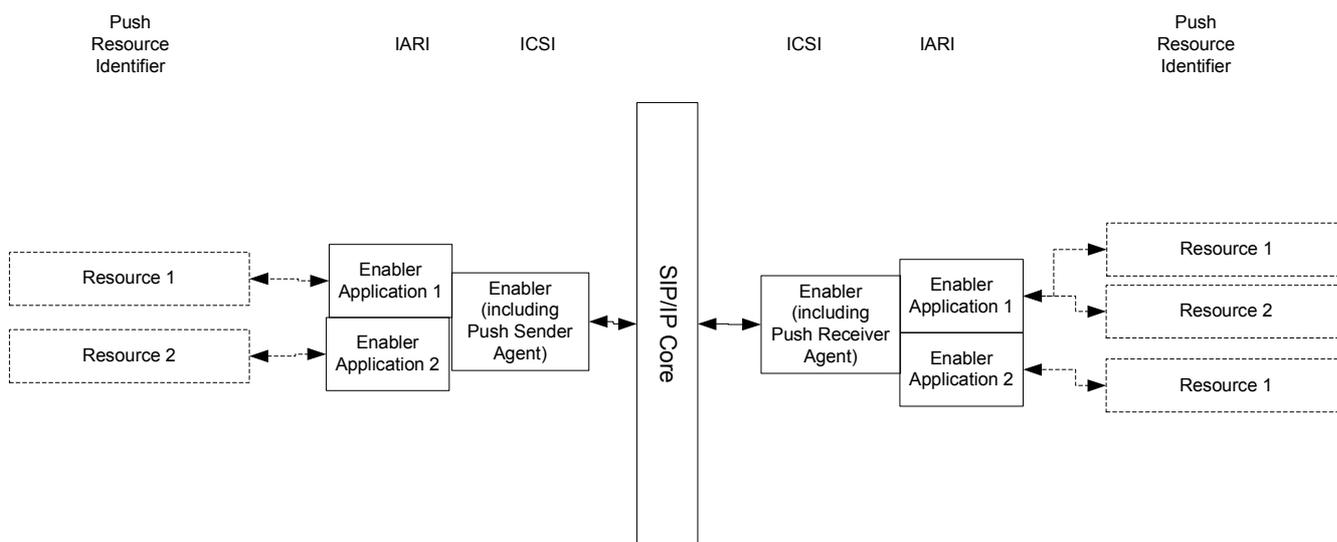

Figure 8: ICSI, IARI and Push Resource Identifier usage

Only the usage of the Push Resource Identifier is specified in this specification. However, if an enabler is deployed in an IMS environment the ICSI, IARI and Push Resource Identifier may be used together.

D.2   Examples

This simple Example shows the use of ICSI, IARI and the Push Resource Identifier coexistent within one push request utilizing the SIP MESSAGE method. The IARI and the Push Event Id are sent as feature tag parameters within the Accept-Contact header. The ICSI is sent in the P-Preferred-Service/P-Asserted-Service header.

```
MESSAGE sip:user2@domain.com SIP/2.0
Via: SIP/2.0/TCP user1pc.domain.com;branch=z9hG4bK776sgdkse
Max-Forwards: 70
P-Preferred-Identity: "John Doe" <sip:john.doe@home1.net>
From: sip:user1@domain.com;tag=49583
To: sip:user2@domain.com

Accept-Contact; +g.ims.app_ref="<urn:urn-7:3gpp-application.mmsua >"; + g.oma.pusheventappp="mms.ua"
Call-ID: asd88asd77a@1.2.3.4
CSeq: 1 MESSAGE
Content-Type: text/plain
P-Preferred-Service: urn:urn-7.push
Content-Length: 18
```


```
SIP/2.0 200 OK
Via: SIP/2.0/TCP proxy.domain.com;branch=z9hG4bK123dsghds;received=192.0.2.1
Via: SIP/2.0/TCP user1pc.domain.com;branch=z9hG4bK776sgdkse;received=1.2.3.4
P-Asserted-Identity: "John Doe" <sip:john.doe@home1.net> From:
sip:user1@domain.com;tag=49394
To: sip:user2@domain.com;tag=ab8asdasd9
Call-ID: asd88asd77a@1.2.3.4
CSeq: 1 MESSAGE
P-Asserted-Service: urn:urn-7.push
Content-Length: 0
```

Appendix E.   SIP/IP Core Network Considerations

The SIP Push Architectural Model, specified in [PushSIPArch], specifies the SIP/IP Core network functional requirements for using SIP Push. Two SIP/IP network architectures that meet these requirements are considered by this version of the document. They include the 3rd Generation Partnership Project (3GPP) IP Multimedia Subsystem (IMS) and the Third Generation Partnership Project 2 (3GPP2) Multimedia Domain (MMD). Additional requirements and considerations for implementing SIP Push within these architectures are specified in the following sub-section. Other SIP/IP Core network Architectural considerations may be added in the future.

E.1   3GPP IMS and 3GPP2 MMD Network Architectures

Within the context of 3GPP IMS and 3GPP2 MMD networks, SIP Push should be considered as the primary mechanism for push-based services. When the SIP/IP Core network complies with the 3GPP IMS or 3GPP2 MMD specifications, the following additional requirements and considerations in the following subsections apply.

E.1.1   Architecture Compliance

The Push Sender Agent and Push Receiver Agent SHALL comply with the 3GPP IMS or 3GPP2 MMD requirements, mechanisms and procedures, such as session establishment, according to rules and procedures of [3GPP TS 24.229] and [3GPP2 X.S0013-004-A], respectively. Specifically:

- The P-1 reference point defined for the Push Receiver Agent SHALL conform to the Gm reference point when the Push Receiver Agent is implemented on the UE, or to the ISC interface when the Push Receiver Agent is implemented on an Application Server, as specified in [3GPP TS 23.228] and [3GPP2 X.S0013-002-A], respectively.

- The P-2 reference point defined for Push Sender Agent shall conform to the ISC interface as defined in [3GPP TS 23.228] and [3GPP2 X.S0013-002-A], respectively.

E.1.2   Registration Procedures

In order for the Push Sender Agent to be aware of a registration request (SIP REGISTER) from the Push Receiver Agent, when implemented on the UE, the SIP/IP Core network can initiate a third-party REGISTER request upon receiving a registration request from the Push Receiver Agent. Such third party registrations can be triggered based on filter criteria set for REGISTER requests that indicate support for SIP Push as specified in [3GPP TS 23.228] and [3GPP2 X.S0013-002-A]. Alternatively, the Push Sender Agent can subscribe to the "reg" event package as specified in [3GPP TS 24.229] and [3GPP2 X.S0013-002-A].

E.1.3   Security Considerations

The 3GPP IMS and 3GPP2 MMD Network Architectures provide for mutual authentication and integrity protection between the Push Receiver Agent (when implemented on a UE) and the Push Sender Agent (if implemented as part of a trusted network element), as well as confidentiality protection of SIP signalling. Access network security requirements to accomplish these are specified for 3GPP IMS and 3GPP MMD in [3GPP TS 33.203] and [3GPP2 S.R0086-0], respectively. Inter- and Intra-domain security procedures for accomplishing secure communication between trusted network elements is specified in [3GPP TS 33.210] for 3GPP IMS and [3GPP2 S.R0086-0] for 3GPP MMD.

Specific requirements are listed below:

- Push Sender Agents (in the role of an Application Server) and Push Receiver Agents (implemented on the UE or as part of an Application Server) in 3GPP IMS networks SHALL comply with all applicable security requirements and procedures, such as mutual authentication, in [3GPP TS 33.203], [3GPP TS 33.210] and [3GPP TS 24.229] for 3GPP IMS, and [3GPP2 S.R0086-0] and [3GPP2 X.S0013-002-A] for 3GPP MMD.

- When the Push Sender Agent, in the role of an IMS Application Server, is not part of a trusted network element (intra-domain security is not sufficient), it SHOULD authenticate and establish secure communications to Proxies or Push Receiver Agent as specified in [3GPP TS 33.210] for 3GPP IMS and [3GPP2 S.R0086-0] for 3GPP MMD.

- It SHALL be possible to assert the Push Sender Agent when it is acting as the originating user agent, by the SIP/IP Core network, as specified in [3GPP TS 24.229] and [3GPP2 X.S0013-002-A] for 3GPP IMS and 3GPP2 MMD, respectively.

Appendix F.   Static Conformance Requirements

The notation used in this appendix is specified in [IOPPROC].

F.1   Client Conformance Requirements

The table below enumerates the client conformance requirements. A client being a Push Receiver Agent wishes to receive Push Content.

Item	Function	Reference	Requirements
SIPPUSH-PRA-C-001-M	Support receive Push Content from the Push Sender Agent	8	(SIPPUSH-PRA-C-003-O OR SIPPUSH-PRA-C-004-O)
SIPPUSH-PRA-C-002-M	Register to the SIP/IP Core network	6	
SIPPUSH-PRA-C003-O	SIP MESSAGE method (Pager Mode)	8.1.1	
SIPPUSH-PRA-C004-O	SIP INVITE & MSRP methods (Session-Mode)	8.2.1	
SIPPUSH-PRA-C005-O	Capability Delivery (SIP OPTIONS method)	5.1	

F.2   Server Conformance Requirements

The table below enumerates the server conformance requirements. A server being a Push Sender Agent wishes to send content.

Item	Function	Reference	Requirements
SIPPUSH-PSA-S-001-M	Creating push request to deliver the content to the push receiver agent	8	(SIPPUSH-PSA-S-002-O, OR SIPPUSH-PSA-S-003-O)
SIPPUSH-PSA-S-002-O	SIP MESSAGE method (Pager Mode)	8.1.2	
SIPPUSH-PSA-S-003-O	SIP INVITE & MSRP methods (Session-Mode)	8.2.2	
SIPPUSH-PSA-S-004-O	Capability Reception (SIP OPTIONS method)	5.2	