Network Working Group
Internet-Draft
Expires: November 3, 2007

K. Drage
Alcatel-Lucent
May 2, 2007

# A Session Initiation Protocol (SIP) Extension for the Identification of Services

draft-drage-sipping-service-identification-00

## Status of this Memo



## Copyright Notice

Abstract

This document describes private extensions to the Session Initiation Protocol (SIP) that enable a network of trusted SIP servers to assert the service of authenticated users. The use of these extensions is only applicable inside an administrative domain with previously agreed-upon policies for generation, transport and usage of such information. This document does NOT offer a general service identification model suitable for use between different trust domains, or use in the Internet at large.

## 1. Introduction

This document describes private extensions to the Session Initiation Protocol (SIP) that enable a network of trusted SIP servers to assert the service and for users entitled to that service. The use of these extensions is only applicable inside an administrative domain with previously agreed-upon policies for generation, transport and usage of such information. This document does NOT offer a general service model suitable for use between different trust domains, or use in the Internet at large.

OPEN ISSUE: At some point in this document, we need to define what we mean by the term service. It is hoped that the proposed SIPPING charter item will provide sufficient information to tie a definition to a common concept within that document.

During a session setup proxies may need to understand what service the request is related to in order to know what application server to contact or other service logic to invoke. The SIP INVITE request contains all of the information necessary to determine the service. However, the calculation of the service may be computational and database intensive. For example, a given trust domain's definition of a service might include request authorization. Moreover the analysis may require examination of the SDP.

For example, an INVITE request with video SDP directed to a video-on-demand Request-URI could be marked as an IPTV session. An INVITE request with push-to-talk over cellular (PoC) routes could be marked as a PoC session. An INVITE request with a Require header field containing an option tag of "foogame" could be marked as a foogame session.

NOTE: If the information contained within the SIP INVITE request is not sufficient to uniquely identify a service, the remedy is to extend the SIP signalling to capture the missing element.

Open issue: Capture here a reference to the proposed SIPPING document which will explain exactly this.

By providing a mechanism to compute and store the results of the domain specific service calculation, this optimization allows a single trusted proxy to perform an analysis of the request and authorize the requestor's permission to request such a service. The proxy may then include a service identifier that relieves other trusted proxies and trusted UAs from performing further duplicate analysis of the request for their service identification purposes. In addition, this extension allows user agent clients outside the trust domain to provide a hint of the requested service.

This extension does not provide for the dialog or transaction to be rejected if the service is not supported end-to-end. SIP provides other mechanisms, such as the option-tag and use of the Require and Proxy-Require header fields, where such functionality is required. Rather no service identification exists and the session proceeds as if no specific service had been identified, of the basis of information contained in SDP and in other SIP header fields.

This mechanism is specifically a mechanism to manage the information needs of intermediate routeing devices between the calling user and the user represented by the Request-URI. Between end users, caller preferences and callee capabilities as specified in RFC 3840 [9] and RFC 3841 [10] provide an appropriate mechanism for indicating such service issues. These mechanisms have been extended by draft-rosenberg-sip-app-media-tag [11] to provide further capabilities in this area.

The mechanism proposed in this document relies on a new header field called 'P-Asserted-Service' that contains a URN.

P-Asserted-Service: urn:xxx.exampletelephony.version1-application-v1

A proxy server which handles a request can, after authenticating the originating user in some way (for example: Digest authentication), to ensure that the user is entitled to that service, insert such a P-Asserted-Service header field into the request and forward it to other trusted proxies. A proxy that is about to forward a request to a proxy server or UA that it does not trust removes all the P-Asserted-Service header field values.

The formal syntax for the P-Asserted-Service header is presented in Section 4.1.

This document labels services by means of an informal URN. This provides a hierarchical structure for defining services and subservices, and provides an address that can be resolvable for various purposes outside the scope of this document, e.g. to obtain information about the service so described.

## 2. Applicability Statement

This document describes private extensions to SIP (see RFC 3261 [5]) that enable a network of trusted SIP servers to assert the service of end users or end systems. The use of these extensions is only applicable inside a 'Trust Domain' as defined in Short term requirements for Network Asserted Identity (see RFC 3324 [7]). Nodes in such a Trust Domain are explicitly trusted by its users and end-systems to publicly assert the service of each party. The means by which the network determines the service to assert is outside the scope of this document (though it commonly entails some form of authentication).

The mechanism for defining a trust domain is to provide a certain set of specifications known as 'Spec(T)'. and they specify compliance to that set of specifications. Spec(T) MUST specify behavior as documented in RFC 3323 [6].

This document does NOT offer a general service model suitable for inter-domain use or use in the Internet at large. Its assumptions about the trust relationship between the user and the network may not apply in many applications. For example, these extensions do not accommodate a model whereby end users can independently assert their service by use of the extensions defined here. Furthermore, since the asserted services are not cryptographically certified, they are subject to forgery, replay, and falsification in any architecture that does not meet the requirements of RFC 3324 [7].

The asserted services also lack an indication of who specifically is asserting the service, and so it must be assumed that the Trust Domain is asserting the service. Therefore, the information is only meaningful when securely received from a node known to be a member of the Trust Domain.

Despite these limitations, there are sufficiently useful specialized deployments that meet the assumptions described above, and can accept the limitations that result, to warrant informational publication of this mechanism. An example deployment would be a closed network which emulates a traditional circuit switched telephone network.

## 3. 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 BCP 14, RFC 2119 [2].

Throughout this document requirements for or references to proxy servers or proxy behavior apply similarly to other intermediaries within a Trust Domain (ex: B2BUAs).

The term Trust Domain in this document has the meaning as defined in RFC 3324 [7].

## 4. Syntax of the Header Fields

The following syntax specification uses the augmented Backus-Naur Form (BNF) as described in RFC 2234 [3].

### 4.1. The P-Asserted-Service Header

The P-Asserted-Service header field is used among trusted SIP entities (typically intermediaries) to carry the service information of the user sending a SIP message as it was verified by authentication.

```
PAssertedService = "P-Asserted-Service" HCOLON PAssertedService-value
PAssertedService-value = Service-ID
```

See section 4.4 for the definition of Service-ID in ABNF.

A P-Asserted-Service header field value MUST consist of exactly one textstring. There may be one or two P-Asserted-Identity values. Proxies can (and will) add and remove this header field.

This document adds the following entry to Table 2 of RFC 3261 [5]:

| Header field | where | proxy | ACK | BYE | CAN | INV | OPT | REG | SUB |
|---|---|---|---|---|---|---|---|---|---|
| P-Asserted-Service | R | admr | - | - | - | o | o | - | o |

| Header field | NOT | PRA | INF | UPD | MSG | REF | PUB |
|---|---|---|---|---|---|---|---|
| P-Asserted-Service | - | - | - | - | o | o | o |

The semantics of multiple P-Asserted-Service header fields appearing in the same request is not defined.

### 4.2. The P-Preferred-Service Header

The P-Preferred-Service header field is used from a user agent to a trusted proxy to carry the preferred service of the user sending the SIP message wishes to be used for the P-Asserted-Service field value that the trusted element will insert.

```
PPreferredService = "P-Preferred-Service" HCOLON PPreferredService-value
PPreferredService-value = Service-ID
```

See section 4.4 for the definition of Service-ID in ABNF.

This document adds the following entry to Table 2 of RFC 3261 [5]:

| Header field | where | proxy | ACK | BYE | CAN | INV | OPT | REG | SUB |
|---|---|---|---|---|---|---|---|---|---|
| P-Preferred-Service | R | dr | - | - | - | o | o | - | o |

| Header field | NOT | PRA | INF | UPD | MSG | REF | PUB |
|---|---|---|---|---|---|---|---|
| P-Preferred-Service | - | - | - | - | o | o | o |

The semantics of multiple P-Preferred-Service header fields appearing in the same request is not defined.

### 4.3. Service Definition

Definition of services and their characteristics is outside the scope of this document. Other standards organizations, vendors and operators may define their own services and register them.

A hierarchical structure is defined consisting of service identifiers, subservice identifiers and application identifiers.

OPEN ISSUE: Other material contributed as drafts to the SIPPING group have identified the need to distinguish between service identifiers and application identifiers. This has been added in the syntax below, but it is currently not clear whether it is needed. If the sole purpose is to identify a particular API within the end terminal, then it may well be that the extensions provided by [11] fulfil this purpose within the genuine usage of a media feature tag.

The service and subservice identifiers identify the service as described in section 1.

An application identifier identifies an application that uses a service in order to provide a specific capability to the end-user. The application uses specific service and provides the end user service through the reuse of the SIP communication part of service. The application does not extend the definition of the service. The application identifier identifies the application utilising the service.

IANA maintains a registry of service identifier values that have been assigned. This registry is created by the actions of section 8.2 of this document.

Subservice identifiers are not managed by IANA. It is the responsibility of the organisation that registered the service to manage the subservices.

Application identifiers are not managed by IANA. It is the

responsibility of the organisation that registered the service to manage the applicable applications.

## 4.4. Registration Template

Below, we include the registration template for the URN scheme according to RFC 3406 [8]. The URN scheme is defined as an informal NID.

Namespace ID: urn:xxx

Registration Information: Registration version: 1; registration date: 2007-04-21

Declared registrant of the namespace: TBD

Declaration of syntactic structure: The URN consists of a hierarchical service identifier, with a sequence of labels separated by periods. The left-most label is the most significant one and is called 'top-level service identifier', while names to the right are called 'sub-services'. The set of allowable characters is the same as that for domain names (see RFC 1123 [1]) and a subset of the labels allowed in RFC 3958 [9]. Labels are case-insensitive and MUST be specified in all lower-case. For any given service identifier, labels can be removed right-to- left and the resulting URN is still valid, referring a more generic service. In other words, if a service identifier 'x.y.z' exists, the URNs 'x' and 'x.y' are also valid service identifiers. The service identifier can be followed by one or more application identifiers separated by semi-colons. Similar character rules apply to that for service identifiers. There is no substructure for application identifiers.

```
Service-ID      = "urn:xxx:" urn-service-id
urn-service-id  = top-level *("." sub-service-id)
                      *("-"application-id)
top-level       = let-dig [ *26 let-dig ]
sub-service-id  = let-dig [ *let-dig ]
application-id  = let-dig [ *let-dig ]
let-dig         = ALPHA / DIGIT
```

Note to RFC editor: replace xxx with the assigned 3 numeric digit identifier.

Relevant ancillary documentation: None

Identifier uniqueness considerations: A service identifier identifies a service, indicated in the service registration (see IANA Considerations (Section 8)). Uniqueness is guaranteed by the IANA registration.

Identifier persistence considerations: The service identifier for the same service is expected to be persistent, although there naturally cannot be a guarantee that a particular service will continue to be available globally or at all times.

Process of identifier assignment: The process of identifier assignment is described in the IANA Considerations (Section 8).

Process for identifier resolution: There is no single global resolution service for service identifiers.

Rules for Lexical Equivalence: 'service' identifiers are compared according to case-insensitive string equality.

Conformance with URN Syntax: The BNF in the 'Declaration of syntactic structure' above constrains the syntax for this URN scheme.

Validation mechanism: Validation determines whether a given string is currently a validly- assigned URN (see RFC 3406 [8]). Due to the distributed nature of usage and since not all services are available everywhere, validation in this sense is not possible

Scope: The scope for this URN can be local to a single domain, or may be more widely used.

# 5. Usage of the P-Preferred-Service and P-Asserted-Service header fields

## 5.1. Usage of the P-Preferred-Service and P-Asserted-Service header fields in Requests

### 5.1.1. Procedures at User Agent Clients (UAC)

The UAC MAY insert a P-Preferred-Service in a request that creates a dialog, or a request outside of a dialog. This information can assist the proxies in identifying appropriate service capabilities to apply to the call.

### 5.1.2. Procedures at Intermediate Proxies

A proxy in a Trust Domain can receive a request from a node that it trusts, or a node that it does not trust. When a proxy receives a request from a node it does not trust and it wishes to add a P-Asserted-Service header field, the proxy MUST identify the service appropriate to the capabilities (e.g. SDP) in the request, MAY authenticate the originator of the request (in order to determine whether the user is subscribed for that service), and use the identity which results from this checking and authentication to insert a P-Asserted-Service header field into the request.

If the proxy receives a request from a node that it trusts, it can use the information in the P-Asserted-Service header field, if any, as if it had authenticated the user itself.

If there is no P-Asserted-Identity header field present, a proxy MAY add one containing it using its own analysis of the information contained in the SIP request. If the proxy received the request from an element that it does not trust and there is a P-Asserted-Service header present, the proxy MUST replace that header field contents with a new analysis or remove this header field.

If a proxy forwards a request to a node outside the proxy's trust domain, there MUST NOT be a P-Asserted-Service header field in the forwarded request.

### 5.1.3. Procedures at User Agent Servers (UAS)

For a UAS outside the trust domain, the P-Asserted-Service header is removed before it reachs this entity, therefore there are no procedures for such a device.

However, if a User Agent Server receives a request from a previous element that it does not trust, it MUST NOT use the P-Asserted-

Service header field in any way.

If a UA is part of the Trust Domain from which it received a request containing a P-Asserted-Service header field, then it can use the value freely but it MUST ensure that it does not forward the information to any element that is not part of the Trust Domain.

### 5.2. Usage of the P-Preferred-Service and P-Asserted-Service header fields in Responses

There is no usage of these header field in responses.

## 6. Examples of Usage

In this example, proxy.example.com creates a P-Asserted-Service header field from the user identity it discovered from SIP Digest authentication, and the list of services appropriate to that user, and the services that correspond to the SDP information included in the request. It forwards this information to a trusted proxy which forwards it to a trusted gateway. Note that these examples consist of partial SIP messages that illustrate only those headers relevant to the authenticated identity problem.

* F1 useragent.example.com -> proxy.example.com

```
INVITE sip:+14085551212@example.com SIP/2.0
Via: SIP/2.0/TCP useragent.example.com;branch=z9hG4bK-123
To: <sip:+14085551212@example.com>
From: "Anonymous" <sip:anonymous@anonymous.invalid>;tag=9802748
Call-ID: 245780247857024504
CSeq: 1 INVITE
Max-Forwards: 70
```

* F2 proxy.example.com -> useragent.example.com

```
SIP/2.0 407 Proxy Authorization
Via: SIP/2.0/TCP useragent.example.com;branch=z9hG4bK-123
To: <sip:+14085551212@example.com>;tag=123456
From: "Anonymous" <sip:anonymous@anonymous.invalid>;tag=9802748
Call-ID: 245780247857024504
CSeq: 1 INVITE
Proxy-Authenticate: .... realm="sip.example.com"
```

* F3 useragent.example.com -> proxy.example.com

```
INVITE sip:+14085551212@example.com SIP/2.0
Via: SIP/2.0/TCP useragent.example.com;branch=z9hG4bK-124
To: <sip:+14085551212@example.com>
From: "Anonymous" <sip:anonymous@anonymous.invalid>;tag=9802748
Call-ID: 245780247857024504
CSeq: 2 INVITE
Max-Forwards: 70
Proxy-Authorization: .... realm="sip.example.com" user="fluffy"
```

* F4 proxy.example.com -> proxy.pstn.net (trusted)

```
INVITE sip:+14085551212@proxy.pstn.net SIP/2.0
Via: SIP/2.0/TCP useragent.example.com;branch=z9hG4bK-124
Via: SIP/2.0/TCP proxy.example.com;branch=z9hG4bK-abc
To: <sip:+14085551212@example.com>
From: "Anonymous" <sip:anonymous@anonymous.invalid>;tag=9802748
Call-ID: 245780247857024504
CSeq: 2 INVITE
Max-Forwards: 69
P-Asserted-Service: "example-telephony"
```

* F5 proxy.pstn.net -> gw.pstn.net (trusted)

```
INVITE sip:+14085551212@gw.pstn.net SIP/2.0
Via: SIP/2.0/TCP useragent.example.com;branch=z9hG4bK-124
Via: SIP/2.0/TCP proxy.example.com;branch=z9hG4bK-abc
Via: SIP/2.0/TCP proxy.pstn.net;branch=z9hG4bK-a1b2
To: <sip:+14085551212@example.com>
From: "Anonymous" <sip:anonymous@anonymous.invalid>;tag=9802748
Call-ID: 245780247857024504
CSeq: 2 INVITE
Max-Forwards: 68
P-Asserted-Service: urn:xxx.exampletelephony.version1;application-v1
```

## 7. Security considerations

The mechanism provided in this document is a partial consideration of the problem of service identification in SIP. For example, these mechanisms provide no means by which end users can securely share service information end-to-end without a trusted service provider. This information is secured by transitive trust, which is only as reliable as the weakest link in the chain of trust.

## 8. IANA considerations

### 8.1. P-Asserted-Servce and P-Preferred-Service header fields

This document specifies two new SIP headers: P-Asserted-Service and P-Preferred-Service. Their syntax is given in Section 3. These headers are defined by the following information, which has been added to the header sub-registry under http://www.iana.org/assignments/sip-parameters.

| Header Name | compact | Reference |
|---|---|---|
| P-Asserted-Service | | [RFCxxxx] |
| P-Preferred-Service | | [RFCxxxx] |

Note to the RFC editor: substitute xxxx with the RFC number of this document.

### 8.2. Definition of Service-ID values

Services are identified by labels managed by IANA, according to the processes outlined in RFC 2434 [4] in a new registry called "Service-ID Labels". Thus, creating a new service requires IANA action. The policy for adding service labels is 'specification required'.

Subservice identifiers are not managed by IANA. It is the responsibility of the organisation that registered the service to manage the subservices.

Application identifiers are not managed by IANA. It is the responsibility of the organisation that registered the service to manage the applicable applications.

Entries in the registration table have the following format:

| Service | Reference | Description |
|---|---|---|
| foo | RFCxyz | Brief description of the 'foo' service |

## 9. References

[1] Braden, R., "RFC 1123: Requirements for Internet Hosts -- Application and Support", October 1989.

[2] Bradner, S., "RFC 2119: Key words for use in RFCs to Indicate Requirement Levels", March 1997.

[3] Crocker, D. and P. Overell, "RFC 2234: Augmented BNF for Syntax Specifications: ABNF", November 1997.

[4] Narten, T. and H. Alvestrand, "RFC 2434: Guidelines for Writing an IANA Considerations Section in RFCs", October 1998.

[5] Rosenberg, J., "RFC 3261: SIP: Session Initiation Protocol", June 2002.

[6] Jennings, C., Peterson, J., and M. Watson, "RFC 3323: Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", November 2002.

[7] Watson, M., "RFC 3324: Short Term Requirements for Network Asserted Identity", November 2002.

[8] Daigle, L., van Gulik, D., and P. Faltstrom, "RFC 3406: Uniform Resource Names (URN) Namespace Definition Mechanisms", October 2002.

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

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

[11] Rosenberg, J., "rosenberg-sip-app-media-tag: A Session Initiation Protocol (SIP) Media Feature Tag for MIME Application Sub-Types", May 2007.

Author's Address

Keith Drage
Alcatel-Lucent
Optimus, Windmill Hill Business Park
Swindon, Wilts
UK

Email: drage@alcatel-lucent.com

## Full Copyright Statement



## Intellectual Property



## Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).