

SIPCORE Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: December 16, 2012

C. Holmberg  
I. Sedlacek  
Ericsson  
H. Kaplan  
Acme Packet  
June 14, 2012

**Mechanism to indicate support of features and capabilities in the  
Session Initiation Protocol (SIP)  
draft-ietf-sipcore-proxy-feature-04.txt**

**Abstract**

This specification creates a new IANA registry, "Proxy-Feature Feature Caps Trees", for registering "feature caps", which are used by SIP entities not represented by the URI of the Contact header field to indicate support of features and capabilities, where media feature tags cannot be used to indicate the support.

This specification also defines a new SIP header field, Feature-Caps, to convey feature caps in SIP messages.

**Status of this Memo**

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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

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

This Internet-Draft will expire on December 16, 2012.

**Copyright Notice**

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                        |   |                    |
|------------------------|---|--------------------|
| <a href="#">1.</a>     | <a href="#">Introduction</a>                                  | <a href="#">4</a>  |
| <a href="#">2.</a>     | <a href="#">Conventions</a>                                   | <a href="#">4</a>  |
| <a href="#">3.</a>     | <a href="#">Definitions</a>                                   | <a href="#">4</a>  |
| <a href="#">4.</a>     | <a href="#">Feature Caps</a>                                  | <a href="#">4</a>  |
| <a href="#">4.1.</a>   | <a href="#">Introduction</a>                                  | <a href="#">5</a>  |
| <a href="#">4.2.</a>   | <a href="#">Registration Trees</a>                            | <a href="#">5</a>  |
| <a href="#">4.2.1.</a> | <a href="#">General</a>                                       | <a href="#">5</a>  |
| <a href="#">4.2.2.</a> | <a href="#">Global Tree</a>                                   | <a href="#">5</a>  |
| <a href="#">4.2.3.</a> | <a href="#">SIP Tree</a>                                      | <a href="#">6</a>  |
| <a href="#">4.3.</a>   | <a href="#">Registration Template</a>                         | <a href="#">6</a>  |
| <a href="#">4.4.</a>   | <a href="#">Feature Cap Specification Requirements</a>        | <a href="#">8</a>  |
| <a href="#">4.4.1.</a> | <a href="#">General</a>                                       | <a href="#">8</a>  |
| <a href="#">4.4.2.</a> | <a href="#">Overall Description</a>                           | <a href="#">8</a>  |
| <a href="#">4.4.3.</a> | <a href="#">Feature Cap Values</a>                            | <a href="#">8</a>  |
| <a href="#">4.4.4.</a> | <a href="#">Usage Restrictions</a>                            | <a href="#">9</a>  |
| <a href="#">4.4.5.</a> | <a href="#">Examples</a>                                      | <a href="#">9</a>  |
| <a href="#">5.</a>     | <a href="#">Feature-Caps Header Field</a>                     | <a href="#">9</a>  |
| <a href="#">5.1.</a>   | <a href="#">Introduction</a>                                  | <a href="#">9</a>  |
| <a href="#">5.2.</a>   | <a href="#">User Agent and Proxy Behavior</a>                 | <a href="#">10</a> |
| <a href="#">5.2.1.</a> | <a href="#">General</a>                                       | <a href="#">10</a> |
| <a href="#">5.2.2.</a> | <a href="#">B2BUA Behavior</a>                                | <a href="#">10</a> |
| <a href="#">5.2.3.</a> | <a href="#">Registrar Behavior</a>                            | <a href="#">11</a> |
| <a href="#">5.2.4.</a> | <a href="#">Proxy behavior</a>                                | <a href="#">11</a> |
| <a href="#">5.3.</a>   | <a href="#">SIP Message Type and Response Code Semantics</a>  | <a href="#">11</a> |
| <a href="#">5.3.1.</a> | <a href="#">General</a>                                       | <a href="#">11</a> |
| <a href="#">5.3.2.</a> | <a href="#">SIP Dialog</a>                                    | <a href="#">12</a> |
| <a href="#">5.3.3.</a> | <a href="#">SIP Registration (REGISTER)</a>                   | <a href="#">12</a> |
| <a href="#">5.3.4.</a> | <a href="#">SIP Stand-Alone Transactions</a>                  | <a href="#">13</a> |
| <a href="#">6.</a>     | <a href="#">Syntax</a>  | <a href="#">13</a> |
| <a href="#">6.1.</a>   | <a href="#">General</a>                                       | <a href="#">13</a> |
| <a href="#">6.2.</a>   | <a href="#">Syntax: feature cap</a>                           | <a href="#">13</a> |
| <a href="#">6.2.1.</a> | <a href="#">General</a>                                       | <a href="#">13</a> |
| <a href="#">6.2.2.</a> | <a href="#">ABNF</a>  | <a href="#">13</a> |
| <a href="#">6.3.</a>   | <a href="#">Syntax: Feature-Caps header field</a>             | <a href="#">14</a> |
| <a href="#">6.3.1.</a> | <a href="#">ABNF</a>  | <a href="#">14</a> |
| <a href="#">7.</a>     | <a href="#">IANA Considerations</a>                           | <a href="#">14</a> |
| <a href="#">7.1.</a>   | <a href="#">Registration of the Feature-Caps header field</a> | <a href="#">14</a> |
| <a href="#">7.2.</a>   | <a href="#">Proxy-Feature Feature Caps Trees</a>              | <a href="#">15</a> |



|                        |  |                    |
|------------------------|--|--------------------|
| <a href="#">7.2.1.</a> | Introduction . . . . .                         | <a href="#">15</a> |
| <a href="#">7.2.2.</a> | Global Feature Cap Registration Tree . . . . . | <a href="#">15</a> |
| <a href="#">7.2.3.</a> | SIP Feature Cap Registration Tree . . . . .    | <a href="#">15</a> |
| <a href="#">8.</a>     | Security Considerations . . . . .              | <a href="#">15</a> |
| <a href="#">9.</a>     | Acknowledgements . . . . .                     | <a href="#">16</a> |
| <a href="#">10.</a>    | Change Log . . . . .                           | <a href="#">16</a> |
| <a href="#">11.</a>    | References . . . . .                           | <a href="#">17</a> |
| <a href="#">11.1.</a>  | Normative References . . . . .                 | <a href="#">17</a> |
| <a href="#">11.2.</a>  | Informative References . . . . .               | <a href="#">17</a> |
|                        | Authors' Addresses . . . . .                   | <a href="#">18</a> |



## **1. Introduction**

The Session Initiation Protocol (SIP) [[RFC3261](#)] "Caller Preferences" extension, defined in [RFC 3840](#) [[RFC3840](#)], provides a mechanism that allows a SIP message to convey information relating to the originator's features and capabilities, using the Contact header field.

This specification creates a new IANA registry, "Proxy-Feature Feature Caps Trees", for registering "feature caps", which are used by SIP entities not represented by the URI of the Contact header field to indicate support of features and capabilities, where media feature tags cannot be used to indicate the support. Such cases are:

- o - The SIP entity acts as a SIP proxy.
- o - The SIP entity acts as a SIP registrar.
- o - The SIP entity acts as a B2BUA, where the Contact header field URI represents another SIP entity.

This specification also defines a new SIP header field, Feature-Caps, to convey feature caps in SIP messages.

NOTE: Unlike media feature tags, feature caps are intended to only be used with the SIP protocol.

## **2. Conventions**

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [BCP 14](#), [RFC 2119](#) [[RFC2119](#)].

## **3. Definitions**

Downstream SIP entity: SIP entity in the direction towards which a SIP request is sent.

Upstream SIP entity: SIP entity in the direction from which a SIP request is received.

## **4. Feature Caps**



### **4.1. Introduction**

Feature caps are used by SIP entities not represented by the URI of the Contact header field to indicate support of features and capabilities, where media feature tags cannot be used to indicate the support.

A value, or a list of values, that provides additional information about the supported feature or capability, can be associated with a feature cap.

[Section 5](#) defines how feature caps are conveyed using the Feature-Caps header field.

The feature cap ABNF is defined in [Section 6.2.2](#).

### **4.2. Registration Trees**

#### **4.2.1. General**

The following subsections define registration trees, distinguished by the use of faceted names (e.g., names of the form "tree.feature-name"). The registration trees are defined in the IANA "Proxy-Feature Feature Caps Trees" registry.

The trees defined herein are similar to the global tree and sip tree defined for media feature tags, in [RFC 2506](#) [[RFC2506](#)] and [RFC 3840](#) [[RFC3840](#)]. Other registration trees are outside the scope of this specification.

NOTE: In contrast to [RFC 2506](#) and [RFC 3840](#), this specification only defines a global tree and a sip tree, as they are the only trees defined in those RFCs that have been used for defining SIP-specific media feature tags.

When a feature cap is registered in any registration tree, no leading "+" is used in the registration.

#### **4.2.2. Global Tree**

The global feature cap tree is similar to the media feature tag global tree defined in [RFC 2506](#) [[RFC2506](#)].

A feature cap for the global tree will be registered by the IANA after review by a designated expert. That review will serve to ensure that the feature cap meets the technical requirements of this specification.





A feature cap in the global tree will be distinguished by the leading facet "g.". An organization can propose either a designation indicative of the feature, (e.g., "g.blinktags") or a faceted designation including the organization name (e.g., "g.organization.blinktags").

When a feature cap is registered in the global tree, it needs to meet the "Expert Review" policies defined in [RFC 5226](#) [[RFC5226](#)]. A designated area expert will review the proposed feature cap, and consult with members of related mailing lists.

#### [4.2.3.](#) SIP Tree

The sip feature cap tree is similar to the media feature tag sip tree defined in [RFC 3840](#) [[RFC3840](#)].

A feature cap in the sip tree will be distinguished by the leading facet "sip.".

When a feature cap is registered in the sip tree, it needs to meet the "IETF Consensus" policies defined in [RFC 5226](#) [[RFC5226](#)]. An RFC, which contains the registration of the feature cap, MUST be published.

#### [4.3.](#) Registration Template

To: sip-feature-caps@apps.ietf.org (feature caps mailing list)  
Subject: Registration of feature cap XXXX

| Instructions are preceded by '|'. Some fields are optional.

Feature cap name:

Summary of feature indicated by this feature cap:

| The summary should be no longer than 4 lines. More  
| detailed information can be provided in the SIP feature  
| cap specification.

Feature cap specification reference:

| The referenced specification MUST contain the information  
| listed in section XX of XXXX (IANA: Replace XXXX with  
| assigned RFC number of this specification.

Values appropriate for use with this feature cap:



| If no values are defined for the feature cap,  
| indicate "N/A". Details about feature cap values  
| MUST be defined in the feature cap specification.

The feature cap is intended primarily for  
use in the following applications, protocols,  
services, or negotiation mechanisms: [optional]

| For applications, also specify the number of the  
| first version which will use the feature cap,  
| if applicable.

Examples of typical use: [optional]

Considerations particular to use in individual  
applications, protocols, services, or negotiation  
mechanisms: [optional]

Interoperability considerations: [optional]

Security considerations:

Privacy concerns, related to exposure of personal  
information:

Denial of service concerns related to consequences  
of specifying incorrect values:

Other:

Additional information: [optional]

Keywords: [optional]

Related feature caps: [optional]

Name(s) & email address(es) of person(s) to  
contact for further information:

Intended usage:

| one of COMMON, LIMITED USE or OBSOLETE

Author/Change controller:

Other information: [optional]

| Any other information that the author deems



| interesting may be added here.

Figure 1: Registration Template

#### **4.4. Feature Cap Specification Requirements**

##### **4.4.1. General**

A feature cap specification **MUST** address the issues defined in the following subsections, or document why an issue is not applicable for the specific feature cap. A reference to the specification **MUST** be provided when the feature cap is registered with IANA (see [Section 4.3](#)).

It is bad practice for feature cap specifications to repeat procedures (e.g. general procedures on the usage of the Feature-Caps header field and feature caps) defined in this specification, unless needed for clarification or emphasis purpose.

A feature cap specification **MUST NOT** weaken any behavior designated with "SHOULD" or "MUST" in this specification. However, a specification **MAY** strengthen "SHOULD", "MAY", or "RECOMMENDED" requirements to "MUST" strength if features and capabilities associated with the SIP feature cap require it.

##### **4.4.2. Overall Description**

The feature cap specification **MUST** contain an overall description of the feature cap: how it is used to indicate support of a feature, a description of the feature associated with the SIP feature cap, a description of any additional information (conveyed using one or more feature cap values) that can be conveyed together with the feature cap, and a description of how the associated feature may be exercised/invoked.

##### **4.4.3. Feature Cap Values**

A feature cap can have an associated value, or a list of values.

The feature cap specification **MUST** define the syntax and semantics of any value defined for the feature cap, including possible restrictions related to the usage of a specific value. The feature cap specification **MUST** define the value(s) in accordance with the syntax defined in [section 6.2.2](#).

A feature cap value is only applicable for the feature cap for which it has been defined. For other feature caps, the value has to be



defined explicitly, even if the semantics are identical.

It is STRONGLY RECOMMENDED to not re-use a value that already has been defined for another feature cap, unless the semantics of the values are the same.

#### **4.4.4. Usage Restrictions**

If there are restrictions on how SIP entities can insert a SIP feature cap, the feature cap specification MUST document such restrictions.

There might be restrictions related to whether entities are allowed to insert a feature cap in registration related messages, standalone transaction messages, or dialog related messages, whether entities are allowed to insert a feature cap in requests or responses, whether entities also need to support other features and capabilities in order to insert a feature cap, and whether entities are allowed to indicate support of a feature in conjunction with another feature.

#### **4.4.5. Examples**

It is RECOMMENDED that the feature cap specification provide demonstrative message flow diagrams, paired with complete messages and message descriptions.

Note that example message flows are by definition informative, and do not replace normative text.

## **5. Feature-Caps Header Field**

### **5.1. Introduction**

The Feature-Caps header field is used by SIP entities to convey support of features and capabilities, by setting feature caps. Feature caps conveyed in a Feature-Caps header field indicate that the SIP entity that inserted the header field supports the associated features and capabilities.

NOTE: It is not possible to, as a Feature-Caps header field value, convey the address of the SIP entity that inserted the Feature-Caps header field. If additional data about a supported feature needs to be conveyed, such as the address of the SIP entity that indicated support of the feature, then the feature definition needs to define a way to convey that information as a value of the associated feature cap.





The feature cap specification MUST specify for which SIP methods and message types, and the associated semantics, the feature cap is applicable. See [Section 4](#) for more information. No semantics is defined for feature caps present in SIP methods and message types not covered by the associated feature cap specification.

Within a given Feature-Caps header field, feature caps are listed in a non-priority order, and for a given header field any order of listed SIP feature caps have the same meaning. For example, "foo;bar" and "bar;foo" have the same meaning (i.e. that the SIP entity that inserted the feature caps supports the features and capabilities associated with the "foo" and "bar" feature caps.

## **[5.2.](#) User Agent and Proxy Behavior**

### **[5.2.1.](#) General**

If the URI in a Contact header field of a request or response represents a SIP entity, the entity MUST NOT indicate supported features and capabilities using a Feature-Caps header field within that request or response.

When a SIP entity receives a SIP request, or response, that contains one or more Feature-Caps header fields, the feature caps in the header field inform the entity about the features and capabilities supported by the entities that inserted the header fields. Procedures how features and capabilities are invoked are outside the scope of this specification, and MUST be described by individual feature cap specifications.

When a SIP entity adds a Feature-Caps header field to a SIP message, it MUST place the header field before any existing Feature-Caps header field in the message to be forwarded, so that the added header field becomes the top-most one. Then, when another SIP entity receives a SIP request or the response, the SIP feature caps in the top-most Feature-Caps header field will represent the supported features and capabilities "closest" to the entity.

### **[5.2.2.](#) B2BUA Behavior**

The procedures in this Section applies to UAs that are part of B2BUAs that are referenced in the message by a Record-Route header field rather than by the URI of the Contact header field.

When a UA sends a SIP request, if the UA wants to indicate support of features and capabilities towards its downstream SIP entities, it inserts a Feature-Caps header field to the request, containing one or more feature caps associated with the supported features and



capabilities, before it forwards the request.

If the SIP request is triggered by another SIP request that the B2BUA has received, the UA MAY forward received Feature-Caps header fields by copying them to the outgoing SIP request, similar to a SIP proxy, before it inserts its own Feature-Caps header field to the SIP request.

When a UA receives a SIP response, if the UA wants to indicate support of features and capabilities towards its upstream SIP entities, it inserts a Feature-Caps header field to the response, containing one or more feature caps associated with the supported features and capabilities, before it forwards the response.

If the SIP response is triggered by another SIP response that the B2BUA has received, the UA MAY forward received Feature-Caps header field by copying them to the outgoing SIP response, similar to a SIP proxy, before it inserts its own Feature-Caps header field to the SIP response.

#### **5.2.3. Registrar Behavior**

If a SIP registrar wants to indicate support of features and capabilities towards its upstream SIP entities, it inserts a Feature-Caps header field, containing one or more feature caps associated with the supported features and capabilities, to a REGISTER response.

#### **5.2.4. Proxy behavior**

When a SIP proxy receives a SIP request, if the proxy wants to indicate support of features and capabilities towards its downstream SIP entities, it inserts a Feature-Caps header field to the request, containing one or more SIP feature caps associated with the supported features and capabilities, before it forwards the request.

When a proxy receives a SIP response, if the proxy wants to indicate support of features and capabilities towards its upstream SIP entities, it inserts a Feature-Caps header field to the response, containing one or more SIP feature caps associated with the supported features and capabilities, before it forwards the response.

### **5.3. SIP Message Type and Response Code Semantics**

#### **5.3.1. General**

This Section describes the general usage and semantics of the Feature-Caps header field for different SIP message types and response codes. The usage and semantics of a specific feature cap



MUST be described in the associated feature cap specification.

NOTE: Future specifications can define usage and semantics of the Feature-Caps header field for SIP methods, response codes and request types not specified in this specification.

The Feature-Caps header field ABNF is defined in [Section 6.3.1](#).

### **[5.3.2](#). SIP Dialog**

The Feature-Caps header field can be used within an initial SIP request for a dialog, within a target refresh SIP request, and within any 18x or 2xx response associated with such requests.

If a feature cap is inserted in a Feature-Caps header field of an initial request for a dialog, or within a response of such request, it indicates to the receivers of the request (or response) that the feature associated with the feature cap is supported for the duration of the dialog, until a target refresh request is sent for the dialog, or the dialog is terminated.

Unless a feature cap is inserted in a Feature-Caps header field or a target refresh request, or within a response of such request, it indicates to the receivers of the request (or response) that the feature is no long supported for the dialog.

For a given dialog a SIP entity MUST insert the same feature caps in all 18x and 2xx responses associated with a given transaction.

### **[5.3.3](#). SIP Registration (REGISTER)**

The Feature-Caps header field can be used within a SIP REGISTER request, and within the 200 (OK) response associated with such request.

If a feature cap is conveyed in a Feature-Caps header field of a REGISTER request, or within an associated response, it indicates to the receivers of the message that the feature associated with the feature cap is supported for the registration, until the registration of the contact that was explicitly conveyed in the REGISTER request expires, or until the registered contact is explicitly refreshed and the refresh REGISTER request does not contain the feature cap associated with the feature.

NOTE: While a REGISTER response can contain contacts that have been registered as part of other registration transactions, support of any indicated feature only applies to the contact(s) that were explicitly conveyed in the associated REGISTER request.



This specification does not define any semantics for usage of the Feature-Caps header field in pure registration binding fetching messages (see [Section 10.2.3 of RFC 3261](#)), where the REGISTER request does not contain a Contact header field. Unless such semantics is defined in a future extension, fetching messages will not have any impact on previously indicated support of features and capabilities, and SIP entities MUST NOT insert a Feature-Caps header field to such messages.

If SIP Outbound [[RFC5626](#)] is used, the rules above apply. However, supported features and capabilities only apply for the registration flow on which support has been explicitly indicated.

#### **[5.3.4.](#) SIP Stand-Alone Transactions**

The Feature-Caps header field can be used within a standalone SIP request, and within any 18x or 2xx response associated with such request.

If a feature cap is inserted in a Feature-Caps header field of a standalone request, or within a response of such request, it indicates to the receivers of the request (or response) that the feature associated with the feature cap is supported for the duration of the standalone transaction.

## **[6.](#) Syntax**

### **[6.1.](#) General**

This Section defines the ABNF for Feature-Caps, and for the Feature-Cap header field.

### **[6.2.](#) Syntax: feature cap**

#### **[6.2.1.](#) General**

In a feature cap name (ABNF: fcap-name), dots can be used to implement a SIP feature cap tree hierarchy (e.g. tree.feature.subfeature). The description of usage of such tree hierarchy must be described when registered.

#### **[6.2.2.](#) ABNF**

The ABNF for the feature cap:





```
feature-cap      = "+" fcap-name [EQUAL LDQUOT (fcap-value-list
                               / fcap-string-value ) RDQUOT]
fcap-name        = ftag-name
fcap-value-list  = tag-value-list
fcap-string-value = string-value
;; ftag-name, tag-value-list, string-value defined in RFC 3840
```

NOTE: In comparison with media feature tags, the "+" sign in front of the feature cap name is mandatory.

Figure 2: ABNF

### [6.3.](#) Syntax: Feature-Caps header field

#### [6.3.1.](#) ABNF

The ABNF for the Feature-Caps header fields is:

```
Feature-Caps = "Feature-Caps" HCOLON fc-value
              *(COMMA fc-value)
fc-value      = "*" *(SEMI feature-cap)
```

Figure 3: ABNF

NOTE: A "\*" value means that no information regarding which SIP entity, or domain, that indicate support of features and capabilities is provided.

## [7.](#) IANA Considerations

### [7.1.](#) Registration of the Feature-Caps header field

This specification registers a new SIP header field, Feature-Caps, according to the process of [RFC 3261](#) [[RFC3261](#)].

The following is the registration for the Feature-Caps header field:

RFC Number: RFC XXX

Header Field Name: Feature-Caps



## **7.2. Proxy-Feature Feature Caps Trees**

### **7.2.1. Introduction**

This specification creates a new sub registry to the IANA "Session Initiation Protocol (SIP) Parameters" Protocol Registry, per the guidelines in [RFC 5226](#) [[RFC5226](#)]. The name of the sub registry is "Proxy-Feature Feature Caps Trees".

### **7.2.2. Global Feature Cap Registration Tree**

This specification creates a new feature cap tree in the IANA "Proxy-Feature Feature Caps Trees" registry. The name of the tree is "Global Feature Cap Registration Tree", and its leading facet is "g.". It is used for the registration of feature caps.

The addition of entries into this tree occurs through the Expert Review policies, as defined in [RFC 5226](#). A designated area expert will review the proposed feature cap, and consult with members of related mailing lists. The information required in the registration is defined in [Section 4.3](#) of RFC XXX.

Note that all feature caps registered in the global tree will have names with a leading facet "g.". No leading "+" is used in the registrations in any of the feature cap registration trees.

### **7.2.3. SIP Feature Cap Registration Tree**

This specification creates a new feature cap tree in the IANA "Proxy-Feature Feature Caps Trees" registry. The name of the tree is "SIP Feature Cap Registration Tree", and its leading facet is "sip.". It is used for the registration of feature caps.

The addition of entries into this tree occurs through the IETF Consensus, as defined in [RFC 5226](#). This requires the publication of an RFC that contains the registration. The information required in the registration is defined in [Section 4.3](#) of RFC XXX.

Note that all feature caps registered in the SIP tree will have names with a leading facet "sip.". No leading "+" is used in the registrations in any of the feature cap registration trees.

## **8. Security Considerations**

The security issues for feature caps are similar to the ones defined in [RFC 3840](#) for media feature tags. However, as feature caps will typically not be used to convey capability information of end-user



devices, those aspects of [RFC 3840](#) do not apply to feature caps.

In addition, the [RFC 3840](#) security issue regarding an attacker using the SIP caller preferences extension [[RFC3841](#)] in order to affect routing decisions does not apply, as the mechanism is not defined to be used with feature caps.

Feature caps can provide capability and characteristics information about the SIP entity, some of which might be sensitive. The Feature-Caps header field does not convey address information about SIP entities. However, individual feature caps might provide address information as feature cap values. Therefore, mechanisms for guaranteeing confidentiality and authenticity SHOULD be provided.

## **9. Acknowledgements**

The authors wish to thank everyone in the SIP community that provided input and feedback on the work of this specification.

## **10. Change Log**

[RFC EDITOR NOTE: Please remove this Section when publishing]

Changes from [draft-ietf-sipcore-proxy-feature-03](#)

- o Additional Security Considerations text added.
- o IANA Considerations modified.
- o Editorial corrections.

Changes from [draft-ietf-sipcore-proxy-feature-02](#)

- o Changes based on WGLC comments from Shida Schubert.
- o - Document title changed
- o - Terminology alignment
- o - Note text clarifications
- o Changes based on WGLC comments from Lili Yang.

Changes from [draft-ietf-sipcore-proxy-feature-01](#)

- o Changes based on comments from Paul Kyzivat.
- o IANA Considerations text added.

Changes from [draft-holmberg-sipcore-proxy-feature-04/](#)  
[draft-ietf-sipcore-proxy-feature-00](#)

- o Media feature tags replaced with feature caps, based on SIPCORE consensus at IETF#83 (Paris).
- o Editorial corrections and modifications.

Changes from [draft-holmberg-sipcore-proxy-feature-03](#)



- o Hadriel Kaplan added as co-author.
- o Terminology change: instead of talking of proxies, talk about entities which are not represented by the URI in a Contact header field (<http://www.ietf.org/mail-archive/web/sipcore/current/msg04449.html>).
- o Clarification regarding the usage of the header field in 18x/2xx responses (<http://www.ietf.org/mail-archive/web/sipcore/current/msg04449.html>).
- o Specifying that feature support can also be indicated in target refresh requests (<http://www.ietf.org/mail-archive/web/sipcore/current/msg04454.html>).
- o Feature Cap specification registration information added.

Changes from [draft-holmberg-sipcore-proxy-feature-02](#)

- o Definition, and usage of, a new header field, instead of Path, Record-Route, Route and Service-Route.

Changes from [draft-holmberg-sipcore-proxy-feature-01](#)

- o Requirement section added
- o Use-cases and examples updated based on work in 3GPP

Changes from [draft-holmberg-sipcore-proxy-feature-00](#)

- o Additional use-cases added
- o Direction section added

## **11. References**

### **11.1. Normative References**

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", [RFC 3261](#), June 2002.

### **11.2. Informative References**

- [RFC2506] Holtman, K., Mutz, A., and T. Hardie, "Media Feature Tag Registration Procedure", [BCP 31](#), [RFC 2506](#), March 1999.
- [RFC3840] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", [RFC 3840](#), August 2004.
- [RFC3841] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Caller





Preferences for the Session Initiation Protocol (SIP)",  
[RFC 3841](#), August 2004.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 5226](#), May 2008.

[RFC5626] Jennings, C., Mahy, R., and F. Audet, "Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)", [RFC 5626](#), October 2009.

[3GPP.23.237]  
3GPP, "IP Multimedia Subsystem (IMS) Service Continuity; Stage 2", 3GPP TS 23.237 10.9.0, March 2012.

[3GPP.24.837]  
3GPP, "IP Multimedia (IM) Core Network (CN) subsystem inter-UE transfer enhancements; Stage 3", 3GPP TR 24.837 10.0.0, April 2011.

#### Authors' Addresses

Christer Holmberg  
Ericsson  
Hirsalantie 11  
Jorvas 02420  
Finland

Email: [christer.holmberg@ericsson.com](mailto:christer.holmberg@ericsson.com)

Ivo Sedlacek  
Ericsson  
Scheelewaegen 19C  
Lund 22363  
Sweden

Email: [ivo.sedlacek@ericsson.com](mailto:ivo.sedlacek@ericsson.com)



Hadriel Kaplan  
Acme Packet  
71 Third Ave.  
Burlington, MA 01803  
USA

Email: [hkaplan@acmepacket.com](mailto:hkaplan@acmepacket.com)