SIPPING
Internet-Draft
Intended status: Standards Track
Expires: September 4, 2007

D. Petrie
SIPez LLC.
S. Channabasappa, Ed.
CableLabs
March 3, 2007

A Framework for Session Initiation Protocol User Agent Profile Delivery

draft-ietf-sipping-config-framework-11

Status of this Memo

Copyright Notice

Abstract

This document defines a framework to enable configuration of Session Initiation Protocol (SIP) User Agents in SIP deployments. The framework provides a means to deliver profile data that User Agents need to be functional, automatically and with minimal (preferably none) User and Administrative intervention. The framework describes how SIP User Agents can discover sources, request profiles and receive notifications related to profile modifications. As part of

this framework, a new SIP event package is defined for notification of profile changes. The framework provides for multiple data retrieval options, without requiring or defining retrieval protocols. The framework does not include specification of the profile data within its scope.

Table of Contents

1. Introduction

SIP User Agents require configuration data to function properly. Examples include network, device and user specific information. Ideally, this configuration process should be automatic and require minimal or no user intervention.

Many deployments of SIP User Agents require dynamic configuration and cannot rely on pre-configuration. This framework provides a standard means of providing dynamic configuration which simplifies deployments containing SIP User Agents from multiple vendors.

This framework also addresses modifications to profiles and the corresponding change notifications to the SIP User Agents using a new event package. However, the framework does not define the content or format of the actual profile data, leaving that to future standardization activities.

2. Terminology

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 RFC 2119 [RFC2119].

This document also reuses the SIP terminology defined in [RFC3261] and [RFC3265], and specifies the usage of the following terms.

Device: software or hardware entity containing one or more SIP user agents. It may also contain entities such as a DHCP client.

Device Provider: the entity responsible for managing a given device

Local Network Provider: the entity that controls the local network to which a given device is connected

SIP Service Provider: the entity providing SIP services to users. This can refer to private enterprises or public entities.

Profile: configuration data set specific to an entity (for example, user, device, local network or other).

Profile Type: a particular category of Profile data (for example, User, Device, Local Network or other).

Profile Delivery Server (PDS): the source of a Profile, it is the logical collection of the Profile Notification Component (PNC) and the Profile Content Component(PCC).

Profile Notification Component (PNC): the logical component of a Profile Delivery Server that is responsible for enrolling devices and providing profile notifications.

Profile Content Component (PCC): the logical component of a Profile Delivery Server that is responsible for storing, providing access to, and accepting profile content.

3. Overview

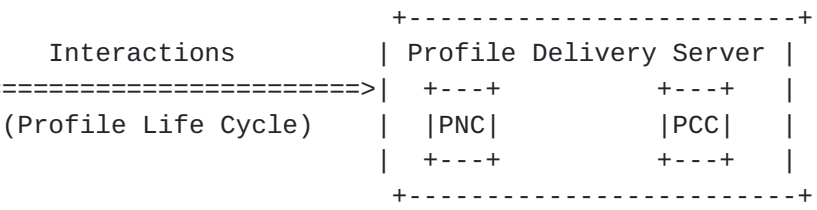
This section provides an overview of the configuration framework. It introduces the reference model and explains key concepts such as the Profile Life Cycle and the Profile Types. It is meant to serve as a reference section for the document, rather than providing a specific logical flow of material, as it may be necessary to revisit these sections for a complete understanding of this document. The detailed framework for the profile delivery, presented in Section 5, is based on the concepts introduced in this section.

3.1. Reference Model

The design of the framework was the result of a careful analysis to identify the configuration needs of a wide range of SIP deployments. As such, the reference model provides for a great deal of flexibility, while breaking down the interactions to their basic forms which can be reused in many different scenarios.

In its simplest form, the reference model for the framework defines the interactions between the Profile Delivery Server(PDS) and the device. The device needs the profile data to effectively function in the network. The PDS is responsible for responding to device requests and providing the profile data. The set of interactions between these entities is referred to as the Profile Life Cycle.

This reference model is illustrated in the diagram below.



PNC = Profile Notification Component
PCC = Profile Content Component

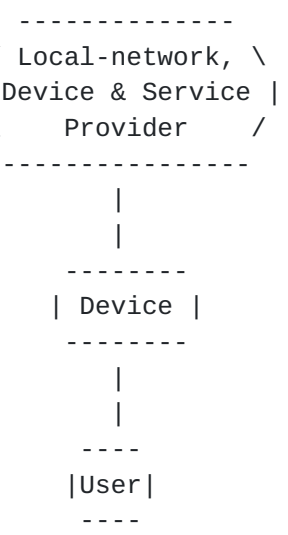
Framework Reference Model

The PDS is subdivided into two logical components:

- Profile Notification Component (PNC), responsible for enrolling devices in Profile event subscriptions and providing Profile change notifications;
- Profile Content Component (PCC), responsible for storing, providing access to, and accepting modifications related to profile content.

SIP deployments vary considerably. For the sake of simplicity, two deployment scenarios representing either end of the SIP deployment spectrum are presented.

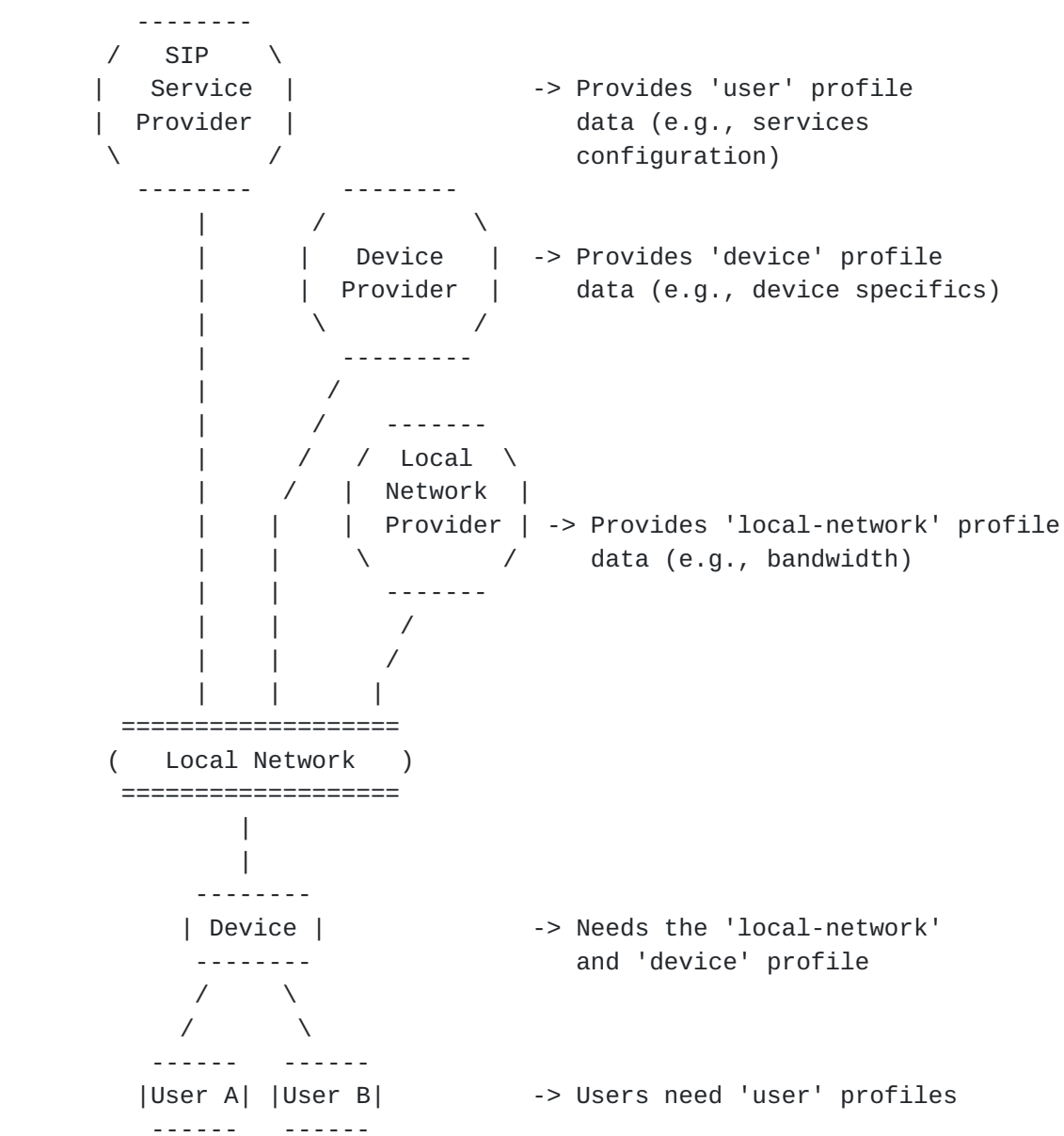
In the simplest scenario, a device connects through a network that is controlled by a single provider who provides the local-network, manages the devices, and offers services to the users. The Provider propogates profile data to the device that contains all the necessary information to obtain services in the network (including information related to the local-network and the users). This is illustrated in the following diagram.



Simple System Level Model

There are also deployments where the device can connect via a local network that is not controlled by the SIP Service Provider, for example, devices that connect via available public WiFi hotspots. In such cases, Local Network Providers may wish to provide local network information such as bandwidth constraints to the devices.

Devices may also be controlled by Device Providers that are independent of the SIP Service Provider who provides user services, for example, kiosks that allow users to access services anywhere. In such cases the profile data may have to be obtained from different profile sources: local network provider, device provider and SIP service provider. This is indicated in the following diagram.



General System Level Model

As illustrated, the simplest deployments present a single profile source whereas others may present multiple profile sources. To be effective, a configuration framework needs to address various deployment scenarios. To address a vast majority of deployments this framework specifies three distinct profiles, each of which can be obtained from a different provider, and a profile life cycle common to any profile type.

The understanding is that deployments in general will support the defined profile types. However, the framework allows for flexibility in specialized cases. The devices are required to support all the three profile types, unless configured otherwise (at a minimum they need to support the device profile). The deployments are required to support the device profile, and user profiles for known users. In the presence of multiple profiles, a retrieval order is specified for the devices. Additional profiles may also be specified outside the scope of this document, but are expected to follow the same profile life cycle.

3.2. Data Model and Profile Types

This framework specifies the following three profiles. Additional extended profiles may also be defined.

Local Network Profile: contains configuration data related to the local network to which a device is directly connected. It is expected to be provided by the Local Network Provider.

Device Profile: cContains configuration data related to a specific device, provided by the Device Provider.

User Profile: contains configuration data related to a specific User, as required to reflect that user's preferences and the particular services subscribed to. It is expected to be provided by the SIP Service Provider providing services.

3.3. Profile Life Cycle

Automated profile delivery requires proactive behavior on the part of a device. It also requires one or more PDSs which provide the profile data. The set of communications that results in profile delivery is characterized by the profile life cycle. Each profile is propogated using the profile life cycle.

The life cycle is initiated when the device enrolls for profile data. Enrollment either results in profile data or in information referencing content indirection. In the case of content indirection, the provided retrieval procedures are used to retrieve the profile. Additionally, the profile life cycle allows for profile change operations by authorized entities. If a profile change operation is successful, it results in profile change notifications to all

enrolled devices.

The specific functional steps are as follows:

Profile Enrollment: the process by which a device requests, and if successful, enrolls with a PDS capable of providing a profile. A successful enrollment is indicated by a notification containing the profile information (contents or content indirection information). Depending on the request, this could also result in a subscription to notification of profile changes.

Profile Content Retrieval: the process by which a device retrieves profile contents, if the profile enrollment resulted in content indirection information.

Profile Change Notification: the process by which a device is notified of any changes to an enrolled profile. This may provide the device with modified profile data or content indirection information.

Profile Change Operation: The process by which an authorized entity - such as a configuration management server or a device - pushes a profile change to the PDS.

4. Use Cases

This section provides a small - non-comprehensive - set of representative use cases to further illustrate how this Framework can be utilized in SIP deployments. The first use case is simplistic in nature, where as the second is relatively complex. The use cases illustrate the effectiveness of the framework in either scenario.

For Security Considerations please refer to Section 9.

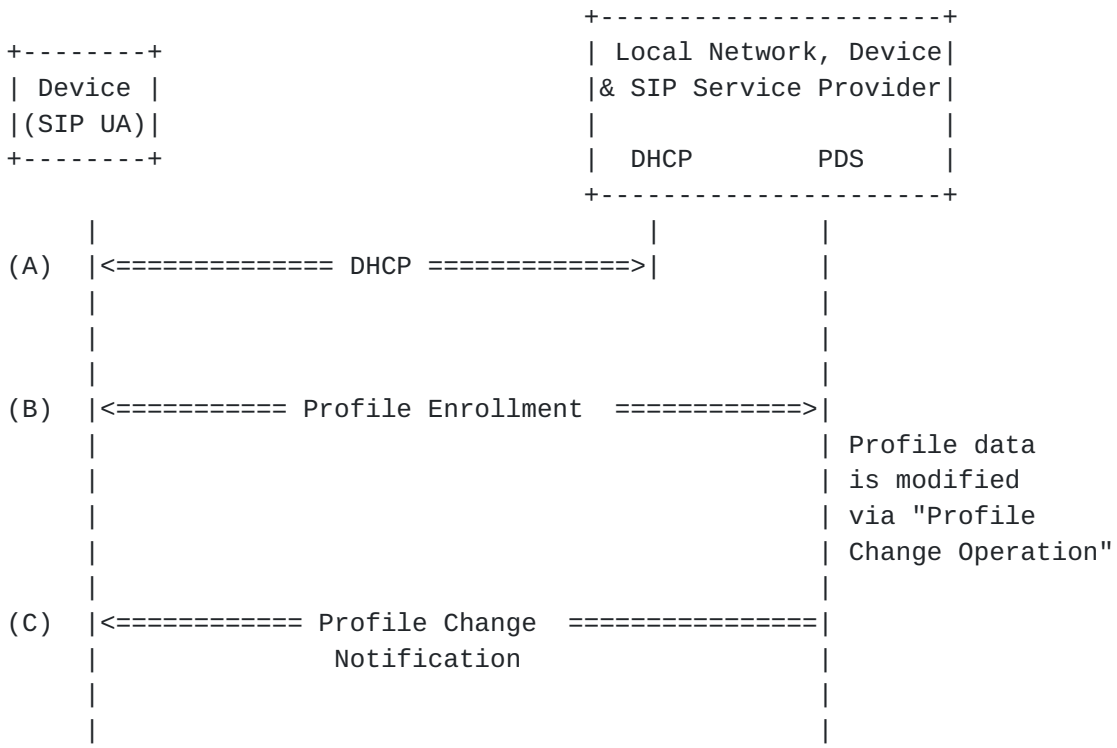
4.1. Simple Deployment Scenario

Description: Consider a deployment scenario (for example, a small private enterprise) where a single entity enables the local network, manages deployed devices and provides SIP services. The devices never connect outside the local network and are each pre-configured with a single user.

The following assumptions apply:

- The device profile data contains all the information necessary for the device to participate in the local network and obtain services
- The device is pre-configured to only request the device profile
- The enrollment notification contains the profile data (profile content retrieval is not required)

The following diagram illustrates this use case and highlights the communications relevant to the framework specified in this document.



The following is an explanation of the interactions in the diagram.

(A) Upon initialization, the device obtains IP configuration parameters using DHCP

(B) The device performs Profile Enrollment for the device profile; the device profile data is contained in the enrollment notification
(C) Due to a modification of the device profile, a Profile Change Notification is sent across to the device, along with the modified profile

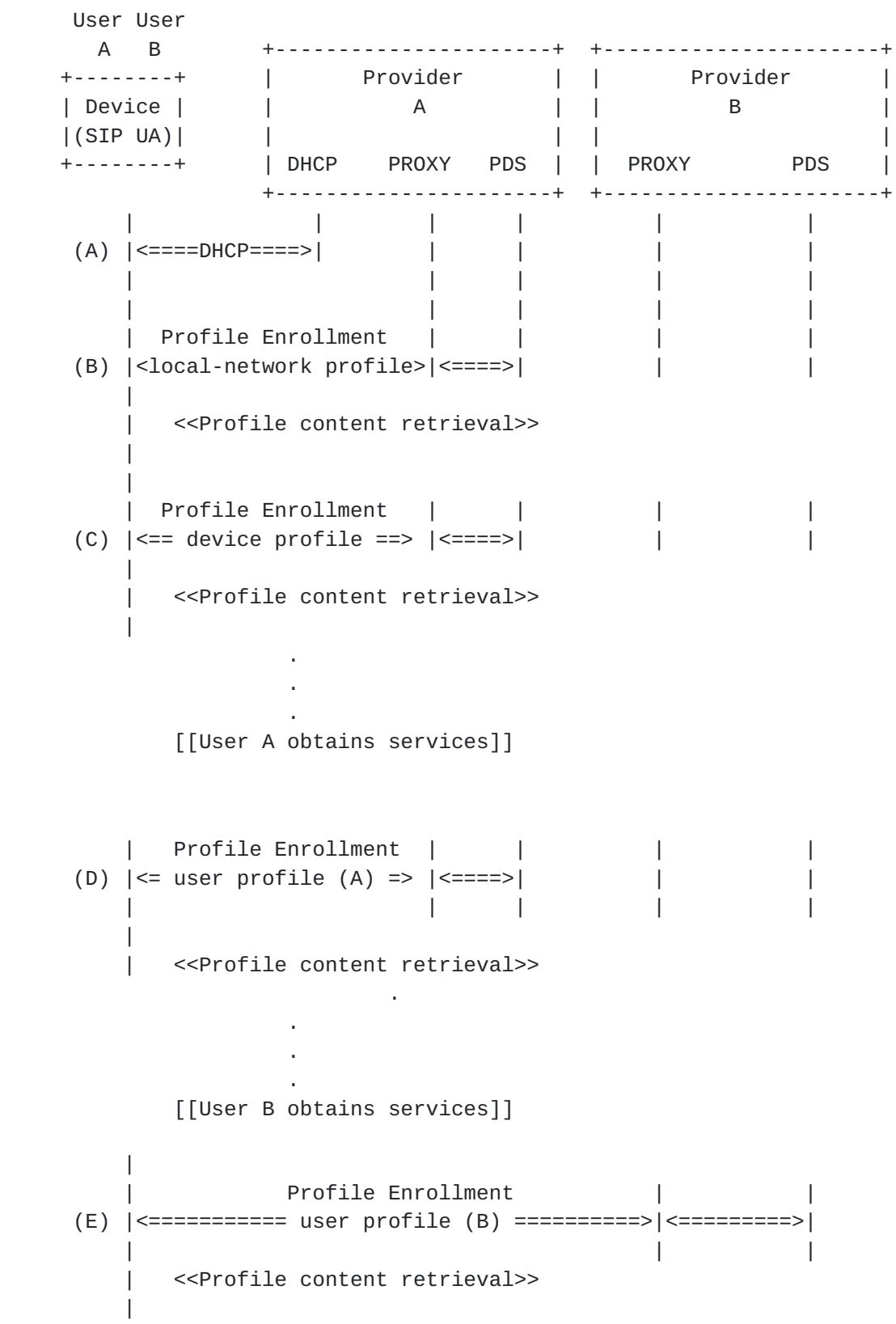
4.2. Devices supporting multiple users from different Service Providers

Description: Consider a single device (for example, Kiosk at an airport) that allows for multiple users to obtain services from a list of pre-configured SIP Service Providers.

The following assumptions apply:

- Provider A is the Device and Local Network Provider for the device, and the SIP Service Provider for user A; Provider B is the SIP Service Provider for user B
- Profile enrollment always results in content indirection information requiring profile content retrieval

The following diagram illustrates the use case and highlights the communications relevant to the framework specified in this document.


User User
A B
Device
(SIP UA)
Provider A
DHCP PROXY PDS
Provider B
PROXY PDS
(A)
<====DHCP====>
Profile Enrollment
(B)
<local-network profile>
<====>
<<Profile content retrieval>>
Profile Enrollment
(C)
<== device profile ==>
<====>
<<Profile content retrieval>>
[[User A obtains services]]
Profile Enrollment
(D)
<= user profile (A) =>
<====>
<<Profile content retrieval>>
[[User B obtains services]]
Profile Enrollment
(E)
<========== user profile (B) ==========>
<=========>
<<Profile content retrieval>>

The following is an explanation of the interactions in the diagram.

(A) Upon initialization, the device obtains IP configuration parameters using DHCP. This also provides the local domain information to help with local-network profile enrollment

(B) The device requests profile enrollment for the local network profile. It receives an enrollment notification containing content indirection information from Provider A's PDS. The device retrieves the profile (this contains useful information such as firewall port restrictions and available bandwidth)

(C) The device then requests profile enrollment for the device profile. It receives an enrollment notification resulting in device profile content retrieval. The device initializes the User interface for services.

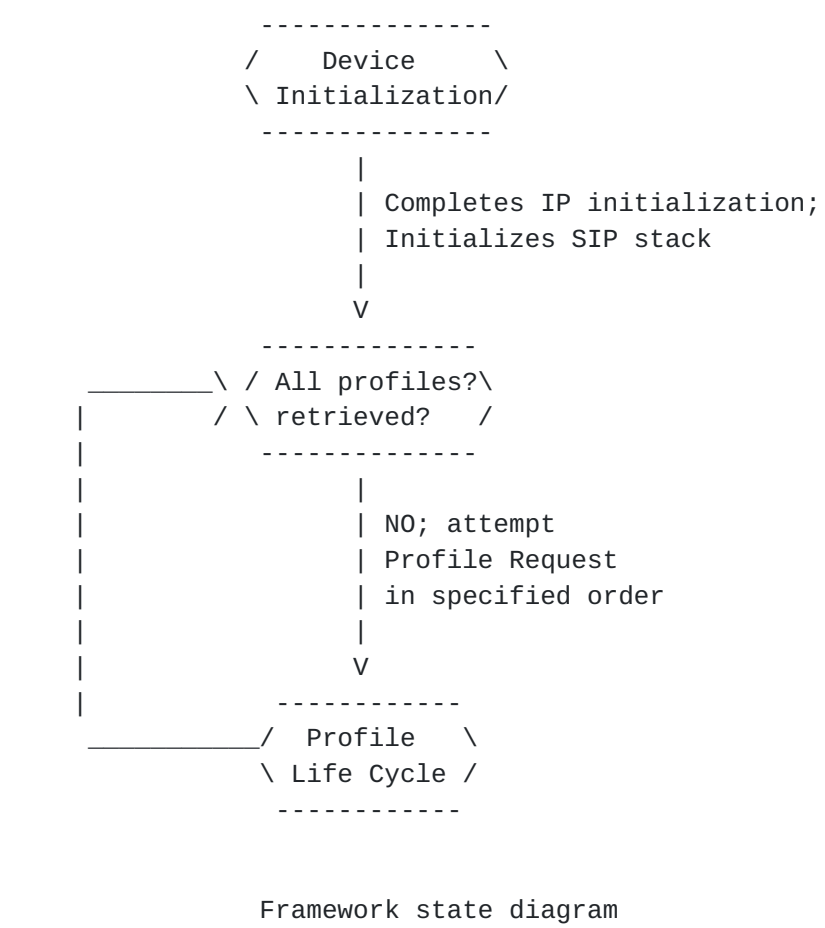
(D) User A with a pre-existing subscription with Provider A attempts communication via the user Interface. The device uses the user supplied information (including any credential information) and requests profile enrollment for user A's profile. Successful enrollment and profile content retrieval results in services for user A.

(E) At a different point in time, user B with a pre-existing subscription with Provider B attempts communication via the user Interface. It enrolls and retreives user B's profile and this results in services for user B.

5. Profile Delivery Framework

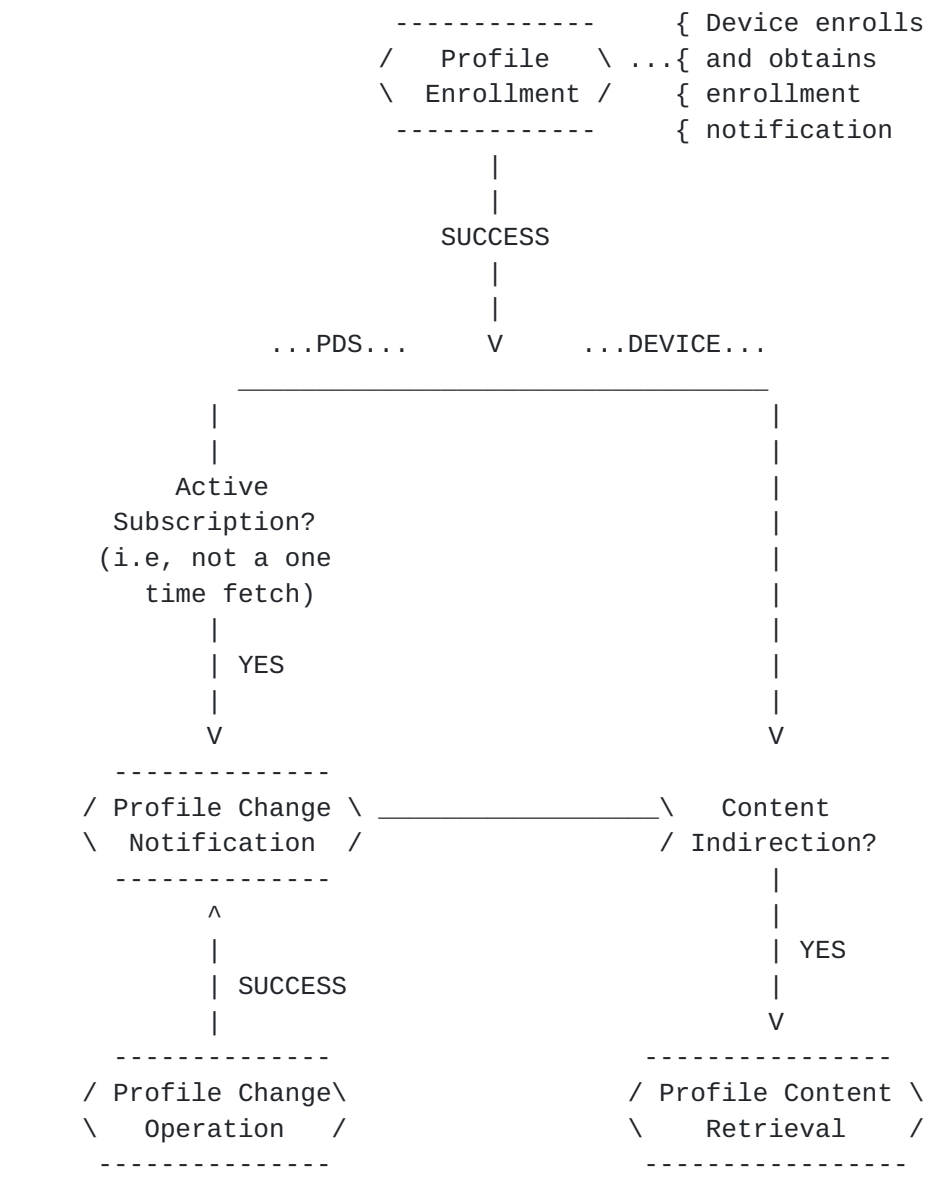
This section details the framework requirements. The Profile Life Cycle (introduced in Section 3), is examined in further detail, with requirements that apply to the device and the PDS. Unless explicitly enhanced or indicated by an implementing specification, the device and the PDS MUST follow the Profile Life Cycle requirements stated in this section for all supported profile types.

A high-level representation of the framework is shown in the following state diagram. Each of the specified profile types is retrieved individually, in the specified order (see below), until all needed Profiles have been received.



Framework state diagram

The Profile Life Cycle, for each profile, is illustrated in the diagram below.



The Profile Life Cycle is initiated when the device transmits an enrollment request for a specific profile. If this is accepted, it results in an enrollment notification that contains the profile data or profile content indirection information. Unless the enrollment request indicates a one-time profile request, it also results in enrollment for profile change notifications. If the profile is modified at any point in time, the profile change notification is transmitted to the device. Notifications due to profile enrollment or change operation may result in content indirection in which case

the device uses profile content retrieval to obtain the profile data.

The Profile Life Cycle is the same for all the profile types, but there are different requirements in each step based on the profile types. This framework defines three profile types and an order that MUST be followed by the device in requesting them (when it retrieves two or more of the defined profile types), as follows:

- local-network
- device
- user

The sub-sections that follow specify the Profile Life Cycle details, with specific requirements based on each profile type.

5.1. Profile Enrollment

The first step to obtaining a profile is PDS Enrollment. This is initiated by the device and involves:

- creating a profile enrollment subscription
- transmitting a profile enrollment request
- receiving a profile enrollment notification

The processes are interlinked and retries encompass all three phases. For example, if the enrollment request does not result in a profile enrollment notification, the device is required to retry alternate profile enrollment subscription creation options. Only when all the enrollment subscription creation options are exhausted does the device assume that the profile enrollment has failed. The processes themselves are illustrated in the following sub-sections.

5.1.1. Creation of Enrollment Subscription

Each profile type requires its own subscription and based on the entity requesting it, presents certain unique requirements (for example, the device identifier is provided for the device profile type where as the user identifier is provided for the user profile type). Further, the profile types are aimed at different PDSs and hence are identified differently (for example, the local-network is identified by the local domain name where as the Service Provider is identified based on the Service Provider's domain name). Some of this information can be obtained in multiple ways (such as local domain information that can be configured statically or dynamically) and the device may have to try different information sources to

obtain the required information (for example, dynamic configuration can override statically configured information). Based on these considerations, the framework defines different rules for obtaining and presenting the information for each profile type. Additionally, when more than one information source is possible for the information, it is presented as well. This is highlighted in the following sub-sections.

5.1.1.1. SIP SUBSCRIBE for the Local-Network profile type

Before attempting to create a SIP SUBSCRIBE requesting the local-network profile, the device MUST have established local network connectivity. It MUST also have knowledge of the local network domain either via static configuration or dynamic discovery via DHCPv4 ([RFC2131]) or DHCPv6 ([RFC3315]). The following requirements apply:

- the user part of the Request URI MUST NOT be provided. The host and port part of the Request URI MUST be set to the concatenation of "sipuaconfig" and the local network domain
- a user AOR, if known to the device MUST be used to populate the "From" field, unless privacy requirements prohibit its use (this is useful if the user has privileges in the local network beyond those of the default user)
- if a user AOR is not known, the user portion of the "From" field MUST be set to "anonymous"; the host and port portion of the Request URI MUST be set to the concatenation of "sipuaconfig" and the local network domain
- the "device-id" event header parameter MUST be set to the device identifier that the device will use to request the device profile

For example: If the device requested and received the local domain name via DHCP to be: airport.example.net, then the Local-Network Profile SUBSCRIBE Request URI would look like:

sip:sipuaconfig.airport.example.net

The Event header would look like the following if the device decided to provide MAC%3a00DF1E004CD0@airport.example.net as the device identifier. (Alice may have a prior arrangement with the local network operator giving her special privileges.)

```
Event: ua-profile;profile-type=local-network;
       device-id="sip:MAC%3a00DF1E004CD0@airport.example.net"
```

The local-network profile SUBSCRIBE Request URI does not have a user part so that the URI is distinct between the "local" and "device"

URIs when the domain is the same for the two. This provides a means of routing to the appropriate PDS in domains where they are distinct servers.

The From field is populated with the user AOR, if available. This allows the local network provider to propagate user-specific profile data, if available. The "device-id" event header parameter is set to the device identifier. Even though every device may get the same (or similar) Local-Network Profile, the uniqueness of the "device-id" event header provides an important capability. Having unique From fields allows the management of the local network to track devices present in the network and consequently also manage resources such as bandwidth and port allocation.

5.1.1.2. SIP SUBSCRIBE for the Device Profile Type

The device profile type allows the Service Provider managing a device to provide device-specific configuration information. To enable this, the Request URI needs to identify the device and the PDS domain within which it is recognizable. Accordingly, this Framework presents the following requirements for the formation of a Subscription Request URI to request the "device" profile type

- the user portion of the Request URI MUST be set to a unique device Identifier
- the host and port portion of the Request URI MUST be set to the PDS domain

The following sub-sections explain identification of - and the requirements related to - the device Identifier and the PDS domain discovery.

5.1.1.2.1. Device Identifier

The device profile could be specific to each device in a SIP deployment (for example, vendor/model) or shared across device types (for example, based on services and service tiers). Further, the same device might be provided different configuration profiles based on deployment models. Device Identifiers play a significant role in ensuring delivery of the correct profile and hence need to be unique within a PDS domain to support the various deployment models.

This Framework requires that device Identifiers MUST be unique and persistent over the lifetime of a device. Device Identifier representations auto-generated by devices SHOULD be based on MAC address or UUID ([RFC4122]) based representations. A device may use alternate device identifiers (for example, SIP URIs) obtained via

pre-configuration or dynamic configuration (for example, device profile).

If a MAC address is used, the following requirements apply:

- the device identifier MUST be formatted as the characters "MAC:" followed by a twelve digit hexadecimal upper case representation of the MAC address to form a proper URN ([RFC2141]). The MAC address representation MUST NOT include visual separators such as colons and whitespaces. The representation is denoted using the following ABNF syntax

```
mac-ident = MAC ":" 12UHEX
MAC       = %x4d.41.43      ; MAC in caps
UHEX      = DIGIT / %x41-46 ; uppercase A-F
```

- the MAC address MUST only be used to represent a single device. It MUST NOT be used if more than one device can potentially use the same MAC Address (for example, multiple software entities on a single platform). In such cases, the UUID representation SHOULD be used

If a UUID is used, the following requirements MUST apply:

- the same approach to defining a user agent Instance ID as [RFC4122] MUST be used
- when the URN is used as the user part of the URI, it MUST be URL escaped

  The colon (":") is not a legal character (without being escaped) in the user part of an addr-spec ([RFC4122]). For example the instance ID: urn:uuid:f81d4fae-7ced-11d0-a765-00a0c91e6bf6@example.com would be escaped to look as follows in a URI: sip:urn%3auuid%3af81d4fae-7ced-11d0-a765-00a0c91e6bf6@example.com

  The ABNF for the UUID representation is provided in [RFC4122]

5.1.1.2.2. PDS Domain Discovery

A device needs to identify the PDS domain to form the host and port part of the Request URI. Ideally, this information should be obtained via a single method. However, support for various deployment models implies multiple device environments (for example, residential routers, enterprise LANs, WLAN hotspots and dialup modem) and presents hurdles to specifying a single method (for example, if a device is always in the SIP Service Provider's network one could use DHCP). To accommodate multiple deployment scenarios, the framework specified in this document presents multiple approaches.

Devices MUST follow the procedures specified below in the order presented, unless exceptions are made by device manufacturers or Device Providers who may provide an option for the user to choose the order (to suit specific deployment models, for example).

1. Service Provider pre-configuration

The device MAY be pre-configured with information that can be utilized to identify the host and port of the Request URI. The information can be provided - as examples - when the device is manufactured, by using Service Provider entities (flash card, SIM card) or via a Service Provider specific method (for example, information or methods that lead to self subscription). If the device is specified to utilize this approach, it MUST attempt to do so before trying other methods. The details of how this is accomplished are beyond the scope of this document.

2. IP Configuration

If pre-configuration is not an option, or not available, IP configuration MUST be utilized to try and obtain information that can help with identification of the host and port for the Request URI. The framework defines the following methods within this procedure to accomplish this. device MUST follow the methods defined, in the order specified, i.e. if the first option cannot be accomplished or results in a failure, then next method is tried. Failure of a specific method is indicated when the device cannot successfully complete Profile Enrollment.

2a. DHCP option for SIP server:

Devices that support DHCP MUST attempt to obtain the host and port of the outbound proxy during the DHCP process, using the DHCP option for SIP servers defined in [RFC3361] or [RFC3319] (for IPv4 and IPv6 respectively), and use these as the host and port part of the request URI.

For example, a MAC based device identifier with a DHCP SIP servers option indicating example.com, the Request URI would be constructed as sip:MAC%3aABC123EFD456@example.com

2b. Local IP Network Domain:

- devices that support DHCP MUST attempt to obtain the local IP network domain during the DHCP process, using DHCP option 15 and use these as the host and port part of the request URI using the technique specificed in [RFC3263]

+ For example, a MAC based devices identifier with a DHCP option 15 indicating local.example.com, the Request URI would be constructed as sip:MAC%3aABC123EFD456@local.example.com

- If the local IP network domain is available (previous method), but the usage of the local IP Network domain results in a failure, the device MUST use the local IP network domain, prefixing it using the label "sipuaconfig."

+ For example, a MAC based device Identifier with a DHCP option 15 indicating local.example.com, the Request URI would be constructed as sip:MAC%3aABC123EFD456@sipuaconfig.local.example.com

3. Manual

If pre-configuration and IP Configuration are not options or result in failures, the device SHOULD provide a means for the user to present information that may help with the retrieval process. Exceptions to this requirement MAY include devices with no user interface appropriate for such entry.

This framework provides the following alternatives which can be considered individually or together, in any order.

Device Provider PDS information: The user SHOULD be allowed to present the host and port information which can help with the creation of the Subscription URI to locate a PDS capable of providing the profile.

Device Provider Configuration Server information The user MAY be allowed to present information pertaining to a configuration server that provides the device profile, not using a PDS as defined in this framework. This framework specifies one such possible process in Section 5.5.1.

5.1.1.3. SIP SUBSCRIBE for the User Profile Type

The user profile allows the responsible SIP Service Provider to provide user-specific configuration. This is based on the user's identity that is usually known in the network (for example, associated with a subscription). Similar to the profiles provided to devices, the content and propagation of user Profiles may partake differently, based on deployment scenarios (for example, users belonging to the same subscription might - or might not - be provided the same profile). However, each user is uniquely identified in a

SIP Service Provider's network using an Address Of Record (AOR). Devices implementing this framework MUST use the user's AOR to populate the Request URI.

A device MAY obtain the user's AOR using various methods such as pre-configuration, via the device profile or dynamically via a user Interface.

5.1.1.4. Caching of SIP Subscription URIs

Creation of Subscription URIs is vital for successful Profile Enrollment. Unlike the user Profile - Local-Network and device profiles are expected to be requested based on discovered information (for example, domain name discovered via DHCP). These profile types have different goals and hence, caching of the Subscription URI should be carefully considered.

The Local-Network profile type is aimed at obtaining information from the local network. The local network can change across device initializations (for example, user moves the device from a home network to a workplace LAN). Thus, the device SHOULD NOT remember local-network profile subscription URIs across initializations. The device SHOULD re-create the Subscription URI every time it moves to a new network or gets re-initialized. Exceptions may be cases where the device can unambiguously determine changes to the local network.

The device profile type is aimed at obtaining information from the SIP Service Provider managing the device. Once established, the Service Provider does not change often (an example of an exception would be the re-use of devices across Service Providers). However, if the discovery process is used, the device can only be sure of having reached the Service Provider upon successful Profile Enrollment and Profile Notification. Thus, the device SHOULD cache the Subscription URI for the device profile. When cached, the device should use the cached Subscription URI upon a reset. Exceptions include cases where the device identifier has changed (for example, new network card with a new MAC address), Service Provider information has changed (for example, user initiates change) or the device cannot obtain its profile using the Subscription URI.

> Devices SHOULD NOT cache the Subscription URI for the device profile type until successful Profile Notification. The reason for this is that a PDS may send 202 responses to SUBSCRIBE requests and NOTIFY responses to unknown devices (see Section 6.6) with no profile data or URIs. Thus, successful Profile Notification is the only sure way to know that the Subscription URI is valid.

5.1.2. Profile Enrollment Request Transmission

A device requesting a profile type specified in this document - and is successful in forming a Subscription URI - MUST enroll using the event package defined, and as specified, in this framework (see Section 6) . The following requirements apply:

- the device MUST cater to the Event Package requirements specified in Section 6.2 (for example, indicate the profile type being requested in the profile-type parameter)
- the device MUST use the Subscription URI pertaining to the profile type being requested, as specified in Section 5.1

The SIP infrastructure receiving such requests is expected to relay and process profile enrollment requests. When a Profile Enrollment request is received by a PDS, it SHOULD accept and respond to any profile requests. Exceptions are when Service Provider policy prevents such a response (for example, requesting entity is unknown).

Successful Profile Enrollment involves the following

- Acceptance of the SUBSCRIBE request by a PDS (indicated via a 200 response)
- Receipt of an initial Profile Notification within the timeouts as specified in [RFC3265]

A device SHOULD follow suitable BackOff and Retry mechanisms if a successful Profile Enrollment does not happen within the expected period.

5.1.3. Profile Enrollment Notification

Successful Profile Enrollment is indicated by an enrollment notification. This provides either a) the profile contents b) content indirection information. If content indirection information is provided, the device retrieves the profile using Profile Content Retrieval. If the profile contents are provided, the following requirements hold good:

- the device MUST make the new profiles effective within the specified timeframe, as described in Section 6.2
- the device SHOULD cache (i.e. store persistently) the contents of retrieved profiles, until overridden by subsequent Profile Change Notifications (this avoids situations where a PDS is unavailable, leaving the device without required configuration)

Failure to receive the initial NOTIFY following a successful enrollment MUST be treated the same as a failed enrollment. In such

a scenario, the device MUST retry using alternate methods for creation of the enrollment subscription and transmit an enrollment request. If all the enrollment subscription creation have been exhausted, the device MUST treat it as a failure to obtain the profile and take appropriate measures.

For NOTIFY content please refer to Section 6.5.

5.2. Profile Content Retrieval

Upon successful Profile Enrollment, the device can retrieve the documents pertaining to the requested profile directly or via the URI(s) provided in the NOTIFY request as specified in Section 6.5. Profile Content Retrieval protocols and frameworks are out of scope for this specification.

5.3. Profile Change Operation

Configuration Profiles can change over time. Modifications can be initiated by various entities (for example, via the device, back-office components and end-user web interfaces for configuration servers) and for various reasons (such as, change in user preferences, modifications to services, enterprise-imposed common features or restrictions). This framework allows for such changes to be communicated to the PDS, using the term Profile Change Operation.

Any changes to a Profile as a result of Profile Change Operation MUST result in a Profile Notification to all enrolled devices for that Profile, if any.

Definition of specific mechanisms for Profile Change Operation are out of scope of this document.

5.4. Profile Change Notification

Whenever a profile is changed, a PDS compliant with this framework MUST NOTIFY all the devices currently subscribed to the profile under consideration. This process is termed Profile Change Notification.

For NOTIFY content please refer to Section 6.5.

5.5. Additional Considerations

This section provides a special case for retrieval of the device profile and highlights considerations and requirements on external entities such as Profile Data Frameworks.

5.5.1. Manual retrieval of the Device Profile

At a minimum, a device requires the device profile to be able to function effectively. However, the methods specified in this document may fail to provide a device with a profile. To illustrate with an example, consider the case of a device that finds itself behind a local network which does not provide information about DNS servers in the network (for example, misconfigured home network). In such cases, it would be beneficial to employ an alternative means to obtain the profile information (for example, resolvable DNS Servers could be part of the device profile). While this specification recommends that such a method be made available, it also specifies one such option using HTTP that is described in this sub-section. devices expected to encounter scenarios where propogation of the device profile can be hindered may employ the specified - or any alternative - process.

The method being described involves the device to utilize a HTTPS URI (and any required credentials) based on either pre-configuration or manual entry by the user (in cases where such an interface is possible). This can lead to the retrieval of the device profile which may contain the properties for the SUBSCRIBE Request URI and credentials for Profile Enrollment and Profile Notification. This approach bootstraps the process in a different step in the cycle, but uses the same framework.

Further, this document defines a new HTTP request header "Event". The syntax of the HTTP Event header is the same as the SIP Event header defined in this document. Similar to the SIP Event header the purpose of the HTTP Event header is to define the content of the state information to be retrieved. In particular, the state information is the device, user or local-network profile for the device. The SIP Event header parameters for this event package ("profile-type", "vendor", "model", "version") are also mandatory for the HTTP Event header as they are used to provide information as to what profile type is requested along with information about the device which may impact the contents of the profile. When the device starts with retrieval of the profile via HTTPS (instead of a SIP SUBSCRIBE to the event package), the device MUST provide the Event header defined.

5.5.2. Device Types

The examples in this framework tend to associate devices with entities that are accessible to end-users. However, this is not necessarily the only type of device that can utilize the specified Framework. devices can be entities such as user Interfaces (that

allow for device Configuration), entities in the network that do not directly communicate with any users (for example, Service Provider deployed gateways) or elements in the Service Provider's network (for example, SIP servers).

5.5.3. Profile Data

This framework does not specify the contents for any profile type. Follow-on standardization activities can address profile contents. However, it makes the following assumptions and recommendations:

- When the device receives multiple profiles, the contents of each profile type will only contain data relevant to the entity it represents. As an example, consider a device that obtains all the defined profiles. Information pertaining to the local network is contained in the 'local-network' profile and not the'user' profile. This does not preclude relevant data about a different entity from being included in a profile type, for example, the 'device' profile type may contain information about the users allowed to access services via the device. A profile may also contain starting information to obtain subsequent Profiles
- Data overlap SHOULD be avoided across profile types, unless necessary. If data overlap is present, prioritization of the data is left to data definitions. As an example, the device profile may contain the list of codecs to be used by the device and the user Profile (for a user on the device) may contain the codecs preferred by the user. Thus, the same data (usable codecs) is present in two profiles. However, the data definitions may indicate that to function effectively, any codec chosen for communication needs to be present in both the profiles.

5.5.4. Profile Data Frameworks

This framework specified in this document does not address profile data representation, storage or retrieval protocols. It assumes that the PDS has a PCC based on existing or other Profile Data Frameworks, for example, XCAP ([I-D.ietf-simple-xcap]).

While it does not impose vast constraints on any such framework, it does allow for the propagation of profile content to PDS (specifically the PCC). Thus, Profile Data or Retrieval frameworks used in conjunction with this framework MAY consider techniques for propagating incremental, atomic changes to the PDS. For example, a means for propagating changes to a PDS is defined in XCAP ([I-D.ietf-simple-xcap]).

5.5.5. Additional Profile Types

This document specifies three profile types: local-network, device and user. However, there may be use cases for additional profile types. For example, profile types for application specific profile data. Definition of such additional profile types is not prohibited, but considered out of scope for this document.

5.5.6. Deployment considerations

The framework defined in this document was designed to address various deployment considerations, some of which are highlighted below.

Provider relationships:

- The local network provider and the SIP service provider can often be different entities, with no administrative or business relationship with each other;
- There may be multiple SIP service providers involved, one for each service that a user subscribes to (telephony service, instant messaging, etc.); this Framework does not specify explicit behavior in such a scenario, but it does not prohibit its usage either
- Each user accessing services via the same device may subscribe to different sets of services, from different Service Providers;

User-device relationship:

- The relationship between devices and users can be many-to-many (for example, a particular device may allow for many users to obtain subscription services through it, and individual users may have access to multiple devices);
- Each user may have different preferences for use of services, and presentation of those services in the device user interface;
- Each user may have different personal information applicable to use of the device, either as related to particular services, or independent of them.

6. Event Package Definition

The framework specified in this document proposes and specifies a new SIP Event Package as allowed by [RFC3265]. The purpose is to allow for devices to subscribe to specific profile types with PDSs and for the PDSs to notify the devices with - or pointers to - profile data.

The requirements specified in [RFC3265] apply to this package. The following sub-sections specify the Event Package description and the associated requirements. The framework requirements are defined in

Section 5.

6.1. Event Package Name

The name of this package is "ua-profile". This value appears in the Event header field present in SUBSCRIBE and NOTIFY requests for this package as defined in [RFC3265].

6.2. Event Package Parameters

This package defines the following new parameters for the event header:

"profile-type", "vendor", "model", "version", "effective-by", "device-id" and "network-user".

The following rules apply:

- All the new parameters, with the exception of the "effective-by" parameter MUST only be used in SUBSCRIBE requests and ignored if they appear in NOTIFY requests
- The "effective-by" parameter is for use in NOTIFY requests only and MUST be ignored if it appears in SUBSCRIBE requests

The semantics of these new parameters are specified in the following sub-sections.

6.2.1. profile-type

The "profile-type" parameter is used to indicate the token name of the profile type the user agent wishes to obtain data or URIs for and to be notified of subsequent changes. This document defines three logical types of profiles and their token names. They are as follows:

local-network Specifying "local-network" type profile indicates the desire for profile data (URI when content indirection is used) specific to the local network.

device Specifying "device" type profile(s) indicates the desire for the profile data (URI when content indirection is used) and change notification of the contents of the profile that is specific to the device or user agent.

user Specifying "user" type profile indicates the desire for the profile data (URI when content indirection is used) and change notification of the profile content for the user.

The "profile-type" is identified is identified in the Event header parameter: profile-type. A separate SUBSCRIBE dialog is used for each profile type. The profile type associated with the dialog can then be used to infer which profile type changed and is contained in the NOTIFY or content indirection URI. The Accept header of the

SUBSCRIBE request MUST include the MIME types for all profile content types for which the subscribing user agent wishes to retrieve profiles or receive change notifications.

In the following syntax definition using ABNF, EQUAL and token are defined in [RFC3261]. It is to be noted that additional profile types may be defined in subsequent documents.

```
Profile-type   = "profile-type" EQUAL profile-value
profile-value  =  profile-types / token
profile-types  = "device" / "user" / "local-network"
```

The "device", "user" or "local-network" token in the profile-type parameter may represent a class or set of profile properties. Follow-on standards defining specific profile contents may find it desirable to define additional tokens for the profile-type parameter. Also additional content types may be defined along with the profile formats that can be used in the Accept header of the SUBSCRIBE to filter or indicate what data sets of the profile are desired.

6.2.2. vendor, model and version

The "vendor", "model" and "version" parameter values are tokens specified by the implementer of the user agent. These parameters MUST be provided in the SUBSCRIBE request for all profile types. The implementer SHOULD use their DNS domain name (for example, example.com) as the value of the "vendor" parameter so that it is known to be unique. These parameters are useful to the PDS to affect the profiles provided. In some scenarios it is desirable to provide different profiles based upon these parameters. For example, feature property X in a profile may work differently on two versions of the same user agent. This gives the PDS the ability to compensate for or take advantage of the differences. In the following ABNF defining the syntax, EQUAL and quoted-string are defined in [RFC3261].

```
Vendor       =  "vendor" EQUAL quoted-string
Model        =  "model" EQUAL quoted-string
Version      =  "version" EQUAL quoted-string
```

6.2.3. device-id

The "device-id" parameter MUST be set when subscribing for "local-network" profiles. This identifies the device requesting the local-network profile.

If the value of the "profile-type" parameter is not "local-network", the "device-id" parameter has no defined meaning and is ignored. In the following ABNF, EQUAL, LDQUOT, RDQUOT and addr-spec are defined in [RFC3261].

Device-Id = "device-id" EQUAL LDQUOT addr-spec RDQUOT

6.2.4. network-user

The "network-user" parameter MAY be provided in a subscription for a "device" profile. In such cases the device is requesting the PDS to recognize the indicated user as the default user for itself.

If the value of the "profile-type" parameter is not "device", the "network-user" parameter has no defined meaning and is ignored. If the "network-user" parameter is provided in the SUBSCRIBE request, it MUST be present in the NOTIFY request as well. In the following ABNF, EQUAL, LDQUOT, RDQUOT and addr-spec are defined in [RFC3261].

Network-User = "network-user" EQUAL LDQUOT addr-spec RDQUOT

6.2.5. effective-by parameter

The "effective-by" parameter in the Event header of the NOTIFY request specifies the maximum number of seconds before the user agent must attempt to make the new profile effective. The "effective-by" parameter MAY be provided in the NOTIFY request for any of the profile types. A value of 0 (zero) indicates that the subscribing user agent must attempt to make the profiles effective immediately (despite possible service interruptions). This gives the PDS the power to control when the profile is effective. This may be important to resolve an emergency problem or disable a user agent immediately. The "effective-by" parameter is ignored in all messages other than the NOTIFY request. In the following ABNF, EQUAL and DIGIT are defined in [RFC3261].

Effective-By = "effective-by" EQUAL 1*DIGIT

6.2.6. Summary of event parameters

The following are example Event headers which may occur in SUBSCRIBE requests. These examples are not intended to be complete SUBSCRIBE requests.

```
Event: ua-profile;profile-type=device;
       vendor="vendor.example.com";model="Z100";version="1.2.3"

Event: ua-profile;profile-type="user";
```

vendor="premier.example.com";model="trs8000";version="5.5"

The following are example Event headers which may occur in NOTIFY requests. These example headers are not intended to be complete SUBSCRIBE requests.

Event: ua-profile;effective-by=0

Event: ua-profile;effective-by=3600

The following table shows the use of Event header parameters in SUBSCRIBE requests for the three profile types:

profile-type	device	user	local-network
vendor	m	m	m
model	m	m	m
version	m	m	m
device-id			m
network-user	o		
effective-by			

m - mandatory
s - SHOULD be provided
o - optional

Non-specified means that the parameter has no meaning and should be ignored.

The following table shows the use of Event header parameters in NOTIFY requests for the three profile types:

profile-type	device	user	local-network
vendor			
model			
version			
device-id			o
network-user	o		
effective-by	o	o	o

6.3. SUBSCRIBE Bodies

This package defines no use of the SUBSCRIBE request body. If present, it MUST be ignored.

Future enhancements to the framework may specify a use for the SUBSCRIBE request body (for example,, mechanisms using etags to

minimize Profile Notifications to devices with current profile versions).

6.4. Subscription Duration

The duration of a subscription is specific to SIP deployments and no specific recommendation is made by this Event Package. If absent, a value of 86400 seconds is RECOMMENDED since the presence (or absence) of a device subscription is not time critical to the regular functioning of the PDS.

It is to be noted that a one-time fetch of a profile can be accomplished by setting the 'Expires' parameter to a value of Zero, as specified in [RFC3265].

6.5. NOTIFY Bodies

The framework specifying the Event Package allows for the NOTIFY body to contain the profile data or a pointer to the profile data using content indirection. The framework does not define any profile data and delegates specification of utilized MIME types Profile Data Frameworks. For profile data delivered via content indirection, the following apply:

- the Content-ID MIME header, as described in [RFC4483] MUST be used for each Profile document URI
- at a minimum, the "http:" and "https:" URI schemes MUST be supported; other URI schemas MAY be supported based on the Profile Data Frameworks (examples include FTP [RFC0959], HTTP [RFC2616], HTTPS [RFC2818], LDAP [RFC4510], XCAP [I-D.ietf-simple-xcap], XCAP-DIFF [I-D.ietf-simple-xcap-diff])

The NOTIFY body SHOULD include a MIME type specified in the 'Accept' header of the SUBSCRIBE. Further, if the Accept header of the SUBSCRIBE included the MIME type message/external-body (indicating support for content indirection) the content indirection SHOULD be used in the NOTIFY body for providing the profiles. If none are specified, the Profile Data frameworks are responsible for, and MUST specify, the MIME type to be assumed.

6.6. Notifier Processing of SUBSCRIBE Requests

A successful SUBSCRIBE request results in a NOTIFY with either profile contents or a pointer to it (via Content Indirection). If the NOTIFY is expected to contain profile contents or the Notifier is unsure, the SUBSCRIBE SHOULD be either authenticated or transmitted over an integrity protected SIP communication channels. Exceptions

to authenticating such SUBSCRIBEs include cases where the identity of the Subscriber is unknown and the Notifier is configured to accept such requests.

The Notifier MAY also authenticate SUBSCRIBE messages even if the NOTIFY is expected to only contain a pointer to profile data. Securing data sent via Content Indirection is covered in Section 9.

If the profile type indicated in the "profile-type" Event header parameter is unavailable or the Notifier is configured not to provide it, the Notifier SHOULD return a 404 response to the SUBSCRIBE request. If the specific user or device is unknown, the Notifier MAY either accept or reject the subscription.

When the Event header "profile-type" is "device" and the user agent has provided the user's AOR in the "network-user" parameter, the profile delivery server MAY set or change the default user associated with the device indicated in the Subscription request. However, the Notifier SHOULD authenticate the user indicated before making such a change.

6.7. Notifier Generation of NOTIFY Requests

As specified in [RFC3265], the Notifier MUST always send a NOTIFY request upon accepting a subscription. If the device or user is unknown and the Notifier choose to accept the subscription, the Notifier MAY either respond with profile data (for example, default profile data) or provide no profile information (i.e. no body or content indirection).

If the URI in the SUBSCRIBE request is a known identity and the requested profile information is available (i.e. as specified in the profile-type parameter of the Event header), the Notifier SHOULD send a NOTIFY with profile data. Profile data MAY be sent as profile contents or via Content Indirection (if the content indirection MIME type was included in the Accept header). To allow for Content Indirection, the Subscriber MUST support the "http:" or "https:" URI schemas. If the Subscriber wishes to support alternative URI schemas it MUST be indicated in the "schemes" Contact header field parameter as defined in [RFC4483]. If the subscriber does not specify the URI scheme, the Notifier may use either "http:" or "https:".

The Notifier MAY specify when the new profiles must be made effective by the Subscriber by specifying a maximum time in seconds (zero or more) in the "effective-by" event header parameter.

If the SUBSCRIBE was received over an integrity protected SIP communications channel, the Notifier SHOULD send the NOTIFY over the

same channel.

6.8. Subscriber Processing of NOTIFY Requests

A Subscriber to this event package MUST adhere to the NOTIFY request processing behavior specified in [RFC3265]. If the Notifier indicated an effective time (using the "effective-by" Event Header parameter), it SHOULD attempt to make the profiles effective within the specified time. Exceptions include deployments that prohibit such behavior in certain cases (for example, emergency sessions are in progress). When profile data cannot be applied within the recommended timeframe and this affects device behavior, any actions to be taken SHOULD be defined by the profile data definitions. By default, the Subscriber is RECOMMENDED to make the profiles effective as soon as possible.

The Subscriber MUST always support "http:" or "https:" and be prepared to accept NOTIFY messages with those URI schemas.The subscriber MUST also be prepared to receive a NOTIFY request with no body. The subscriber MUST NOT reject the NOTIFY request with no body. The subscription dialog MUST NOT be terminated by a NOTIFY with no body.

6.9. Handling of Forked Requests

This Event package allows the creation of only one dialog as a result of an initial SUBSCRIBE request as described in section 4.4.9 of [RFC3265]. It does not support the creation of multiple subscriptions using forked SUBSCRIBE requests.

6.10. Rate of Notifications

The rate of notifications for the profiles in this framework is deployment specific, but expected to be infrequent. Hence, the Event Package specification does not specify a throttling or minimum period between NOTIFY requests

6.11. State Agents

State agents are not applicable to this Event Package.

7. Examples

This section provides examples along with sample SIP message bodies relevant to this framework. Both the examples are derived from a snapshot of Section 4.1, specifically the request for the device

profile. The examples are purely informative and in case of conflicts with the framework or protocols used for illustration, the latter should be deemed normative.

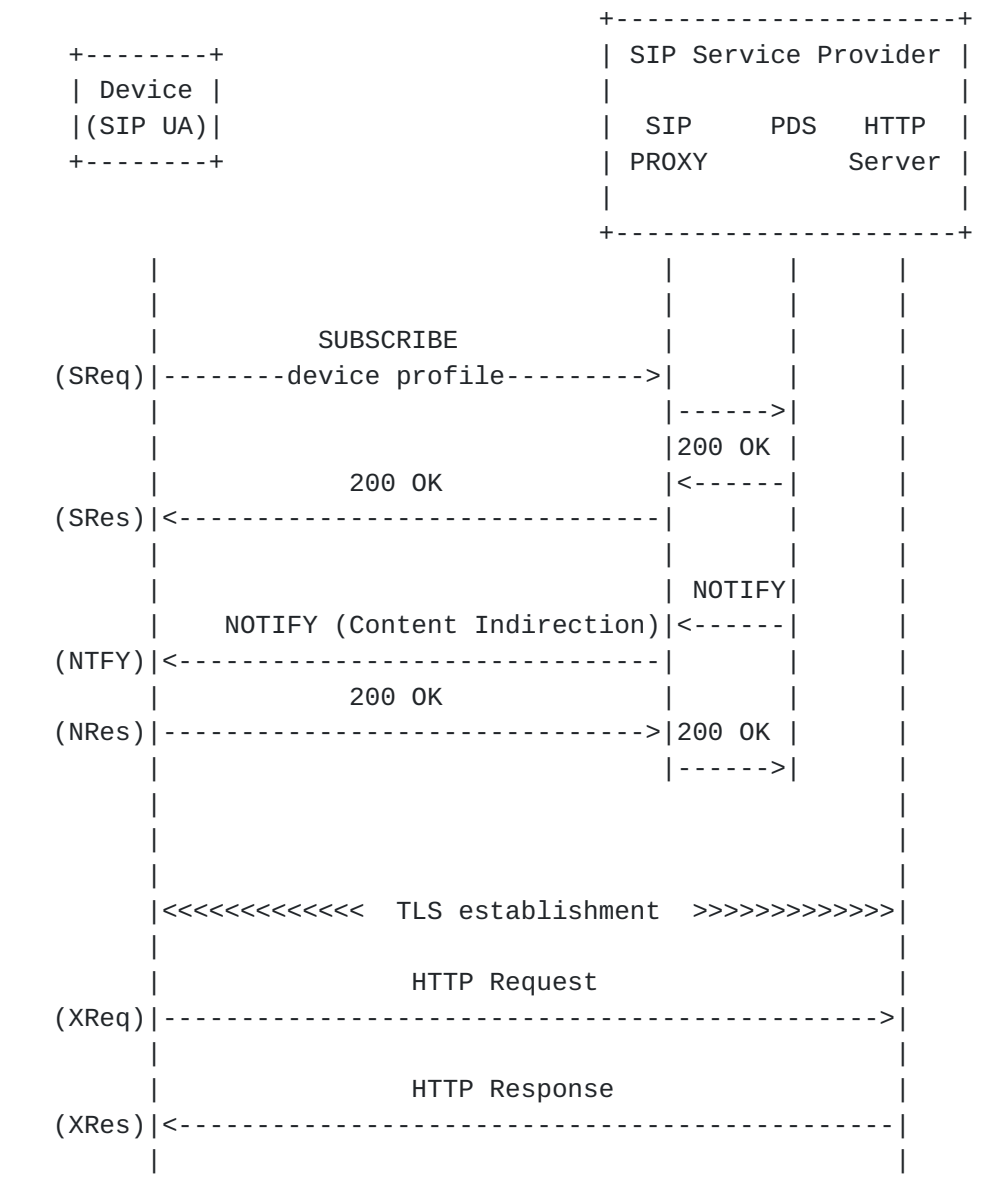
7.1. Example 1: Device requesting profile

This example illustrates the detailed message flows between the device and the SIP Service Provider's network for requesting and retrieving the profile (the flow uses the device profile as an example).

The following are assumed for this example:

- Device is assumed to have established local network connectivity; NAT and Firewall considerations are assumed to have been addressed by the SIP Service Provider
- examples are a snapshot only and do not illustrate all the interactions between the device and the Service Provider's network (and none between the entities in the SIP Service Provider's network)
- All SIP communication with the SIP Service Provider happens via a SIP Proxy
- HTTP is assumed to be the Profile Data method used (any suitable alternative can be used as well)
- TLS is assumed to be the protocol for securing the Profile Content Retrieval (any other suitable protocol can be employed); authentication and security requirements are not addressed

The flow diagram and an explanation of the messages follow.



(SReq)

the device transmits a request for the 'device' profile using the SIP SUBSCRIBE utilizing the Event Package specified in this framework.

* Note: Some of the header fields (for example, Event, via) are continued on a separate line due to format constraints of this document

```
SUBSCRIBE sip:MAC%3a000000000000@sip.example.net SIP/2.0
Event: ua-profile;profile-type=device;vendor="vendor.example.net";
 model="Z100";version="1.2.3";network-user="sip:user@sip.example.net"
From: sip:MAC%3A000000000000@sip.example.net;tag=1234
To: sip:MAC%3A000000000000@sip.example.net
Call-ID: 3573853342923422@192.0.2.44
CSeq: 2131 SUBSCRIBE
Contact: sip:MAC%3A000000000000@sip.example.net
Via: SIP/2.0/TCP 192.0.2.41;
  branch=z9hG4bK6d6d35b6e2a203104d97211a3d18f57a
Accept: message/external-body, application/x-z100-device-profile
Content-Length: 0
```

(SRes)

the SUBSCRIBE request is received by a SIP Proxy in the Service Provider's network which transmits it to the PDS. The PDS accepts the response and responds with a 200 OK

* Note: The device and the SIP proxy may have established a secure communications channel (for example, TLS)

(NTFY)

subsequently, the PDS transmits a SIP NOTIFY message indicating the profile location

* Note: Some of the fields (for example, content-type) are continued on a separate line due to format constraints of this document

```
NOTIFY sip:MAC%3A000000000000@192.0.2.44 SIP/2.0
Event: ua-profile;effective-by=3600
From: sip:MAC%3A000000000000@sip.example.net;tag=abca
To: sip:MAC%3A000000000000@sip.example.net;tag=1231
Call-ID: 3573853342923422@192.0.2.44
CSeq: 322 NOTIFY
Via: SIP/2.0/UDP 192.0.2.3;
  branch=z9hG4bK1e3effada91dc37fd5a0c95cbf6767d0
MIME-Version: 1.0
Content-Type: message/external-body; access-type="URL";
              expiration="Mon, 01 Jan 2010 09:00:00 UTC";
              URL="http://sip.example.net/z100-000000000000.html";
              size=9999;
              hash=10AB568E91245681AC1B

Content-Type: application/x-z100-device-profile
Content-ID: <39EHF78SA@sip.example.net>
.
.
.
```

(NRes)

Device accepts the NOTIFY message and responds with a 200 OK

(XReq)

once the necessary secure communications channel is established, the device sends an HTTP request to the HTTP server indicated in the NOTIFY

(XRes)

the HTTP server responds to the request via a HTTP response containing the profile contents

7.2. Example 2: Device obtaining change notification

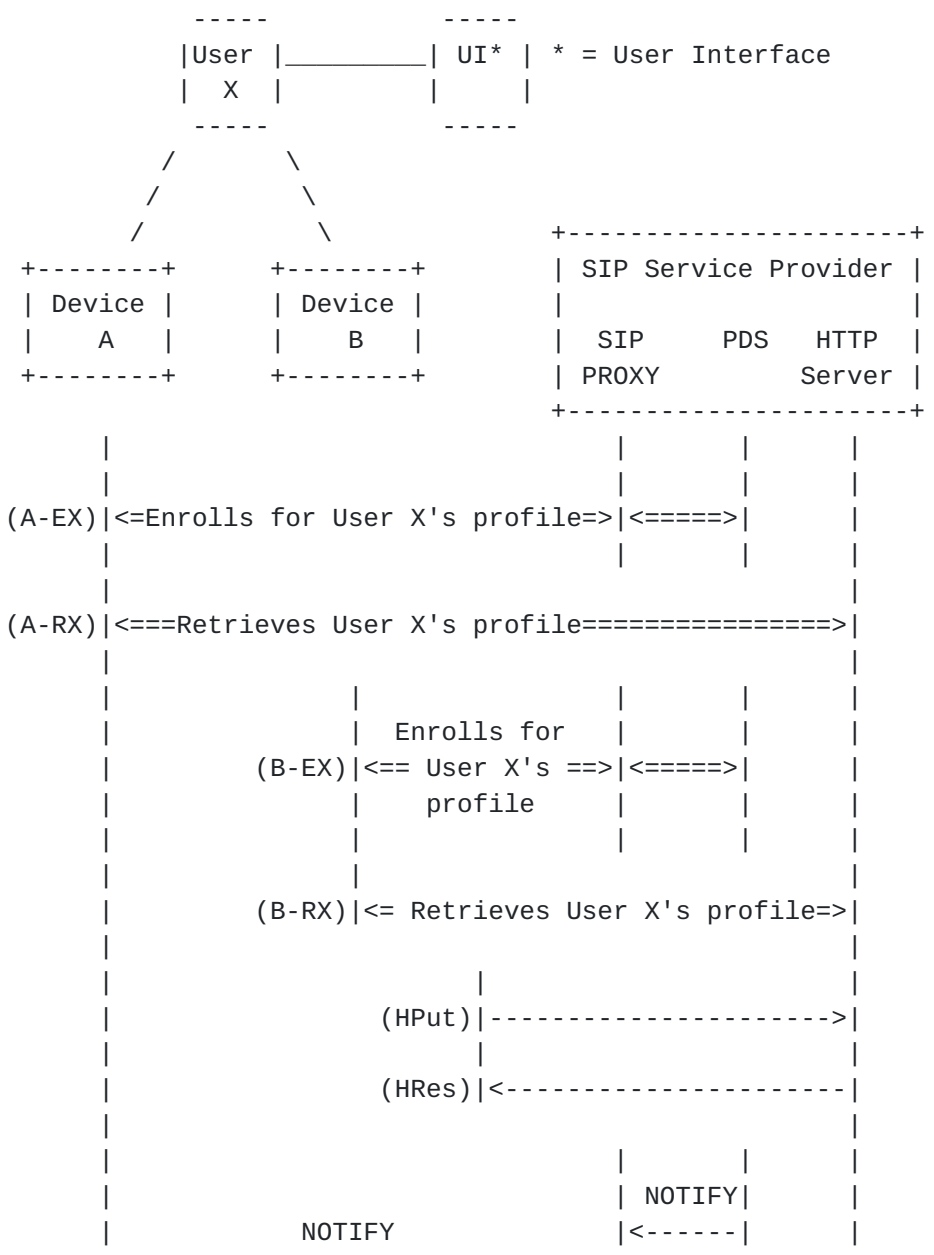
The following example illustrates the case where a user (X) is simultaneously accessing services via two different devices (for example, Multimedia entities on a PC and PDA) and has access to a user Interface (UI) that allows for changes to the user profile.

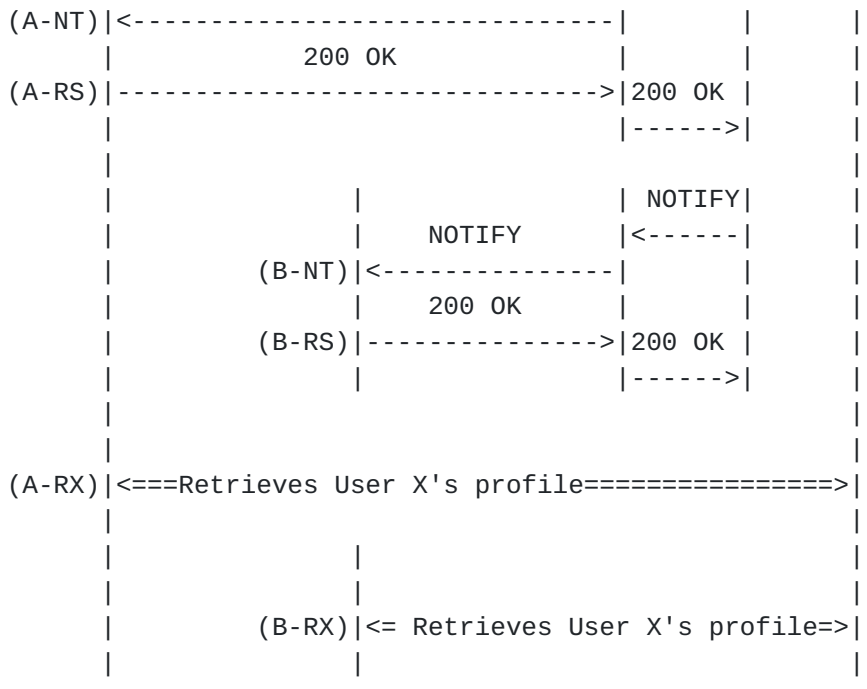
The following are assumed for this example:

- The devices (A & B) obtain the necessary profiles from the same SIP Service Provider
- The SIP Service Provider also provides a user Interface (UI) that allows the user to change preferences that impact the user profile

The flow diagram and an explanation of the messages follow.

- Note: The example only shows retrieval of user X's profile, but it may request and retrieve other profiles (for example, local-network, Device).





(A-EX) Device A discovers, enrolls and obtains notification related to user X's profile

(A-RX) Device A retrieves user X's profile

(B-EX) Device B discovers, enrolls and obtains notification related to user X's profile

(B-RX) Device B retrieves user X's profile

(HPut) Changes affected by the user via the user Interface (UI) are uploaded to the HTTP Server

* Note: The UI itself can act as a device and subscribe to user X's profile. This is not the case in the example shown.

(HRes) Changes are accepted by the HTTP server

(A-NT) PDS transmits a NOTIFY message to device A indicating the changed profile. A sample message is shown below:

Note: Some of the fields (for example, Via) are continued on a separate line due to format constraints of this document

```
NOTIFY sip:userX@192.0.2.44 SIP/2.0
Event: ua-profile;effective-by=3600
From: sip:userX@sip.example.net;tag=abcd
To: sip:userX@sip.example.net.net;tag=1234
Call-ID: 3573853342923422@192.0.2.44
CSeq: 322 NOTIFY
Via: SIP/2.0/UDP 192.0.2.3;
  branch=z9hG4bK1e3effada91dc37fd5a0c95cbf6767d1
MIME-Version: 1.0
Content-Type: message/external-body; access-type="URL";
              expiration="Mon, 01 Jan 2010 09:00:00 UTC";
              URL="http://www.example.com/user-x-profile.html";
              size=9999;
              hash=123456789AAABBBCCCDD
.
.
.
```

(A-RS) Device A accepts the NOTIFY and sends a 200 OK

(B-NT) PDS transmits a NOTIFY message to device B indicating the changed profile. A sample message is shown below:

Note: Some of the fields (for example, Via) are continued on a separate line due to format constraints of this document

```
NOTIFY sip:userX@192.0.2.43 SIP/2.0
Event: ua-profile;effective-by=3600
From: sip:userX@sip.example.net;tag=abce
To: sip:userX@sip.example.net.net;tag=1235
Call-ID: 3573853342923422@192.0.2.43
CSeq: 322 NOTIFY
Via: SIP/2.0/UDP 192.0.2.3;
  branch=z9hG4bK1e3effada91dc37fd5a0c95cbf6767d2
MIME-Version: 1.0
Content-Type: message/external-body; access-type="URL";
              expiration="Mon, 01 Jan 2010 09:00:00 UTC";
              URL="http://www.example.com/user-x-profile.html";
              size=9999;
              hash=123456789AAABBBCCCDD
.
.
.
```

(B-RS) Device B accepts the NOTIFY and sends a 200 OK
(A-RX) Device A retrieves the updated profile pertaining to user X
(B-RX) Device B retrieves the updated profile pertaining to user X

8. IANA Considerations

There are two IANA considerations associated with this document, SIP Event Package and HTTP header. These are outlined in this section.

8.1. SIP Event Package

This specification registers a new event package as defined in [RFC3265]. The following information required for this registration:

Package Name: ua-profile
Package or Template-Package: This is a package
Published Document: RFC XXXX (Note to RFC Editor: Please fill in XXXX with the RFC number of this specification).
Persons to Contact: Daniel Petrie dan.ietf AT SIPez DOT com, sumanth@cablelabs.com
New event header parameters: profile-type, vendor, model, version, effective-by, device-id, network-user (the profile-type parameter has predefined values. The new event header parameters do not)

The following table illustrates the additions to the IANA SIP Header Field Parameters and Parameter Values: (Note to RFC Editor: Please fill in XXXX with the RFC number of this specification)

Header Field	Parameter Name	Predefined Values	Reference
Event	profile-type	Yes	[RFCXXXX]
Event	vendor	No	[RFCXXXX]
Event	model	No	[RFCXXXX]
Event	version	No	[RFCXXXX]
Event	effective-by	No	[RFCXXXX]
Event	device-id	No	[RFCXXXX]
Event	network-user	No	[RFCXXXX]

8.2. New HTTP Event Header

This document defines a new permanent HTTP request header field: Event.

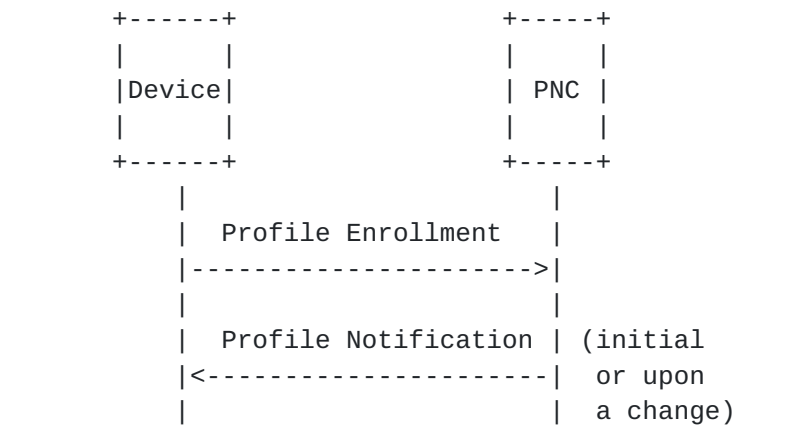
Header field name: Event
Applicable protocol: http
Status: standard

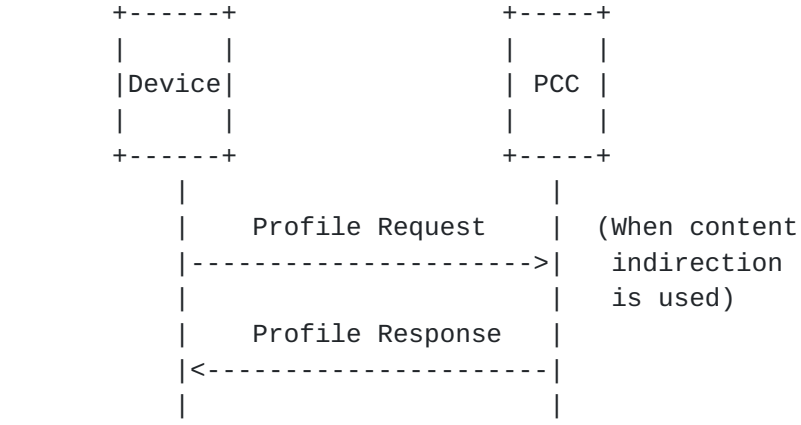
Author/Change controller: IETF
Specification document(s): [RFCXXXX] (Note to RFC Editor: Please fill in XXXX with the RFC number of this specification).

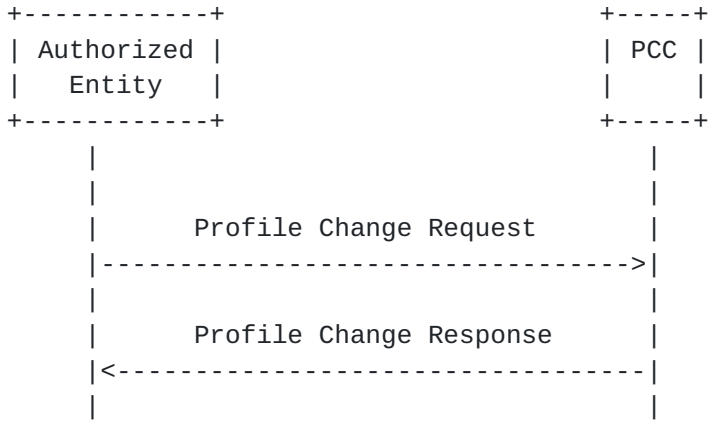
9. Security Considerations

The framework specified in this document allows for the propagation of device profile data (Section 5.5.3). To accomplish this, it specifies a Profile Life Cycle (Section 3.3) and an Event Package (Section 6).

The Profile Life Cycle consists of three distinct communication channels: Profile Enrollment and Change Notification, Profile Content Retrieval, and Profile Change Operation.







PNC = Profile Notification Component
PCC = Profile Content Component

Framework Reference Model

Profile enrollment and change notification allows a device to transmit a request for a specific profile - relayed directly, or via one or more SIP proxies - to a PNC. If the PNC accepts the profile request, it transmits a Profile Notification that contains either: profile data or content indirection information. The profile data can contain information specific to an entity (such as the device or a user) and may contain sensitive information (such as service credentials). Compromise of such data can lead to threats such as impersonation attacks (establishing rogue sessions), theft of service (if services are obtainable), and zombie attacks. Even if the profile data is provided using content indirection, PCC information within the notification can lead to threats such as denial of service attacks (rogue devices bombard the PCC with requests for a specific profile) and attempts to modify erroneous data onto the PCC (since the location and format may be known). It is also important for the device to ensure the authenticity of the PNC since impersonation of the Service Provider can lead to Denial of Service, Man-in-the-Middle attacks, etc.

Profile Content retrieval allows a device to retrieve profile data from a PCC. This communication is accomplished using one of many profile delivery protocols or frameworks, but is considered to be out of scope within this document. However, since the profile data returned is subject to the same considerations as that sent via profile notification, the same threats exist.

Profile Change Operation allows an authorized entity to modify profiles stored on a PCC. The specific entities are based on Service Provider's policy and can include trusted network elements and devices alike. The profile information stored on a PCC can contain information that directs device and user behavior, services offered and may contain sensitive information such as credentials. Thus, allowing entities that are not trusted to perform profile modifications presents threats such as denial-of-service, manipulation of service, impersonation (for example, redirection to rogue networks) and man-in-the-middle attacks.

The framework specified in this document accomplishes the propagation of profile data by utilizing the specified "ua-profile" event package which is based on [RFC3265]. Thus, its usage is expected to comply with the security considerations and requirements (access control, Notifier privacy mechanism, Denial-of-Service attacks, replay attacks, and Man-in-the Middle attacks) specified in Section 5 of [RFC3265]. The remainder of this section presents the specific security requirements for the framework.

9.1. Profile Enrollment and Change Notification

This framework specifies, and allows for the propagation of, three profile types: local-network, device and user. Enrollment and change notification are expected to be accomplished over integrity-protected SIP communication channels and following requirements are presented:

- devices and PNCs complying with this framework MUST implement TLS as specified in [RFC3268], including support for both mutual and one-way authentication (server-side)

- devices and PNCs complying with this framework MUST implement the SIP Digest authentication scheme as specified in [RFC3261]

- a PNC capable of propagating device and user profiles MUST contain a X.509 certificate. This certificate MUST contain the PNC's Fully Qualified Domain Name in the 'SubjectAltName', establishing the PNC as a host in the Service Provider's domain

- a PNC capable of propagating local-network profiles or unauthenticated device profiles MUST support the use of the SIP Identity header as defined in [RFC4474] for inclusion in profile notifications

Each profile type serves a different purpose, and is provided under different circumstances and thus presents slightly different requirements for authentication and protection of communication.

local-network profile

The local-network profile is provided by the local network and usually contains non-sensitive data that is shared among all participants in a local network. However, the framework also allows for the presentation of the user's AOR, if known, for possible privileged user data. This may, or may not, result in user-specific information.

The following requirements are presented:

* the PNC MUST authenticate the identity of the user (if set to anything other than the default) for local-network profile requests that result in user-specific profile data containing sensitive information; for authentication, unless other mechanisms are employed, SIP Digest is used. If the authentication fails, the PNC MUST not include any user-specific information in the local-network profile
* the PNC MAY NOT authenticate requests for the local-network profile that do not result in any user-specific sensitive data (irrespective of the value of the From field)

* the PNC MUST include the SIP Identity header as defined in [RFC4474] within profile notifications sent in response to local-network profile enrollment, unless an integrity-protected channel exists (for example, using S/MIME)
* a device receiving profile notifications for local-network profiles MUST verify the SIP Identity header, unless transmitted over a previously established authenticated, integrity-protected channel. If the header verification fails, the device MUST not use the provided profile and treat it as a local-network profile enrollment failure and take measures such as enrollment retries

device profile

The device profile is expected to contain data specific to the device identity (AOR) being presented in the request. The presented identity may be auto-generated (for example, based on its hardware identity as allowed in section Section 5.1.1.2.1) or obtained via configuration. This identity and associated credentials have the following considerations:

* credentials can be provided via out-of-band mechanisms such as pre-configuration or user interface
* credentials may not be present, but obtained via the initial device profile, if allowed by the Service Provider
* device may use the user's AOR and associated credentials for obtaining the device profile

If the AOR presented in device profile enrollment is known by the PNC, the following requirements are presented:

* the PNC MUST authenticate the AOR presented for enrollment using SIP Digest authentication, unless a previously established mutually authenticated channel exists (for example, using TLS). If the authentication fails, the PNC MUST not provide the requested device-specific profile. In such a scenario, the PNC MAY still provide a generic device profile for minimal services (for example, emergency calls in a telephony deployment, see [I-D.ietf-ecrit-phonebcp])
* if the profile data is provided in the enrollment notificaiton, the PNC MUST transmit it over an integrity-protected, confidential communications channel such as TLS

If the AOR presented in device profile enrollment is not known by the PNC, the following requirements are presented:

* the PNC MUST not provide any sensitive information in the profile data
* the device MUST transmit the request over an integrity-protected SIP communications channel. If none exists, the device MUST establish a TLS connection with the PNC and verify

the PNC's certificate. If the PNC authentication fails or a secure communications channel cannot be established, the device MUST treat it as a device profile enrollment failure and take measures such as retry enrollment

user profile

The user profile is expected to contain data specific to the user identity (AOR) being presented in the request. This identity is expected to be known in the network and associated with credentials. Thus, the following requirements are presented:

* the device MUST transmit the request over an integrity-protected SIP communications channel. If none exists, the device MUST establish a TLS connection with the PNC and verify the PNC's certificate. If the PNC authentication fails or a secure communications channel cannot be established, the device MUST treat this as a user profile enrollment failure and take measures such as retry enrollment
* the PNC MUST authenticate the AOR presented for enrollment using SIP Digest authentication, unless a previously established mutually authenticated channel exists (for example, using TLS). If the user authentication fails, the PNC MUST not provide the requested user-specific information. It MAY provide minimal profile information (such as connection to a customer support webpage)
* if the profile data is provided in the enrollment notificaiton, the PNC MUST transmit it over an integrity-protected, confidential communications channel such as TLS

9.2. Profile Content Retrieval

This framework does not mandate specific profile delivery frameworks, but presents security requirements for profile content retrieval using content indirection. Given the nature of the profiles, the requirements are as follows:

o devices and PCCs compliant with this framework MUST implement HTTP Digest authentication as specified in [RFC2617]; this is used whenever an authentication challenge is initiated using HTTP based protocols specified for interoperability
o a PCC complying with this framework MUST implement HTTPS [RFC2818]; this is used when there are no existing integrity-protected communication channels
o a PCC complying with this framework MUST contain a X.509 certificate. This certificate MUST contain the PNC's Fully Qualified Domain Name in the 'SubjectAltName', establishing the PNC as a host in the Service Provider's domain

The following general requirement applies to all profile types:

- a device MUST request profile content retrieval over an integrity protected channel such as HTTPS. If one does not exist or cannot be established, then the device MUST treat this as a profile content retrieval failure and take measures such as profile content retrieval retries or in the case of retry exhaustion, try enrollment

The following profile-specific usage requirements are presented

local-network profile

  * a PCC MUST challenge a profile content retrieval request if the profile data contains user-specific information; this challenge is against a user's AOR, known by the PCC and the device
  * a PCC MAY challenge a profile content retrieval request even if the profile data contains user-specific information; this challenge is against a user's AOR, if provided

device profile

  * a PCC MUST authenticate a profile content retrieval request if the AOR presented is known. If the authentication fails, the PCC MUST not provide device-specific information. In such a scenario, the PCC MAY still provide a generic device profile for minimal services (for example, emergency calls in a telephony deployment, see [I-D.ietf-ecrit-phonebcp])

user profile

  * a PCC MUST authenticate a profile content retrieval request. If the user authentication fails, the PNC MUST not provide the requested user-specific information. It MAY provide minimal profile information (such as connection to a customer support webpage)

9.3. Profile Change Operation

Changes to profiles will only be made by authorized entities and requires mutual authentication. The following requirements are presented:

- a PCC complying with this framework MUST contain a X.509 certificate. This certificate MUST contain the PNC's Fully Qualified Domain Name in the 'SubjectAltName', establishing the PNC as a host in the Service Provider's domain. This may be the same, or different, from the certificate used for profile content retrieval
- an entity that is allowed to make updates MUST contain a AOR that is known to the network and the requirements for making changes are the same as that for user profile content retrieval, with the

authorized entity playing the role of a user

10. Acknowledgements

Many thanks to those who contributed and commented on the many iterations of this document. Detailed comments were provided by the following individuals: Jonathan Rosenberg from Cisco, Henning Schulzrinne from Columbia University, Cullen Jennings from Cisco, Rohan Mahy from Plantronics, Rich Schaaf from Pingtel, Volker Hilt from Bell Labs, Adam Roach of Estacado Systems, Hisham Khartabil from Telio, Henry Sinnreich from MCI, Martin Dolly from AT&T Labs, John Elwell from Siemens, Elliot Eichen and Robert Liao from Verizon, Dale Worley from Pingtel, Francois Audet from Nortel, Roni Even from Polycom, Jason Fischl from Counterpath, Josh Littlefield from Cisco, Nhut Nguyen from Samsung.

The editor would like to extend a special thanks to the experts who contributed to the restructuring and revisions as proposed by the SIPPING WG, specifically Keith Drage from Lucent (restructuring proposal), Peter Blatherwick from Mitel (who also contributed to the Overview and Introduction sections), Josh Littlefield from Cisco (examples and diagram suggestions), Alvin Jiang of Engin, Martin Dolly from AT&T, Jason Fischl from Counterpath, Donald Lukacs from Telcordia and Eugene Nechamkin from Broadcom. Additionally, sincere appreciation is extended to the chairs (Mary Barnes from Nortel and Gonzalo Camarillo from Ericsson) and the Area Directors (Cullen Jennings from Cisco and Jon Peterson and Cisco) for facilitating discussions, and for reviews and contributions.

11. Change History

[[RFC Editor: Please remove this entire section upon publication as an RFC.]]

11.1. Changes from draft-ietf-sipping-config-framework-10.txt

The following are the changes that have been incorporated into this I-D, resulting from the design team discussions based on Working Group feedback.

- Modified the "From" header of the local network profile to reflect the user's AOR, if any; delegated the device identifier to a new event header termed "device-id"; removed use for 'network-user' within the local-network profile; if there are objections to this, please educate us!

- Added text to indicate DHCPv4 or DHCPv6 to accomodate IPv4 and IPv6 environments
- Replaced generic 'Service Provider' with terms to better represent scenarios
- Analyzed the current SHOULD v/s MUST requirements for the Profile Framework and made modifications
- Referenced RFC4122 instead of OUTBOUND
- Simplified the introductory sections to better illustrate potential deployment possibilities; indicated the minimum profile supported to be 'device'
- Revamped the security considerations sections

11.2. Changes from draft-ietf-sipping-config-framework-09.txt

Following the ad-hoc SIPPING WG discussions at IETF#67 and as per the email from Gonzalo Camarillo dated 12/07/2006, Sumanth was appointed as the new editor. This sub-section highlights the changes made by the editor (as per expert recommendations from the SIPPING WG folks interested in this effort) and the author.

Changes incorporated by the editor:

- Document was restructured based on a) Keith's recommendations in the email dated 11/09/2006 and responses (Peter, Sumanth, Josh) b) subsequent discussions by the ad-hoc group consisting of the editor, the author, expert contributors (Peter Blatherwick, Josh Littlefield, Alvin Jiang, Jason Fischl, Martin Dolly, Cullen Jennings) and the co-chairs . Further changes follow.
- Use cases were made high-level with detailed examples added later on
- Several sections were modified as part of the restructuring (for example, Overview, Introduction, Framework Requirements, Security Sections)
- General editorial updates were made

Changes incorporated by the author:

- Incorporated numerous edits and corrections from CableLabs review.
- Used better ascii art picture of overview from Josh Littlefield
- Fixed the normative text for network-user so that it is now consistant: MAY provide for device profile, MUST provide for local-network profile.

11.3. Changes from draft-ietf-sipping-config-framework-08.txt

The Request URI for profile-type=localnet now SHOULD not have a user part to make routing easier. The From field SHOULD now contain the device id so that device tracking can still be done.
Described the concept of profile-type as a filter and added normative text requiring 404 for profile types not provided.
Moved "application" profile type to draft-ietf-sipping-xcap-config-01. The "application" value for the profile-type parameter will also be used as a requirement that XCAP be supported.
Fixed text on certificate validation.
Added new HTTP header: Event to IANA section and clean up the IANA section.
Added diagram for Service Provider use case schenario.
Added clarification for HTTP Event header.
Added clarification of subscriber handling of NOTIFY with no body.

11.4. Changes from draft-ietf-sipping-config-framework-07.txt

Made XCAP informative reference. Removed "document" and "auid" event header parameters, and Usage of XCAP section to be put in separate supplementary draft.
Fixed ABNF for device-id to be addr-spec only (not name-addr) and to be quoted as well.
Synchronized with XCAP path terminology. Removed XCAP path definition as it is already defined in XCAP.
User agent instance ID is now defined in output (not GRUU).
Clarified the rational for the device-id parameter.
Added text to suggest URIs for To and From fields.
Clarified use of device-id parameter.
Allow the use of the auid and document parameters per request by the OMA.

11.5. Changes from draft-ietf-sipping-config-framework-06.txt

Restructured the introduction and overview section to be more consistent with other Internet-Drafts.
Added additional clarification for the Digest Authentication and Certificate based authentication cases in the security section.
Added two use case scenarios with cross referencing to better illustrate how the framework works. Added better cross referencing in the overview section to help readers find where concepts and functionality is defined in the document.
Clarified the section on the use of XCAP. Changed the Event parameter "App-Id" to "auid". Made "auid" mutually exclusive to "document". "auid" is now only used with XCAP.
Local network subscription URI changed to <device-id>@<local-network> (was anonymous@<local-network>). Having a different Request URI for each device allows the network

management to track user agents and potentially manage bandwidth, port allocation, etc.
Changed event package name from sip-profile to ua-profile per discussion on the list and last IETF meeting.
Changed "local" profile type token to "local-network" per discussion on the list and last IETF meeting.
Simplified "Vendor", "Model", "Version" event header parameters to allow only quoted string values (previously allowed token as well).
Clarified use of the term cache.
Added references for ABNF constructs.
Numerous editorial changes. Thanks Dale!

11.6. Changes from draft-ietf-sipping-config-framework-05.txt

Made HTTP and HTTPS profile transport schemes mandatory in the profile delivery server. The subscribing device must implement HTTP or HTTPS as the profile transport scheme.
Rewrote the security considerations section.
Divided references into Normative and Informative.
Minor edits throughout.

11.7. Changes from draft-ietf-sipping-config-framework-04.txt

Clarified usage of instance-id
Specify which event header parameters are mandatory or optional and in which messages.
Included complete list of event header parameters in parameter overview and IANA sections.
Removed TFTP reference as protocol for profile transport.
Added examples for discovery.
Added ABNF for all event header parameters.
Changed profile-name parameter back to profile-type. This was changed to profile-name in 02 when the parameter could contain either a token or a path. Now that the path is contained in the separate parameter: "document", profile-type make more sense as the parameter name.
Fixed some statements that should have and should not have been normative.
Added the ability for the user agent to request that the default user associated with the device be set/changed using the "device-id" parameter.
A bunch of editorial nits and fixes.

11.8. Changes from draft-ietf-sipping-config-framework-03.txt

Incorporated changes to better support the requirements for the use of this event package with XCAP and SIMPLE so that we can have one

package (i.e. simple-xcap-diff now defines a content type not a package). Added an additional profile type: "application". Added document and app-id Event header parameters in support of the application profile. Define a loose high level data model or relationship between the four profile types. Tried to edit and fix the confusing and ambiguous sections related to URI formation and discovery for the different profile types. Better describe the importance of uniqueness for the instance id which is used in the user part of the device URI.

11.9. Changes from draft-ietf-sipping-config-framework-02.txt

Added the concept of the local network as a source of profile data. There are now three separate logical sources for profile data: user, device and local network. Each of these requires a separate subscription to obtain.

11.10. Changes from draft-ietf-sipping-config-framework-01.txt

Changed the name of the profile-type event parameter to profile-name. Also allow the profile-name parameter to be either a token or an explicit URI.

Allow content indirection to be optional. Clarified the use of the Accept header to indicate how the profile is to be delivered.

Added some content to the Iana section.

11.11. Changes from draft-ietf-sipping-config-framework-00.txt

This version of the document was entirely restructured and re-written from the previous version as it had been micro edited too much.

All of the aspects of defining the event package are now organized in one section and is believed to be complete and up to date with [RFC3265].

The URI used to subscribe to the event package is now either the user or device address or record.

The user agent information (vendor, model, MAC and serial number) are now provided as event header parameters.

Added a mechanism to force profile changes to be make effective by the user agent in a specified maximum period of time.

Changed the name of the event package from sip-config to ua-profile

Three high level security approaches are now specified.

11.12. Changes from draft-petrie-sipping-config-framework-00.txt

Changed name to reflect SIPPING work group item

Synchronized with changes to SIP DHCP [RFC3361], SIP [RFC3261] and [RFC3263], SIP Events [RFC3265] and content indirection [RFC4483]

Moved the device identity parameters from the From field parameters to user-agent header parameters.

Many thanks to Rich Schaaf of Pingtel, Cullen Jennings of Cisco and Adam Roach of Estacado Systems for the great comments and input.

11.13. Changes from draft-petrie-sip-config-framework-01.txt

Changed the name as this belongs in the SIPPING work group.

Minor edits

11.14. Changes from draft-petrie-sip-config-framework-00.txt

Split the enrollment into a single SUBSCRIBE dialog for each profile. The 00 draft sent a single SUBSCRIBE listing all of the desired. These have been split so that each enrollment can be routed differently. As there is a concept of device specific and user specific profiles, these may also be managed on separate servers. For instance in a nomadic situation the device might get its profile data from a local server which knows the LAN specific profile data. At the same time the user specific profiles might come from the user's home environment profile delivery server.

Removed the Config-Expires header as it is largely superfluous with the SUBSCRIBE Expires header.

Eliminated some of the complexity in the discovery mechanism.

Suggest caching information discovered about a profile delivery server to avoid an avalanche problem when a whole building full of devices powers up.

Added the user-profile From header field parameter so that the device can request a user specific profile for a user that is different from the device's default user.

12. References

12.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[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.

[RFC3263] Rosenberg, J. and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", RFC 3263, June 2002.

[RFC3265] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002.

[RFC3268] Chown, P., "Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)", RFC 3268, June 2002.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3319] Schulzrinne, H. and B. Volz, "Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiation Protocol (SIP) Servers", RFC 3319, July 2003.

[RFC3361] Schulzrinne, H., "Dynamic Host Configuration Protocol (DHCP-for-IPv4) Option for Session Initiation Protocol (SIP) Servers", RFC 3361, August 2002.

[RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, July 2005.

[RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, August 2006.

[RFC4483] Burger, E., "A Mechanism for Content Indirection in Session Initiation Protocol (SIP) Messages", RFC 4483, May 2006.

12.2. Informative References

[I-D.ietf-ecrit-phonebcp] Rosen, B. and J. Polk, "Best Current Practice for Communications Services in support of Emergency Calling", draft-ietf-ecrit-phonebcp-00 (work in progress), October 2006.

[I-D.ietf-simple-xcap] Rosenberg, J., "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)", draft-ietf-simple-xcap-12 (work in progress), October 2006.

[I-D.ietf-simple-xcap-diff] Rosenberg, J., "An Extensible Markup Language (XML) Document Format for Indicating A Change in XML Configuration Access Protocol (XCAP) Resources", draft-ietf-simple-xcap-diff-04 (work in progress), October 2006.

[RFC0959] Postel, J. and J. Reynolds, "File Transfer Protocol", STD 9, RFC 959, October 1985.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC2141] Moats, R., "URN Syntax", RFC 2141, May 1997.

[RFC4510] Zeilenga, K., "Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map", RFC 4510, June 2006.

Authors' Addresses

Daniel Petrie
SIPez LLC.
34 Robbins Rd
Arlington, MA 02476
USA

Email: dan.ietf AT SIPez DOT com
URI: http://www.SIPez.com/

Sumanth Channabasappa (Editor)
CableLabs
858 Coal Creek Circle
Louisville, Co 80027
USA

Email: sumanth@cablelabs.com
URI: http://www.cablelabs.com/

Full Copyright Statement

Intellectual Property

Acknowledgment

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