Service Provider Device Configuration
Version 2.0
28 February 2015

This is a Non-binding Permanent Reference Document of the GSMA

Security Classification: Confidential - Full, Rapporteur, and Associate Members

Copyright Notice

Disclaimer

Antitrust Notice

Table of Contents

1 Introduction

1.1 Overview

This document describes an Over The Air (OTA) mechanism that allows a Service Provider to provision mobile and non-mobile devices with the necessary configurations to use their services. It provides an alternative to the Open Mobile Alliance's (OMA) Device Management (DM) approach. For transport, the mechanism mainly relies on the Hyper-Text Transfer Protocol (HTTP).

This configuration can be initiated both from the device and from the network. It allows configuration both over Service Provider controlled access networks (e.g. cellular) and non-Service Provider controlled networks (e.g. a 3rd party provided WLAN [Wireless Local Area Network]). It also allows for the provision of messages from the Service Provider to the user potentially requiring acceptance before the provided configuration can be used.

1.2 Scope

This document covers both the device and network aspects of the configuration. It only describes the generic parts of the configuration. Service specific aspects need to be described in documents relating to that service (for example PRD [Permanent Reference Document] RCC.07 for RCS [Rich Communication Services] based services). It only covers the UNI (User-Network Interface) aspects and does not deal with the internal network and device aspects of the provisioning.

1.3 Abbreviations

Term	Description
AKA	Authentication and Key Agreement
APN	Access Point Name
AuC	Authentication Centre
BSF	Bootstrapping Server Function
B-TID	Bootstrapping Transaction Identifier
CA	Certification Authority
DM	Device Management
DNS	Domain Name System
GAA	Generic Authentication Architecture
GBA	Generic Bootstrapping Architecture
HPLMN	Home Public Land Mobile Network
HTTP	Hyper-Text Transfer Protocol
HTTPS	Hyper-Text Transfer Protocol Secure
IMEI	International Mobile Station Equipment Identity
IMPI	Internet Protocol Multimedia Subsystem Private Identity
IMS	Internet Protocol Multimedia Subsystem
IMSI	International Mobile Subscriber Identity

Term	Description
IP	Internet Protocol
MCC	Mobile Country Code
MNC	Mobile Network Code
MSISDN	Mobile Subscriber Integrated Services Digital Network Number
OMA	Open Mobile Alliance
OMA-CP	Open Mobile Alliance Client Provisioning
OMA-DM	Open Mobile Alliance Device Management
OTA	Over The Air
OTP	One Time Password
PC	Personal Computer
PRD	Permanent Reference Document
PS	Packet Switched
RADIUS	Remote Authentication Dial In User Service
RCS	Rich Communication Services
SIM	Subscriber Identity Module
SIP	Session Initiation Protocol
SMS	Short Message Service
UCS2	2-byte Universal Character Set
UDH	User Data Header
UI	User Interface
UNI	User-Network Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UX	User Experience
Wi-Fi	Trademark of Industry Consortium "Wi-Fi Alliance" used as synonym for WLAN (Wireless Local Area Network)
WLAN	Wireless Local Area Network
XML	Extensible Markup Language

1.4 References

Ref	Doc Number	Title
[1]	[3GPP TS 23.003]	3GPP TS 23.003 Release 10, 3rd Generation Partnership Project; Numbering, addressing and identification http://www.3gpp.org
[2]	[3GPP TS 23.040]	3GPP TS 23.040 Release 10, 3rd Generation Partnership Project; Technical realization of the Short Message Service (SMS) http://www.3gpp.org
[3]	[3GPP TS 24.109]	3GPP TS 24.109 Release 10, 3rd Generation Partnership Project; Bootstrapping interface (Ub) and network application function interface (Ua); Protocol details

Ref	Doc Number	Title
		http://www.3gpp.org
[4]	[3GPP TS 33.220]	3GPP TS 33.220 Release 10, 3rd Generation Partnership Project; Generic Authentication Architecture (GAA);Generic Bootstrapping Architecture (GBA) http://www.3gpp.org
[5]	[PRD-IR.67]	GSMA PRD IR.67 - "DNS/ENUM Guidelines for Service Providers & GRX/IPX Providers" Version 8.0 22 November 2012 http://www.gsma.com/
[6]	[PRD-RCC.15]	GSMA PRD RCC.15 IMS Device Configuration and Supporting Services, Version 1.0, xx October 2014 http://www.gsma.com
[7]	[RFC2119]	"Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997. Available at http://www.ietf.org/rfc/rfc2119.txt
[8]	[RFC3310]	Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA): Generic Syntax IETF RFC http://tools.ietf.org/html/rfc3310
[9]	[RFC3986]	Uniform Resource Identifier (URI): Generic Syntax IETF RFC http://tools.ietf.org/html/rfc3986
[10]	[OMA CP Cont]	Provisioning Content, Approved Version 1.1 – 28 Jul 2009 OMA-WAP-TS-ProvCont-V1_1-20090728-A http://www.openmobilealliance.com

1.5 Conventions

"The key words "must", "must not", "required", "shall", "shall not", "should", "should not", "recommended", "may", and "optional" in this document are to be interpreted as described in [RFC2119]."

2 HTTP Configuration

2.1 Overview

This mechanism is based on HTTP(S) (Hyper-Text Transfer Protocol Secure) requests sent by a device to a Service Provider's configuration server in order to receive the configuration data.

The HTTP(S) configuration requests may be triggered in two different ways:

- Client-triggered HTTP(S) configuration if a Service Provider supporting this mechanism is detected by the client (e.g. SIM-based or by customization).
- Network-triggered HTTP(S) configuration if a Service Provider is not detected by the client. It is used to protect against negative charging impacts in networks that do not support this type of configuration.

Client behaviour is as follows:

- If client-triggered configuration applies: when a device boots up (or when the Subscriber Identity Module [SIM] is swapped without rebooting the device [hot swap]) and no valid configuration is available for the used identity, the device sends an initial HTTP request toward the Service Provider's configuration server to verify the current configuration settings' version.
    - If a non-embedded mobile client or a Personal Computer (PC) client without a SIM has no valid configuration for the used identity, this check should be performed each time the client is started.
- After receiving a Short Message Service (SMS) trigger as described in section 3, there is an HTTP request sent to the Service Provider's configuration server to verify the current configuration settings' version.
    - If the version available on the client does not match the version on the configuration server, the configuration server will include in its response to the client's HTTP request a configuration document in Extensible Markup Language (XML) format containing all configuration settings.

NOTE1: The configuration document is covered in detail in section 4 and is based on the OMA Client Provisioning (OMA-CP) syntax (see [OMA CP Cont]).

- In situations where it is necessary to force a reconfiguration of a device (e.g. SIM card swap), the device resets the version value of its on-hand configuration settings to 0. The server configuration shall therefore always provide a version value greater than 0.
- In scenarios where the Service Provider desires that for all functionality on a device/client that is subject to configuration the device returns to its default state, the HTTP response provided by the configuration server will carry an XML configuration response that carries no configuration parameters and sets the version of the configuration settings to 0, -1 or -2. That default state will be service dependent and may be to simply disable the service. That will be defined in the service specific documents for each service supporting this mechanism.

The details on the exchanges (e.g. the format employed for each requests) are provided in sections 2.2.1, 2.2.2, 2.2.3, 2.2.4, 2.2.5 and 2.5 of this specification.

2.2 Configuration over cellular networks

This section describes configuration of a device carrying the SIM associated with the identity to be used for the configured services over cellular networks and other Service Provider controlled access networks that allow identifying the user's identity based on the access. The section also introduces the general principles of the HTTP based configuration mechanism that are applicable for all access networks.

This HTTP configuration mechanism operates in these circumstances under the following assumptions:

As a security measure and to ensure that a Service Provider is able to implement the necessary procedures to resolve a user's Mobile Subscriber Integrated Services Digital Network Number (MSISDN) (that is Remote Authentication Dial In User Service (RADIUS)

requests, header enrichment and so on), the configuration of devices/clients carrying the SIM associated to a user's main identity can only occur if the device is connected using a mobile PS[1] data network or by using the procedure in section 2.3.3.3 over other networks and, therefore, the device should have the necessary Access Point Name (APN) configuration available to perform the connection.

NOTE: For other devices/clients the mechanisms defined in section 2.5 are used.

- As some of the mechanisms presented in the previous bullet require an initial HTTP request, an HTTP request is performed first:
    - The device/client shall send an HTTP GET request towards the configuration server's qualified domain name. In this initial HTTP GET request the GET parameters outlined in Table 1 should not be included.
    - As a result of successfully receiving and processing this request, the configuration server returns an HTTP 200 OK response.
    - Upon receiving that HTTP 200 OK response, the device shall perform a second GET request towards the same Uniform Resource Locator (URL) (i.e. the configuration server's qualified domain name) using the HTTPS protocol.
    - The configuration server should be able to correlate both HTTP and HTTPS requests from the same device. To achieve this, the configuration server shall provide a cookie as part of the response to the initial HTTP request (Set-Cookie header). The configuration server will expect the client to provide that cookie in the subsequent HTTPS request (in the Cookie header).
- From a User Experience (UX) perspective, the customer is not aware of the auto-configuration process (it is a background process with no pop-ups, alerts or notifications shown to the user on the screen of the device) unless the provisioned data includes a message for the end user.

[1] Please note that if a device does not have a Packet Switched (PS) connection, the auto-configuration can also happen over Wi-Fi. The decision to implement this mechanism is up to the discretion of each Service Provider.

2.2.1 Initial Request

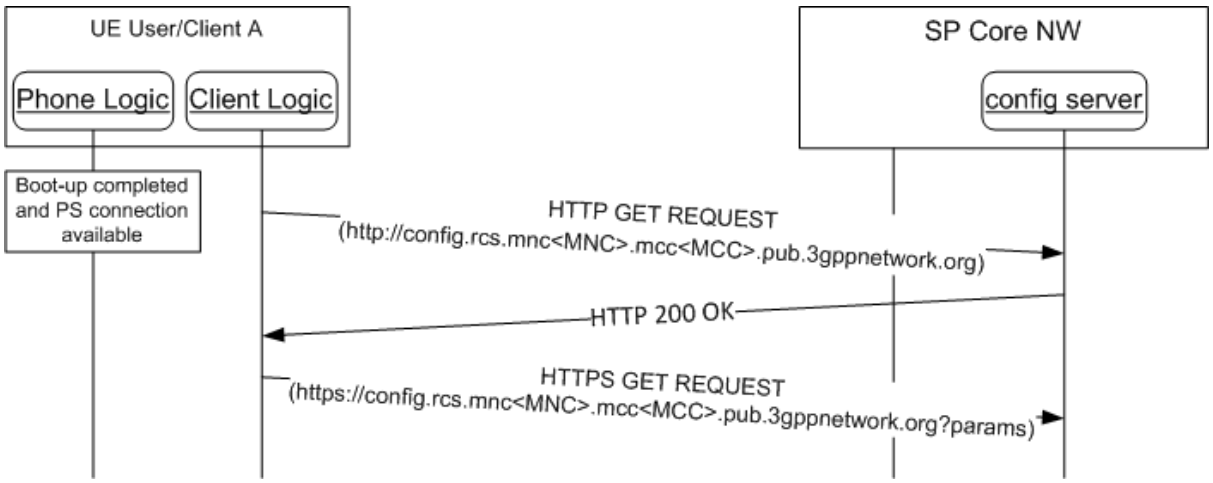

Figure 1: HTTP configuration: Initial requests

Parameters: The following information is included as HTTP GET parameters using a query string:

Parameter	Description	Mandatory	Format
vers	This is either -3, -2, -1, 0 or a positive integer. 0 indicates that the configuration must be updated (e.g. the configuration is damaged; non-existent or an update is needed following a SIM change). A positive value indicates the version of the static parameters (those which are not user dependent) so the server can evaluate whether an update is required. -1 indicates that the device/client has provides the default behaviour for the services that would be configured and has disabled the autoconfiguration query performed at boot. This may be used by the client/device to inform the SP that the functionality was permanently disabled from the device. -2 Indicates that for the services to be configured the default behaviour needs to be provided (including the disabling of the configuration query at boot), but a configuration query might be triggered on user action.	Y	Int (-3, -2, -1, 0 or a positive integer)
IMSI	If available, the subscriber's IMSI (International Mobile Subscriber Identity) shall be sent as a parameter.	N if the OS platform allows it, it shall be included	String (15 digits)
provisioning_version	String that identifies the version of this provisioning specification supported by the client. It shall be set to "*1.0*" (without the	Y	String (4 max), Case-Sensitive

	quotes) for clients following this specification.		
terminal_vendor	String that identifies the terminal OEM.	Y	String (4 max), Case-Sensitive
terminal_model	String that identifies the terminal model.	Y	String (10 max), Case-Sensitive
terminal_sw_version	String that identifies the terminal software version.	Y	String (10 max), Case-Sensitive
IMEI	If available, the subscriber's International Mobile Station Equipment Identity (IMEI) shall be sent as a parameter. Those Service Providers that support a comprehensive device database can ignore the terminal_X parameters and use the IMEI instead, if it was available to the implementation.	N if the OS platform allows it, it shall be included	String (15 digits)
friendly_device_name	If provided by the user, a user friendly identification for the device may be passed along that can be used by the network when presenting the user with an overview of their devices. NOTE: this parameter needs to be included only if required for one of the services to be configured. In which case its mandatory character will be documented in the relevant service specific documents.	N, only to be provided if provided by the user	String (30 max before escaping), Case-Sensitive

Table 1: HTTP configuration: HTTPS request GET parameters

NOTE: a service specific document could define additional parameters to be included

Please note that in case of Service Provider-specific clients, the terminal vendor, model and version parameters format and values should be agreed with the associated Service Provider prior to any device or client commercialization or update.

- The configuration server URL shall be composed based on the home Service Provider's MCC (Mobile Country Code) and MNC (Mobile Network Code) using a "*config*" subdomain of the domain reserved for RCS services in [PRD-IR.67][2]. That is:
  http://config.rcs.mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org
  whereby <MNC> and <MCC> shall be replaced by the respective values of the home network in decimal format and with a 2-digit MNC padded out to 3 digits by inserting a 0 at the beginning (as defined in [PRD-IR.67]).
- The client shall check the MCC and the MNC in the IMSI and compose the configuration Server URL introduced in the previous bullet depending on the Home Public Land Mobile Network (HPLMN).

[2] The RCS domain is used for historic reasons

- If a device is employed by a Service Provider that does not support this configuration mechanism, the configuration server URL will not be resolved. In that scenario the application shall handle it as a “client configuration invalid” scenario.

2.2.2 Configuration server response

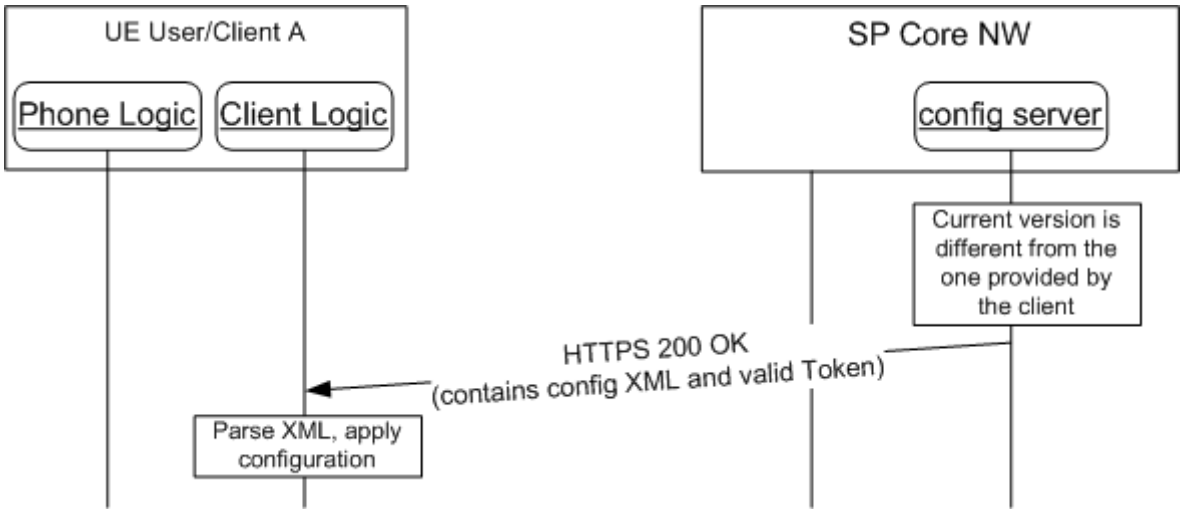

Figure 2: HTTP configuration: Server response

The configuration server (in response to the HTTPS request from a device) shall first validate the client and terminal parameters and then check if the version provided by the client matches the latest version of the configuration available on the server.

The response shall always contain the following parameters:

1. The configuration version
2. The validity of the configuration in seconds
3. The generated TOKEN (including value and validity as illustrated in Table 2).

If the version matches (i.e. no new configuration settings required), the configuration XML document shall be empty except for the version and the validity parameters:

- The version parameter shall be set to the same value X (as illustrated in Table 2) provided by the client in the HTTPS request
- The validity parameter shall be reset to a server configured value Y (as illustrated in Table 2)

```
<?xml version="1.0"?>
<wap-provisioningdoc version="1.1">
        <characteristic type="VERS">
                <parm name=”version” value=”X”/>
                <parm name=”validity” value=”Y”/>
        </characteristic>
        <characteristic type="TOKEN">
                <parm name=”token” value=”Z”/>
                <parm name=”validity” value=”W”/>
        </characteristic>
</wap-provisioningdoc>
```

Table 2: HTTP configuration XML: no configuration changes required

The token shall be stored on the device so it can be used in subsequent configuration requests over non-3GPP access (see section 2.3.2). This value shall be removed together with the rest of the configuration when the device or client is reset.When the client is enabled (i.e. last received configuration had a positive value for version and validity), a change by the user in a device setting that that would result in a different value for one of the generic or service specific parameters shall also trigger a configuration query.

If the Service Provider chooses to temporarily revert the configured functionality on the device/client to its default behaviour, the response shall carry an XML document containing only the version and validity, both set to 0 as illustrated in Table 3:

```
<?xml version="1.0"?>
<wap-provisioningdoc version="1.1”>
        <characteristic type="VERS">
                <parm name=”version” value=”0”/>
                <parm name=”validity” value=”0”/>
        </characteristic>
</wap-provisioningdoc>
```

Table 3: HTTP configuration XML: reset client

If the functionality is temporarily reverted to its default behaviour on a device, the device should perform the configuration query each time it is booted up.

If the Service Provider chooses to permanently revert the configured functionality on a device/client to default behaviour, the configuration query performed at start-up shall be disabled. In this case the response shall carry an XML document containing only the version and the validity, both set to -1 as illustrated in Table 4:

```
<?xml version="1.0"?>
<wap-provisioningdoc version="1.1”>
        <characteristic type="VERS">
                <parm name=”version” value=”-1”/>
                <parm name=”validity” value=”-1”/>
        </characteristic>
</wap-provisioningdoc>
```

Table 4: HTTP configuration XML: reset client and stop configuration query

If the SIM is swapped or the device is reset, the device shall again query for configuration settings on each start-up assuming that client-triggered HTTP(S) configuration applies. There shall be no other way for the user to trigger a new configuration query. As described in section 3.1, the configuration client shall also be re-enabled when a SMS message is received requesting a first time configuration.

If the Service Provider chooses to revert the functionality on a device/client to default behaviour (including the disabling configuration query performed at start-up) until there is a User Interface (UI) dependent user action triggering a new query, the response shall carry an XML document containing only the version and the validity, both set to -2 as illustrated in Table 5:

```
<?xml version="1.0"?>
<wap-provisioningdoc version="1.1”>
        <characteristic type="VERS">
              <parm name=”version” value=”-2”/>
              <parm name=”validity” value=”-2”/>
        </characteristic>
</wap-provisioningdoc>
```

Table 5: HTTP configuration XML: reset client until user input and stop configuration query

If the SIM is swapped or the device is reset, the device shall again query for configuration settings on each start-up assuming that client-triggered HTTP(S) configuration applies. As described in section 3.1, it shall also be re-enabled when a SMS message is received requesting a first time configuration.

If the server has available an updated configuration for the client, the server response shall contain a configuration XML document (i.e. *Content-Type* of *text/xml*) that the client shall parse and apply:

- The XML format of this document is based on the syntax used in OMA-CP (see section 4 for the details) with a new parameter to include the version, the validity and the message section.

Server responses that differ from those already described in this section (i.e. an HTTP error) should trigger a device/client to try and retrieve configuration settings the next time the device starts (or the client is started). In scenarios whereby the server response consists of a 403 Forbidden error, the device/client implementation shall also remove the current configuration (i.e. as if it had received a response with both validity and version set to 0).

2.2.3 User Messages

Optionally (that is the tag may not be present), the XML configuration document may be used to convey a user message associated with the result of the configuration server response. The additional XML section is displayed in Table 6:

```
<?xml version="1.0"?>
<wap-provisioningdoc version="1.1”>
        …
        <characteristic type="MSG">
              <parm name=”title” value=”Example”/>
              <parm name=”message” value=”Hello world”/>
              <parm name=”Accept_btn” value=”1”/>
              <parm name=”Reject_btn” value=”0”/>
        </characteristic>
        …
</wap-provisioningdoc>
```

Table 6: HTTP configuration: User notification/message sample

The meaning of the different parameters is described as follows:

- **Title**: The window title where the user message is displayed.

- **Message**: The message that is displayed to the user. Please note the message may contain references to HTTP addresses (websites) that need to be highlighted and converted into links by the device/client.
- **Accept_btn**: This indicates whether an “Accept” button is shown with the message on the device UI. The action associated with the Accept button on the device/client is to clear the message box.
  A value of 1 indicates that an “Accept” button has to be displayed.
  A value of 0 indicates that no “Accept” button has to be displayed.
- **Reject_btn**: This indicates whether the “Decline” button is shown with the message on the device UI. The action associated with the Reject button on the device/client side is to revert the configured services to their defined default behaviour.
  A value of 1 indicates that a “Decline” button has to be displayed.
  A value of 0 indicates that no “Decline” button has to be displayed.
  This parameter is optional, when not provided a default value of 0 shall be assumed.

NOTE: if a Reject_btn is not to be displayed (i.e. the corresponding parameter is set to 0 or is not included), the configuration shall be enabled regardless of whether the user actually presses the “Accept” button.

The *MSG* characteristic (i.e. the user message) is optional and will only be present for the following types of configuration server responses:

1. The response containing the full configuration settings.
2. The response disabling configuration on the device (version and validity are set to 0 or a negative value).

The device should display the message and the relevant/configured buttons in the following configuration server response scenarios:

- After receiving the full configuration settings, only if:
    - Working configuration was previously unavailable, including an unavailable working configuration following a SIM change; or
    - Following a terminal reset
- After receiving the disabling configuration response.

The device/client shall send language/locale settings to the server to set the language/locale of the user message. The client should therefore include the HTTP *Accept-Language* header in all the requests and set the value of this header consistent with the device locale.

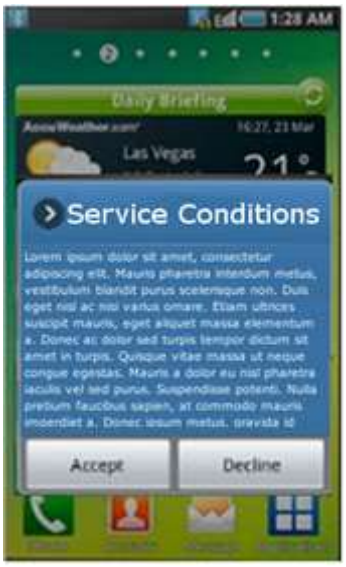

Figure 3: Autoconfiguration server notification example

2.2.4 Use Case Overview

Although previously introduced, this section summarizes the different use cases to indicate the corresponding device behaviour for each scenario:

1. First detection: This is the first time a user makes use of a device. If the process is successful the device receives the correct configuration XML including the validity period of associated configuration parameters. If the device has no issues (i.e. the device receives no errors) during the registration process, the device refrains from contacting the server again until the validity period has expired. As mentioned previously, this process could require several retries to be attempted until the provisioning in IMS is successfully performed.
   Please note that for those devices not having successfully completed the configuration process yet, any Service specific UX available on the device should follow that service's default behaviour (i.e. vanilla behaviour) until a valid configuration is successfully received and processed.
2. Version checking, no changes: If the validity period has expired, or the client has been instructed to retry the configuration process, the device sends a request to verify that it has the correct configuration. If the device already has the latest version, the client receives an XML configuration document containing only the same version as the one that was provided by the client already with the validity period reset to a value specified by the configuration server. This indicates that the configuration the device/client currently has is correct and, as a result, the validity period is renewed as indicated by the updated validity parameter value provided as part of the configuration server response.
3. Version checking, new version available: If the server has a new version of a subset of the fixed configuration parameters (for example the registration Internet Protocol (IP) address) or if the user has requested a reconfiguration through their Service Provider's Customer Care, the device/client receives a new configuration XML the next time the device/client verifies its version
4. Validation process is not OK: If either the device/client or the subscriber is barred from accessing one or more service, the device will either receive an XML with the configuration version and validity attributes set to 0 or a document providing the

configuration only for those services that would be allowed reverting to default behaviour for the others.

Consequently, the device/client must remove the existing configuration and revert to vanilla behaviour (that is any Service-specific UX on the device/client provides only the default behaviour or is disabled).

5. SIM change: If the SIM changes, the previous working configuration should be backed up by the device/client and the device/client should behave as if no configuration is available (that is first-time configuration) and, follow the process described in 2.2.1. Please note that if a working configuration backup associated with the new SIM available on the device/client exists, the validity period should be checked and, if it is still valid, the backup working configuration should be used instead of the device issuing a new configuration request.
6. User with different devices. If a user uses multiple devices, the same configuration shall be valid for all their devices. The described process shall ensure that the device the user is currently using has the latest version.
7. User asks Customer Care to disable (i.e., opt out of) (some) services. In this case the user will be un-provisioned from the network elements providing those services, and when the application asks for a reconfiguration it will, depending on the status of other services, either receive a XML configuration document with the version and validity set to 0 or a document that disables those services while configuring the others. The service shall remain disabled until the user requests Customer Care to provision their device (i.e. to opt in) for the service again. As a result of disabling a service, the capable device/client shall remove the currently working configuration for that service and disable the Service-specific UX (that is reverting to vanilla behaviour).
8. User changes settings that potentially affects service delivery. If the user changes a setting that is relayed to the network as part of the (service-specific) HTTP GET parameters and client-triggered HTTP(S) configuration applies, this shall trigger a new configuration query with the new value of those parameters.

NOTE: All scenarios described above comply with one of the following behaviours of the application on the device:

- First time device/client utilization: if the device/client does not have the correct configuration (version 0 or it is unable to successfully complete the registration process), the device will send a request at each boot sequence (or when the client is restarted) if client-triggered HTTP(S) configuration applies.
- If the configuration server returned a HTTP 511 NETWORK AUTHENTICATION REQUIRED error response on the first time configuration request, the client shall start the SMS based configuration flow as if it were using non-3GPP access (see section 2.3.2).
- The HTTP(S) configuration or re-configuration is triggered as described in section 3
- If the device/client has received the proper configuration, then it shall not request for a new version unless:
    - The validity period has expired, or,
    - It is not able to enable a configured service using the provided configuration

In these cases, the device/client shall immediately request for a new version and not wait until the next reboot/restart.

- If the response received from the configuration server by the client/device is 503 Retry-After, the device/client shall retry the request after the time specified in the "Retry-After" header included as part of the configuration server response.
    - If any other error occurs (for example being unable to resolve the URL or getting an error from the configuration server) the device/client shall retry the procedure during the next time reboot sequence;
    - In the particular case of a client/device receiving a 403 Forbidden, the existing configuration should be removed from the device/client.
- In other error cases (e.g. a 500 Internal Error is issued by the configuration server or the configuration server is unreachable), if a valid configuration is available then, the device/client should keep using it, even if the configuration has expired.
- The following is applicable to both 403 Forbidden and other configuration server error responses:
    - To include scenarios whereby a device migrates to a network without support for this mechanism, the maximum number of unsuccessful consecutive configuration retries allowed by a device (including unsuccessful Domain Name System (DNS) lookup queries) shall be set to 5.
    - If configuration errors persist, the default behaviour for the services to be configured is provided by the client/device and the configuration sequence performed during the boot sequence is disabled.
    - If the SIM is changed or the device is reset, the device should again query for configuration settings on every boot sequence if the client-triggered HTTP(S) configuration applies.

Table 7 enumerates all possible configuration server response codes (including error cases):

Response	Use case	Client behaviour
200 OK	Initial HTTP request response	The client sends the HTTPS request including the cookie
503 Retry after	The server is processing the request/provision	Retry after the time specified in the "Retry-After" header
200 OK + XML with full configuration	New or updated configuration sent to the device	Process the configuration, try to register and if successful, do not try reconfiguration until the validity period is expired or SIM is changed
200 OK + XML with version and validity period only	No update needed	Retry only after validity period or SIM change

200 OK + XML with version and validity period only and both set to 0	Customer or device are not valid or the customer has been deprovisioned from the services to be configured	Retry only after next restart or SIM change If a configuration was available, it shall be removed from the client.
200 OK + XML with version and validity period only and both set to -1	Customer or device are not valid or the customer has been deprovisioned from the services to be configured	The client shall no longer retry autoconfiguration until SIM is changed or a factory reset performed. If a configuration was available, it shall be removed from the client.
500 Internal Server error (or any other HTTP error except 403)	Internal error during configuration/provisioning	Retry on next reboot/the next time the client starts
401 Unauthorized with WWW-Authenticate header containing realm with 3GPP-bootstrapping indication	The configuration server instructs the client to use HTTP digest authentication based on a bootstrapped security association	The client invokes the digest authentication with the configuration server as defined in section 2.4. If no bootstrapped security association exists the bootstrapping procedure is invoked first.
401 Unauthorized in result of a configuration request using a bootstrapped security association	The configuration server requests the renegotiation of the bootstrapped security association.	The client renegotiates the bootstrapped security association with the procedure defined in section 2.4.2.2. A new configuration request shall be sent afterwards using the new bootstrapped security association.
403 Forbidden	Invalid request (e.g. missing parameters, wrong format)	The configuration is removed in the device and version is set to 0. Retry on next reboot, the next time the client starts
409 Conflict	A duplicate value was provided for the friendly_device_name	The user should be asked to provide another value for the friendly_device_name parameter and the configuration request should be retried including the new value NOTE: this return code is only applicable to the friendly_device_name as that is the only parameter controlled by the user that could generate a conflict
511 Network Authentication Required	Network-based authentication is not possible (e.g. in case of non-PS access or security enhanced configuration mechanism over PS access).	Client starts non-PS configuration flow as defined in 2.3 including the cookie if provided.
The configuration server is unreachable	configuration server missing or down	Retry on next reboot, the next time the client starts

Table 7: Summary of autoconfiguration responses and scenarios

2.2.5 Security considerations

For terminals carrying the SIM associated to the user's main identity the connection is carried out over the PS access network, therefore the current design reduces the risk of a man-in-the-middle attack whereby a third party is able to impersonate the configuration server.

To secure interoperability between Service Providers and to reduce complexity on the device/client, the HTTP configuration server shall make use of public root certificates issued by a recognized Certification Authority (CA), that is the root certificates are similar to those used by standard webservers which are widely recognized by browsers and web-runtime implementations both in PCs and devices.

To address security concerns due to mobile application system vulnerabilities (e.g. provisioning of malicious applications that appear to the Configuration Server as "trusted" applications), the security enhanced configuration mechanism could be implemented. In that case, the procedures to resolve the user's MSISDN (that is RADIUS requests, header enrichment and so on) shall be used only for acquiring the user's MSISDN and not for verifying the user's identity. Specifically, an HTTP 511 NETWORK AUTHENTICATION REQUIRED response shall be generated by the configuration server that contains a cookie as part of the response to the initial HTTP request (Set-Cookie header). The client shall then initiate the SMS based configuration mechanism (see section 2.3.2) without requesting the user to provide its MSISDN. The configuration server shall expect the client to provide that cookie in the subsequent HTTPS request (in the Cookie header).

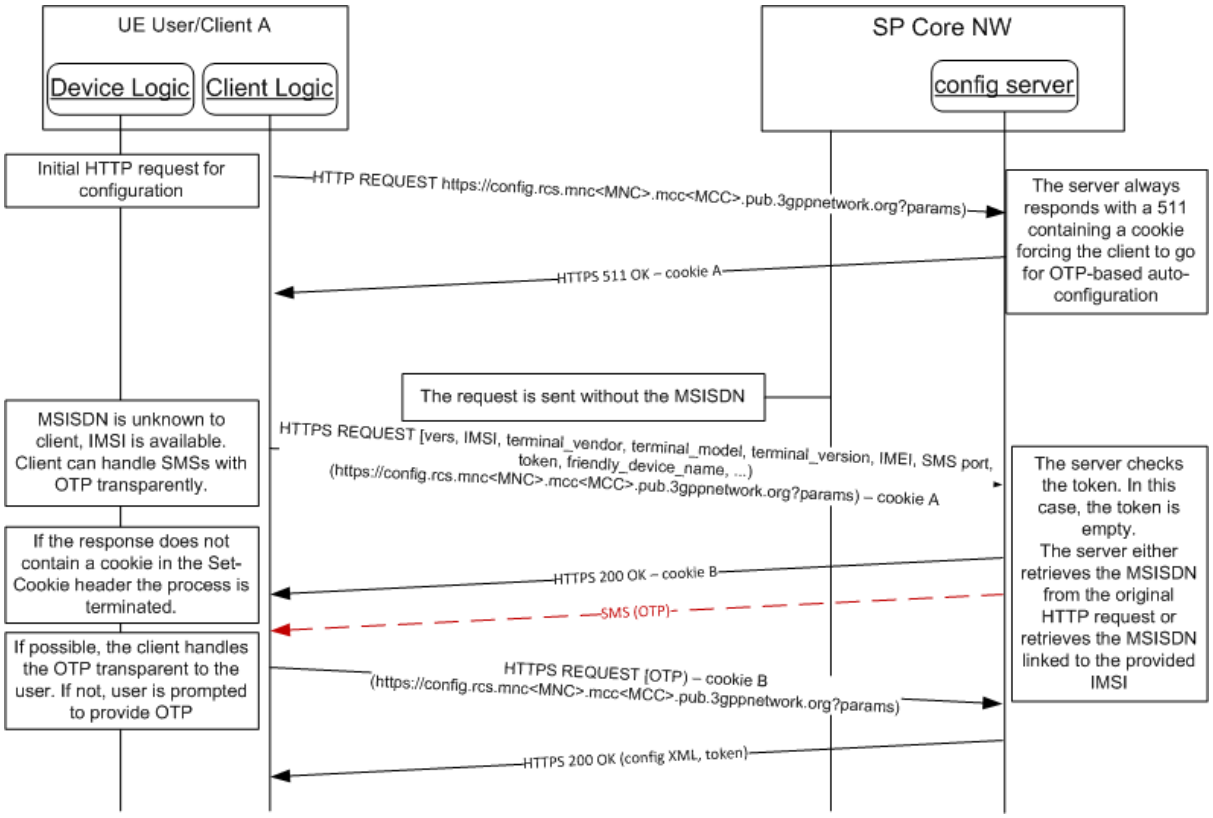

Figure 4: HTTP Configuration: Security enhanced

NOTE: The Service provider shall be able to select between the standard and security enhanced configuration mechanism. It is up to the Service Provider

policy to select the most appropriate configuration mechanisms for particular configuration requests.

2.3 HTTP(S) based client configuration mechanism over non-3GPP access

One of the main limitations of the HTTP configuration mechanism described in section 2.2 is that it only can take place over PS access, as header enrichment is required to identify the subscriber. As an alternative, based on the mechanism presented in section 2.5 to configure additional devices based on an initial SMS exchange, the current section introduces the process to get a primary device configured when 3GPP PS access is not available to the client.

Finally, note that this mechanism shall only be used when it is not possible to perform the configuration over a PS connection.

2.3.1 Overview

Depending on the specific solution, the client may be able to identify that it is not possible to perform the configuration over PS access (e.g. because currently only Wi-Fi connections are available). In that case the client can obtain the configuration by following the procedures in section 2.3.2. For clients that are not aware of the connectivity section 2.3.1.1 provides a specific procedure that can be used for the case where the client can guarantee that any cellular connection in the path to the service provider's configuration server is terminated locally.

2.3.1.1 Clients not able to identify bearer of configuration request

There is a specific case where the solution is not able to identify whether or not configuration is done over cellular access. In these circumstances a solution that is able to ensure that any cellular connection in the path towards the configuration server is terminated on the device itself (e.g. if Wi-Fi is used it is not tethered to a cellular PS connection), shall perform a first request for configuration using the standard HTTP configuration mechanism described in section 2.2:

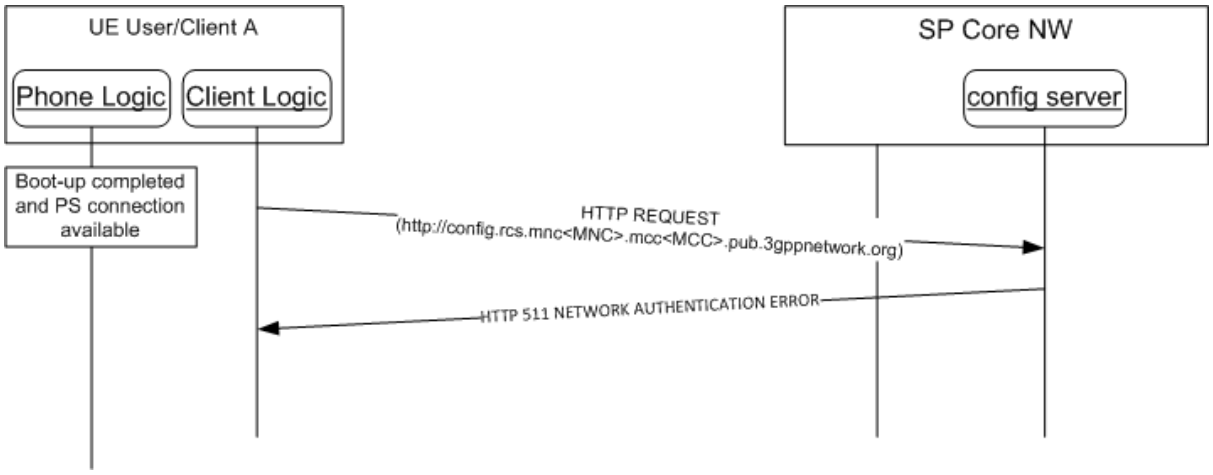

Figure 5: HTTP configuration mechanism: Failed request due to missing header enrichment

NOTE1: The use of another device's PS connection (e.g. a Wi-Fi to cellular-PS-router) may lead to an incorrect identification of the requesting device.

Therefore this request can only be sent reliably by clients that can be aware that any PS connection in the path towards the configuration server is provided by themselves.

NOTE2: Most clients connected over Wi-Fi will not be able to verify that there is no cellular connection used further down the path towards the configuration server and should therefore start immediately with a HTTPS request as described in section 2.3.2.

When this initial request is performed over a non-PS access network, the configuration server is unable to successfully identify/verify the identity of the requester (i.e. RADIUS or header enrichment is no longer an option). In this case, the configuration server shall reply with an HTTP 511 NETWORK AUTHENTICATION REQUIRED error response and the client should continue with the procedure described in section 2.3.2. Otherwise the procedure in section 2.2 shall be followed.

2.3.2 Non-cellular configuration

When performing the configuration over non-cellular access (either because the access is known to be non-cellular or as a result of the procedure in section 2.3.1.1), the client shall follow the SMS based configuration mechanism as detailed below:

If the MSISDN is unavailable (e.g. a previous configuration procedure has not occurred wherein a client is able to identify the MSISDN e.g. because it has been included or as part of the Session Initiation Protocol (SIP) Uniform Resource Identifier (URI) provided in the received XML configuration document) two situations exist:

1. The IMSI is not accessible or the client cannot handle SMS messages with a One-Time Password (OTP) in the background as described in section 2.3.3. In this case, the client shall prompt the user to provide a MSISDN (in E.164 format) for the current device unless a valid cookie is provided from a previous configuration server response.
   The device performs an HTTPS configuration request in the same manner as described in section 2.2.1 (Table 1), plus three additional parameters (i.e. MSISDN, SMS_port and token).
2. The IMSI is available and the client can handle SMS messages with the OTP in the background. In this case, the client shall not prompt the user to provide a MSISDN, and perform a HTTPS configuration request with just two additional parameters (i.e. SMS_port and token).

Parameter	Description	Mandatory	Format
vers	This is either -3, -2, -1, 0 or a positive integer. 0 indicates that the configuration must be updated (e.g. the configuration is damaged, non-existent or an update is needed following a SIM change). A positive value indicates the version of the static parameters (those which are not user dependent) so the server can evaluate whether an update is required. -1 indicates that the device/client is providing the default behaviour for the services that would be configured and has disabled the autoconfiguration query performed at boot. This may be used by the client/device to inform the SP that the functionality was permanently disabled from the device. -2 Indicates that for the services to be configured the default behaviour needs to be provided (including the disabling of the configuration query at boot), but a configuration query might be triggered on user action.	Y	Int (-3, -2, -1, 0 or a positive integer)
IMSI (International Mobile Subscriber Identity)	If available, the subscriber's IMSI shall be sent as a parameter.	N if the OS platform allows it, it shall be included	String (15 digits)
provisioning_version	String that identifies the version of this provisioning specification supported by the client. It shall be set to "*1.0*" (without the quotes) for clients following this specification.	Y	String (4 max), Case-Sensitive
terminal_vendor	String that identifies the device OEM.	Y	String (4 max), Case-Sensitive
terminal_model	String that identifies the device model.	Y	String (10 max), Case-Sensitive
terminal_sw_version	String that identifies the device software version.	Y	String (10 max), Case-Sensitive
IMEI	If available, the subscriber's IMEI shall be sent as a parameter. Those Service Providers that support a comprehensive device database can ignore the terminal_X parameters and use the IMEI instead, if it was available to the implementation.	N if the OS platform allows it, it shall be included	String (15 digits)

msisdn	MSISDN, in E.164 format, of the primary SIM which is used to derive the user's main identity.	N, it is only mandatory if the IMSI is not provided	E.164 (+447900000001) in international format NOTE: In case that msisdn comes with a plus sign, the client shall provide the msisdn value with the plus sign encoded as per [RFC3986] section 2.1.
SMS_port	This parameter sets the User Data Header (UDH) port that has to be used for the SMS that is to be employed to validate the requester through an OTP. If set to 0, the client indicates the server that the SMS UDH procedures are not supported either by the client or the platform, so a standard SMS (user visible) shall be used instead. If not set, the default port value used shall be 37273.	N	Int (0-65355)
token	If this is the first time the device is being configured (or the validity of the token is expired), this should be an empty string. If not, the token obtained in the initial configuration process shall be reused.	Y	String
friendly_device_name	If provided by the user, a user friendly identification for the device may be passed along that can be used by the network when presenting the user with an overview of their devices. NOTE: this parameter needs to be included only if required for one of the services to be configured. In which case its mandatory character will be documented in the relevant service specific documents	N, only to be provided if provided by the user	String (30 max before escaping), Case-Sensitive

Table 8: HTTP configuration for primary devices over non-PS access: HTTPS request GET parameters

3. At this point the configuration server is able to identify whether this is a first time request:

   a) If the token value is empty (i.e. first time to configure over a non-PS access networkwith no previous configuration over 3GPP access, or if the previously retrieved token has expired), the request is identified by the configuration server as a

first time configuration. In this case, and provided the network allows configuring devices using this mechanism, the configuration server responds with an HTTP 200 OK response that includes a new cookie (Set-Cookie header) to be used in the subsequent HTTP(S) requests.

i. Following the request, an SMS message shall be sent to the primary device, i.e. the device using the SIM associated with the MSISDN or IMSI sent in the HTTP request. This SMS message will contain an OTP. The format of this SMS is covered in detail in section 2.3.3.

NOTE: the configuration server provider may implement mechanisms on the server to protect it from suspicious or potentially malicious transactions (e.g. a client causing too many SMS messages)

ii. In parallel, if OTP handling that is transparent to the user is not possible, the device performing the HTTP configuration prompts the user for the OTP. Therefore, the user should manually enter the value that was received via SMS.

NOTE: to handle scenarios wherein it is not possible for the network to send an SMS message in the format of section 2.3.3 to the device, a client should always permit the user to enter the OTP, potentially after some initial delay

iii. The device performing the HTTP configuration makes a second HTTPS request using the following parameters in the GET request:

Parameter	Description	Mandatory	Format
OTP	This is the password received on the primary device using the SIM associated with the provided MSISDN/IMSI	Y	String

Table 9: HTTP configuration for primary devices: Second and final HTTPS request GET parameters

NOTE: the second HTTPS request shall include the cookie obtained in step 3 (cookie header) so that the configuration server is able to correlate the initial and subsequent HTTPS requests.

From this point the procedure is identical to the one described in sections 2.2.2 and 2.2.3. If the response includes a full XML configuration document however, the generated token is added as a parameter to the configuration server's 200 OK response.

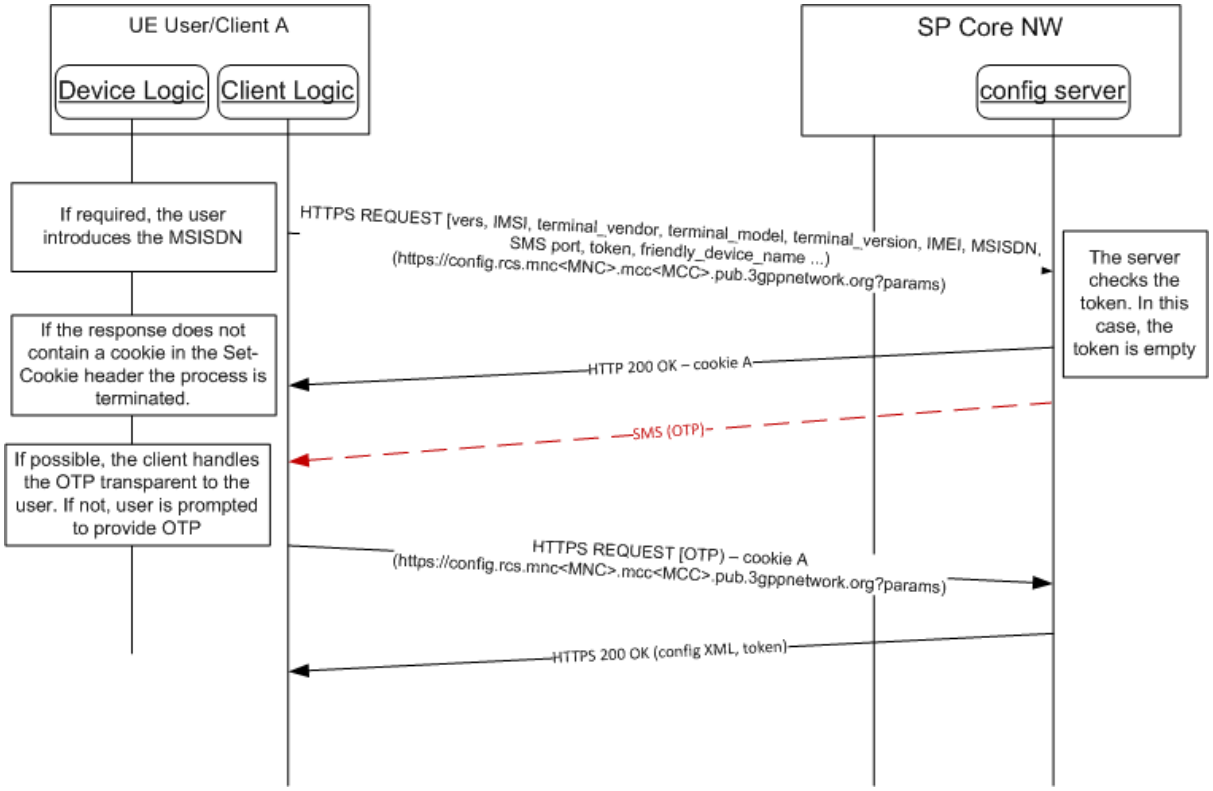

Figure 6: HTTP configuration for primary devices over non-PS access with MSISDN: empty token

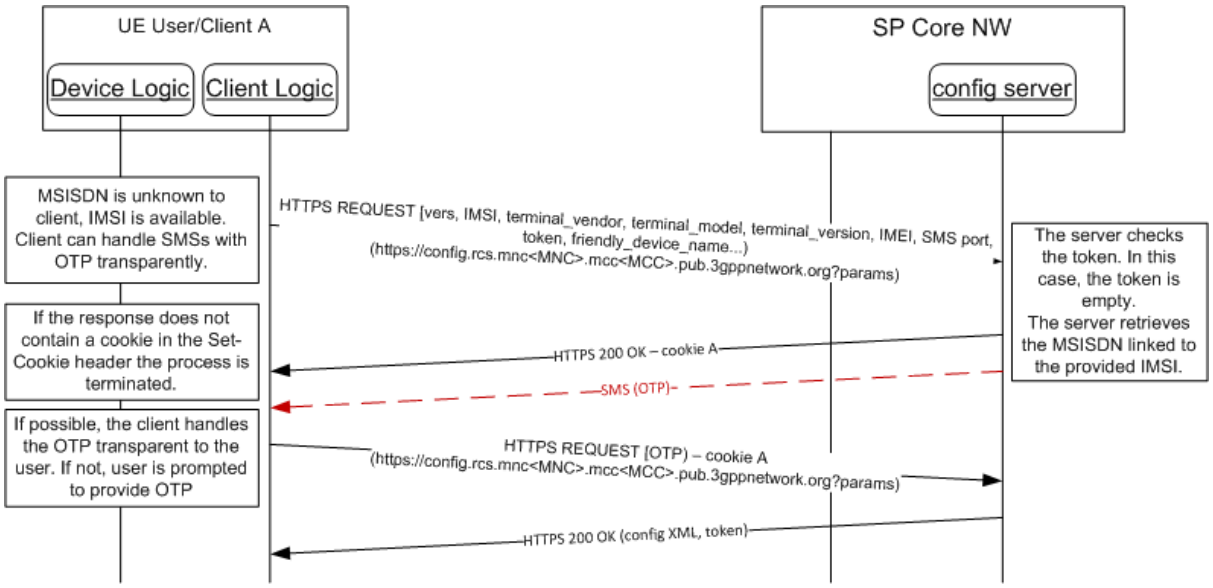

Figure 7: HTTP configuration for primary devices over non-PS access with only IMSI: empty token

b) If the token is valid (i.e. non-empty, and successfully verified by the configuration server), then, from this point the procedure is identical to the one described in sections 2.2.2 and 2.2.3. If the response includes a full XML configuration document however, the generated token is added as a parameter to the configuration server's 200 OK response.

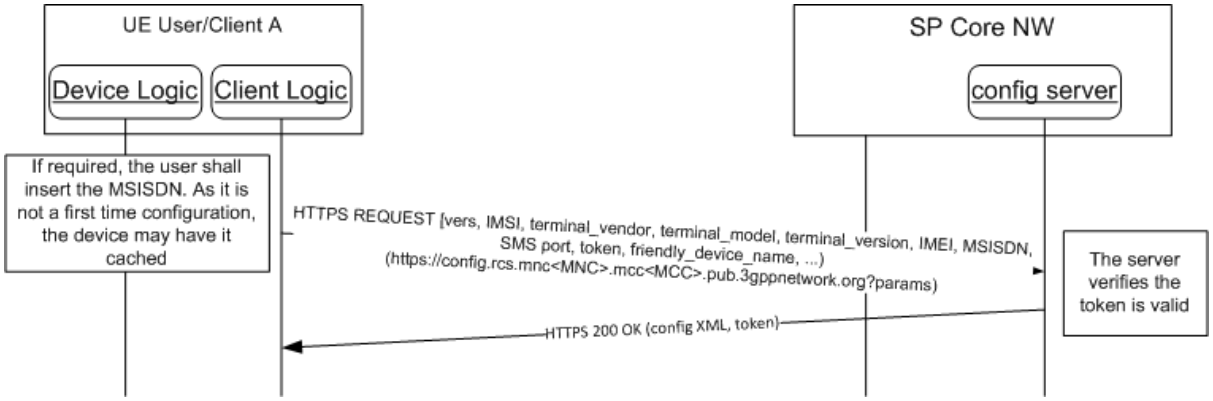

Figure 8: HTTP configuration for primary devices over non-PS access: Valid token

In order to incorporate the token in the configuration server responses, a new characteristic, TOKEN, is provided at the same level at the VERS characteristic so it may be provided in all other kinds of Configuration server responses (e.g. empty configuration XML [with and without a message] and with a full configuration XML) and includes a validity for the token specified in seconds:

```
<?xml version="1.0"?>
<wap-provisioningdoc version="1.1">
        <characteristic type="VERS">
                <parm name=”version” value=”X”/>
                <parm name=”validity” value=”Y”/>
        </characteristic>
        <characteristic type="TOKEN">
                <parm name=”token” value=”Z”/>
                <parm name=”validity” value=”W”/>
        </characteristic>
        -- Rest of the XML if applicable
</wap-provisioningdoc>
```

Table 10: HTTPS configuration XML: Token characteristic

In both cases (i.e. bullets a) and b) above), the token and MSISDN shall be stored on the device. Therefore it is not necessary to re-execute the entire procedure for future requests. These values shall be removed together with the rest of the configuration when the device or client is reset.

2.3.3 SMS format to receive the OTP value

In case of a primary device configuration, the device receiving the SMS containing the OTP password shall match the device where the client is running, the preferred approach is that the SMS is sent in a format that allows the client to intercept the OTP in a transparent manner. In order to do so, the configuration server shall perform the following steps:

1. If the value for the SMS_port parameter included in the HTTPS request sent by the device after receiving a HTTP 511 error response is a positive integer in the range between 1 and 65535 and the configuration server supports the UDH handling procedure (as per [3GPP TS 23.040]) to send a SMS to a specific port, then the following SMS format convention shall be used:
    - DataCodingScheme = 08 (UCS2)
    - UserDataHeader = 06 05 04 4074 0000

UDHL length fields=06 05 04,
Destination port: port provided by the client in HTTP request encoded in hex. If not provided, 37273 (0x9199) shall be the default value.
Source Port: 0000 (0 in decimal)

- Content of the message shall be the OTP encoded in the same format the Service Provider uses to transmit user readable SMS messages.

With this convention, an SMS sent to the device shall be routed to an application listening for SMS on the port indicated by the client and shall be handled transparently to the user.

2. If SMS_port is set to 0, the UDH procedures are not supported either by the client or the platform/OS the client runs on. Consequently, the server shall send a standard SMS and the user shall be prompted by the client to manually provide the OTP code to the client (e.g. via a text box).

Where the Service Provider wants to send a standard SMS for the OTP code, the SMS_port parameter shall be included in the HTTPS 200 OK response sent by the configuration server just before the SMS that carries the OTP code (see HTTP flows presented in figures 9 and 10). The response shall carry an XML document containing the SMS_port parameter set to 0 as illustrated in Table 11:

```
<?xml version="1.0"?>
<wap-provisioningdoc version="1.1">
    <characteristic type="POLICY">
        <parm name="SMS_port" value="0"/>
    </characteristic>
</wap-provisioningdoc>
```

Table 11: HTTPS configuration XML: SMS_port zero policy

In this case the OTP handling that is transparent to the user is not possible and the client prompts the user to enter the OTP which is received via SMS.

NOTE: The Service Provider should allow enough time prior to sending the SMS with the OTP code so as to make sure that the client has received the HTTPS 200 OK response that carries the XML with the SMS_port parameter set to zero. This response shall be sent always considering the provisioning_version parameter and thus ensuring backward compatibility. In case that the Service Provider sets a different value to the SMS_port parameter, this value shall be ignored by the client.

2.3.4 Use cases review

The error conditions and use case scenarios covered in section 2.2.4 also apply for configuration over non-PS access, but in this case any disabling of the client shall be limited to that specific non-PS network. Further configuration attempts shall thus be done when the device connects to a cellular or another non-PS network. In addition to those errors, for the process of performing a configuration over non-PS access networks the following specific error conditions shall be taken into account and supported:

1. In the scenario whereby a user has to be prompted to provide an MSISDN, the given MSISDN may be invalid or unauthorized to retrieve the Service configuration. As a result, the initial request shall be answered by the configuration server with an HTTP 403 FORBIDDEN error response. The client shall inform the user of the problem and may offer to retry with a different MSISDN.

   NOTE1: if the MSISDN belongs to a SIM which is not currently available to the capable device, the SMS sent by the configuration server will not be received. Therefore in this scenario, the client should provide a timeout mechanism and prompt the user after the timeout period has been reached, to re-enter a MSISDN.

   NOTE2: the timeout mechanism utilized by the client is not in scope of this specification.

2. In the scenario where only the IMSI is sent to the network, the network may not support this type of configuration. In that case the Configuration Server shall answer to the initial request with a HTTP 403 FORBIDDEN response. The client shall in this case request the user for their MSISDN and perform the procedure including the MSISDN. This is shown in the flow in Figure 9:

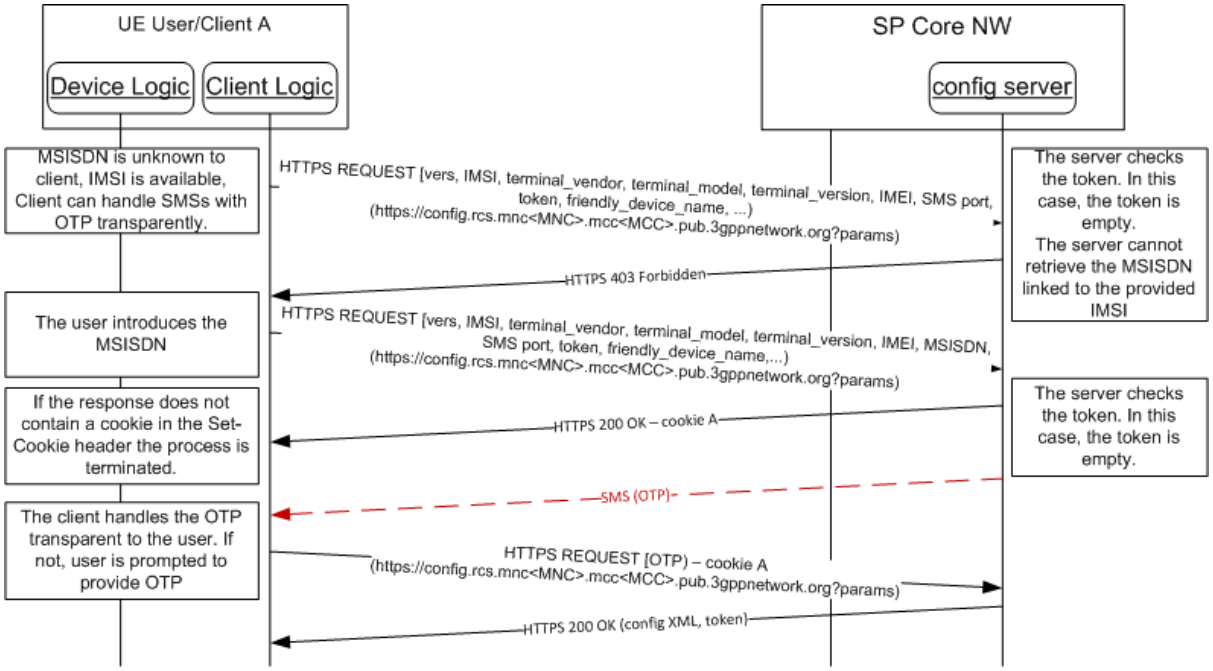

Figure 9: HTTP configuration for primary devices over non-PS access with only IMSI: Not supported by the network

3. The OTP password is invalid. As a result, the configuration server replies with an HTTP 511 NETWORK AUTHENTICATION REQUIRED error response. It is up to the client to provide a user retry mechanism. When retrying, the client shall re-start the configuration process from the beginning.
4. The token is invalid. As a result, the configuration server replies with an HTTP 511 NETWORK AUTHENTICATION REQUIRED error response. It is up to the client to provide a user retry mechanism. When retrying, the client shall re-start the configuration

process from the beginning. Consequently, if a valid token was previously stored, it shall be removed from the device.

2.3.5 Security considerations

The same access security considerations described in section 2.2.5 for the standard HTTP(S) configuration mechanism also apply in this case.

Service Providers may request the client to fall back to the client configuration mechanism over non-3GPP access while requesting configuration in 3GPP access to secure the user identification via header enrichment, as defined in section 2.2.5.

In addition, as a Service Provider Option, the configuration server is able to enforce a policy for the OTP challenge on primary devices being always visible to the user, especially for the case where the client would be able to apply it transparently by use of the SMS UDH procedure. If the client receives a Configuration Response in HTTP 200 OK with a "SMS port zero" policy (see section 2.3.3) then it shall expect the reception of the OTP via user visible SMS. Thus it shall prompt the user to enter the OTP and continue processing with the user input only.

2.4 HTTP(S) based client configuration mechanism with GBA Authentication

2.4.1 Overview

The General Bootstrapping Architecture (GBA) defined in [3GPP TS 33.220] provides mechanisms for AKA based user authentication using the 3GPP Authentication Centre (AuC) and the USIM or ISIM. The HTTP(s) based client configuration mechanism supports the authentication of primary devices via GBA.

The Authentication Procedure consists of two parts. The basis for the user authentication between the device and network applications is a bootstrapped security association. The association provides the client with a Bootstrapping Transaction Identifier (B-TID) and key material which can be used by clients of the device to authenticate the user with network applications.

An application client will use the B-TID and the key material for the authentication with a specific network application. For the Service Provider Device Configuration HTTP Digest Authentication is used.

The end-to-end implementation of GBA is defined in [3GPP TS 33.220]. The protocol extension of the client configuration protocol shall be implemented according to [3GPP TS 24.109].

2.4.2 Use Case review

2.4.2.1 HTTP Digest Authentication

Precondition for the use of this procedure for authentication within the HTTP(s) based client configuration is the support of GBA as defined in [3GPP TS 33.220] on the device and in the Service Provider network.

The GBA Authentication can be applied independent from the access network type (3GPP or non-3GPP). However, a client supporting GBA based HTTP digest authentication shall invoke the client configuration mechanisms in accordance with the access network type, i.e. in 3GPP access as defined in section 2.2, in non-3GPP access as defined in section 2.3.

When sending the secured configuration request the client supporting GBA shall indicate it by the addition of a GBA product token in the User-Agent header as defined in [3GPP TS 24.109].

If the Service Provider's configuration server does not support GBA based HTTP digest authentication it returns responses as defined in sections 2.2 and 2.3 respectively. Device configuration commences as defined at the same place.

If the Service Provider's configuration server supports GBA based authentication then it returns an HTTP 401 Authorization Required response with a WWW-Authenticate header instructing the client to use HTTP digest Authentication with a bootstrapped security association.

If the client has no bootstrapped security association in place it shall invoke the bootstrapping procedure defined in section 2.4.2.2 to generate it.

If the client has a bootstrapped security association in place it shall use the stored key material and the B-TID to generate keys specific to the configuration server as defined in [3GPP TS 33.220]. With the key material and the B-TID it shall generate the Authorization header to be sent in a new secured request for client configuration.

The configuration server will fetch the B-TID and key material from the Bootstrapping Server Function (BSF) and complete the digest authentication. If successful, a 200 OK response containing an Authentication-Info header and the configuration XML will be returned to the client. The configuration server may request the client to renegotiate the bootstrapped security association (e.g. due to expiry) by returning a 401 "Unauthorized" response. When the client receives the 401 "Unauthorized" response it shall renegotiate the bootstrapped security association with the procedure defined in section 2.4.2.2.

The client shall validate the Authentication-Info header information and apply the configuration XML.

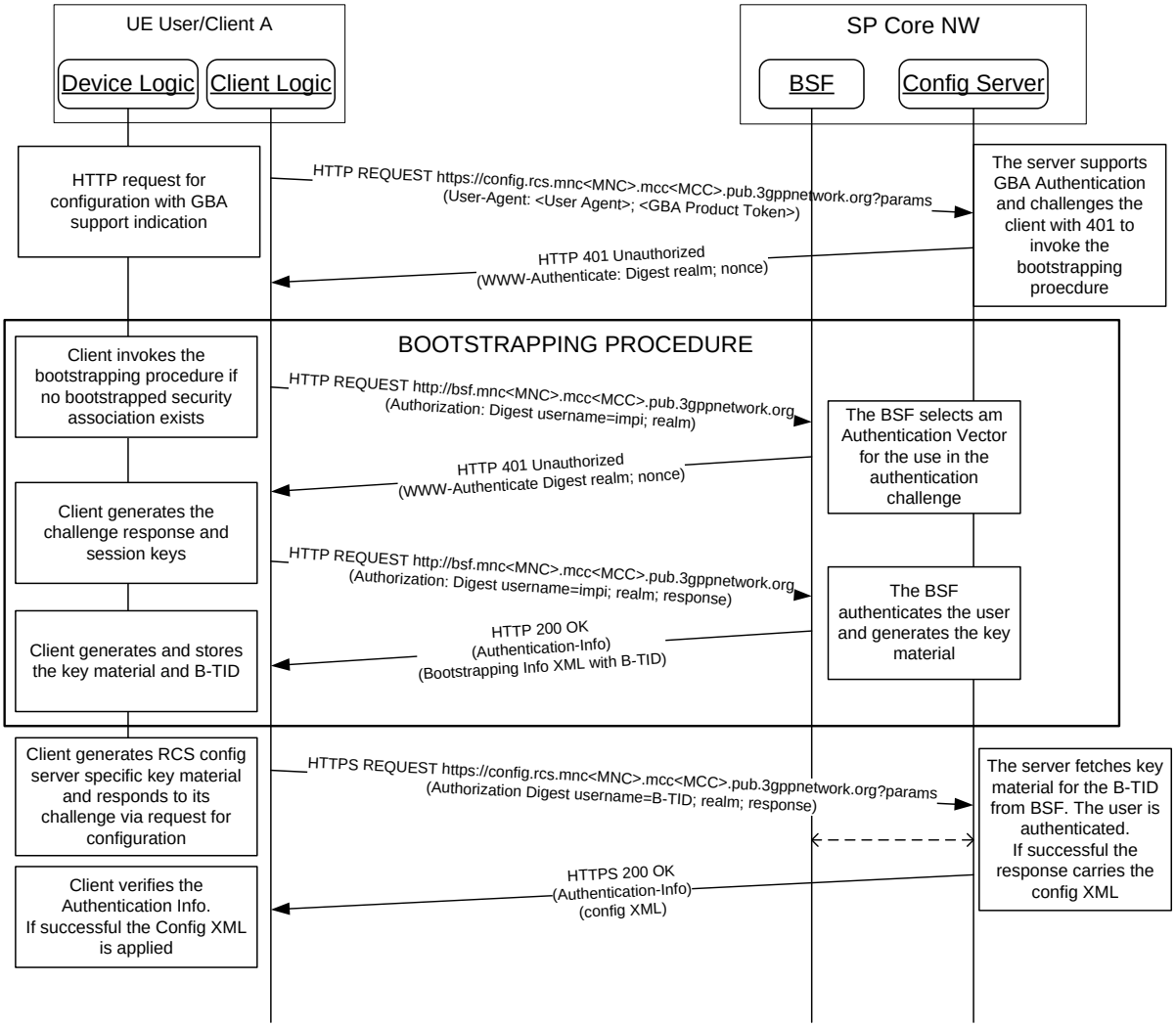

Figure 10: HTTP configuration for primary devices using GBA

2.4.2.2 Bootstrapping Procedure

The device will invoke the bootstrapping procedure with the service provider's BSF to generate a bootstrapped security association as defined in [3GPP TS 24.109]. This section provides an informative overview of the procedure.

The URI of the BSF is derived by the client from the IMSI or the user's private identity (IMPI) as defined in [3GPP TS 23.003]. The client creates an HTTP GET request with an authorization header as defined in [3GPP TS 24.109] and sends it to the BSF. The authorization header contains the user's private identity (IMPI) as username. If the device is not able to get the user's private identity (IMPI) from the SIM, it shall be constructed from the IMSI as defined in [3GPP TS 23.003].

On receipt of the request for authentication the BSF retrieves and selects an Authentication Vector for use in the authorisation challenge [3GPP TS 33.220]. It returns a HTTP 401 Authorization Required response to the client with a WWW-Authenticate header instructing the client to authenticate itself.

The client runs the AKA algorithm [RFC3310] to calculate the challenge response which is sent back to the BSF in the Authorization header of a subsequent HTTP GET request. It also calculates the session keys.

On reception of the GET request the BSF authenticates the user based on the challenge response received from the client. It generates the B-TID for the IMPI and stores the session keys. It informs the client about the success of the authentication in the Authentication-Info header of the 200 OK response. The response contains also the bootstrapping XML with the B-TID. The client generates the key material and stores it for subsequent authorisations.

2.5 Configuration of additional devices sharing the same identity

This section describes the process of autoconfiguration authentication for the scenario in which the SIM associated with the identity to be used for the services requiring configuration is not inserted in the device being provisioned.

2.5.1 First-time configuration

During first-time configuration, the device implementation/client will receive the credentials associated with the primary SIM card of the user regardless of the type of connection they are using (e.g. Wi-Fi, PS) to reach the Configuration server.

The process is as follows:

1. As an option, the device implementation/client will offer the possibility to the user to perform manual provisioning
2. The user is prompted for the MSISDN or SIP URI of the primary device and the Service Provider associated with the primary SIM. The account created is always associated with this primary identity that the user has to input into the application. Please note that, as a pre-condition, the aforementioned identity must already be provisioned using the mechanism described in previous sections.
3. The device performs the HTTPS configuration as presented in section 2.2.1, however, using the following GET parameters instead of the default ones:

Parameter	Description	Mandatory	Format
vers	This is either -3, -2, -1, 0 or a positive integer. 0 indicates that the configuration must be updated (e.g. the configuration is damaged, non-existent or an update is needed following a SIM change). A positive value indicates the version of the static parameters (those which are not user dependent) so the server can evaluate whether an update is required. -1 indicates that the device/client is providing the default behaviour for the services that would be configured and has disabled the autoconfiguration query performed at boot. This may be used by the client/device to inform the SP that the functionality was permanently disabled from the device. -2 Indicates that for the services to be configured the default behaviour needs to be provided (including the disabling of the configuration query at boot), but a configuration query might be triggered on user action.	Y	Int (-3, -2, -1, 0 or a positive integer)
msisdn	MSISDN, in E.164 format, of the primary SIM which is used to derive the identity.	N, Mandatory if sip_uri not provided	E.164 (+447900000001) in international format NOTE: In case that msisdn comes with a plus sign, the client shall provide the msisdn value with the plus sign encoded as per [RFC3986] section 2.1.
sip_uri	SIP URI of the primary device	N, Mandatory if msisdn is not provided	String (50 max), Case-insensitive
provisioning_version	String that identifies the version of this provisioning specification supported by the client. It shall be set to "*1.0*" (without the quotes) for clients following this specification.	Y	String (4 max), Case-Sensitive

token	If this is the first time the additional device is being configured (or the validity of the token is expired), this should be an empty string. If not, the token obtained in the initial configuration process shall be reused here.	Y	String (24 max), Case-Sensitive
device_type	This indicates the type of device where the client is running.	Y	Possible values: - Tablet - PC - Other
friendly_device_name	If provided by the user, a user friendly identification for the device may be passed along that can be used by the network when presenting the user with an overview of their devices. NOTE: this parameter needs to be included only if required for one of the services to be configured. In which case its mandatory character will be documented in the relevant service specific documents.	N, only to be provided if provided by the user	String (30 max before escaping), Case-Sensitive

Table 12: HTTP configuration for additional devices: Initial HTTPS request GET parameters

Please note that the initial HTTP request is not required in this case since the header enrichment requirement is not applicable. Therefore, the device implementation/client will directly perform the HTTPS request as presented in Figure 11.

4. As this is a first time request, the token value is empty; the request is then identified as a first time configuration. In this case, and provided the network allows for configuring additional devices using this mechanism, the HTTP server responds with a HTTP 200 OK response carrying a new cookie (Set-Cookie header) to be used in the subsequent HTTP requests.

   a) Following the request, an SMS message shall be sent to the primary device, i.e. the phone carrying the SIM associated to the MSISDN the user introduced in step 2. This SMS message will contain an OTP. This message shall be a standard SMS (i.e. no UDH procedures required).

   NOTE: When used on IP Multimedia Subsystem (IMS) networks with IMS devices, other ways may be provided to contact the primary device. Those are covered in [PRD-RCC.15]

   b) In parallel, the device performing the HTTP configuration prompts for the OTP. Therefore, the user should manually introduce the code delivered via SMS to the primary device.
   c) Once the user enters the OTP, the device performing the HTTP configuration makes a second HTTPS request using the following parameters in the GET request:

Parameter	Description	Mandatory	Format
OTP	This is the password received on the device carrying the SIM associated with the MSISDN introduced in step 2	Y	String (8 Max), Case-Sensitive

Table 13: HTTP configuration for additional devices: Second and final HTTPS request GET parameters

Please note this second HTTPS request shall carry the cookie obtained in step 4 (cookie header) therefore the HTTP configuration server can correlate the initial and final HTTPS requests.

d) From this point onwards the procedure is identical to the one described in sections 2.2.2 and 2.2.3, however, with the token added as a parameter. If the request is successful, one of the possible 200 OK responses described in section 2.2.2 is provided.

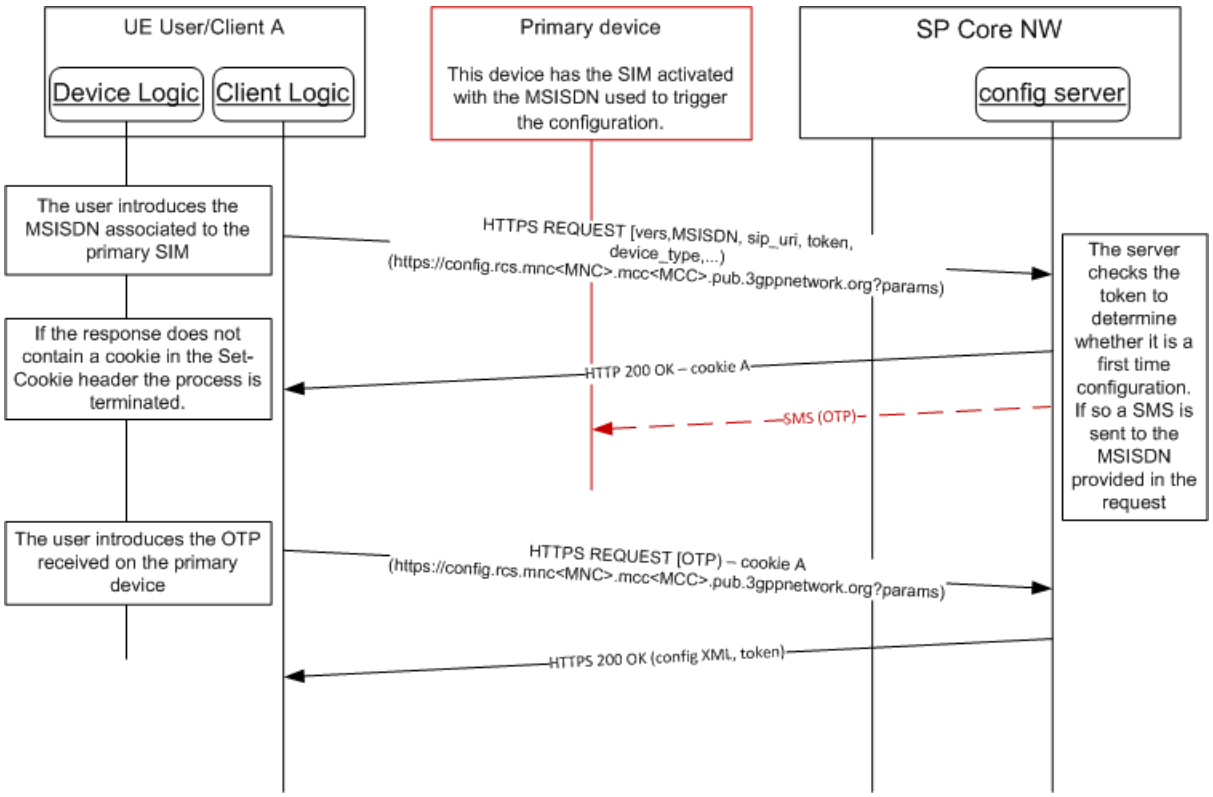

Figure 11: HTTP configuration for additional devices: First time configuration

Please note the token shall be stored with the MSISDN so it is not necessary to repeat this procedure for future requests. These values shall be removed together with the rest of the configuration when the device or client is reset.

2.5.2 Error handling

In the process of performing a first time configuration for additional devices, there are three possible error conditions that the client has to be aware of and handle:

1. The MSISDN used is not valid or it is not authorized (including the case the primary MSISDN is not been provisioned yet to use the services being configured) to get the

configuration/make use of the services. In this case, the initial request will be answered with an HTTP 403 FORBIDDEN error and the client shall inform the user of the issue and may offer to retry with a different MSISDN.

2. The OTP password introduced by the user is not valid. In this case, the HTTP configuration server replies again with a HTTP 511 NETWORK AUTHENTICATION REQUIRED error. It is up to the client implementation to offer the user to retry. If retrying, the client shall start the first time configuration process from the beginning.
3. The HTTP server suffers an internal error (HTTP 5XX [except 511], response coming from the server). In this case, the user shall be informed of the circumstance and offered to retry. If retrying, the client shall start the first time configuration process from the beginning.

2.5.3 Subsequent configuration attempts and life cycle

If the client has access to the token and the MSISDN used for the first time configuration, has a value, the initial request is performed

1. An initial request like in the case of the first time configuration of additional devices is made, this time including the token parameter set to the value received on the previous successful configuration attempt
2. If successful, from this point onwards the procedure is identical to the one described in sections 2.2.2 and 2.2.3, however, with the token added as a parameter. If the request is successful, one of the possible 200 OK responses described in section 2.2.2 is provided.

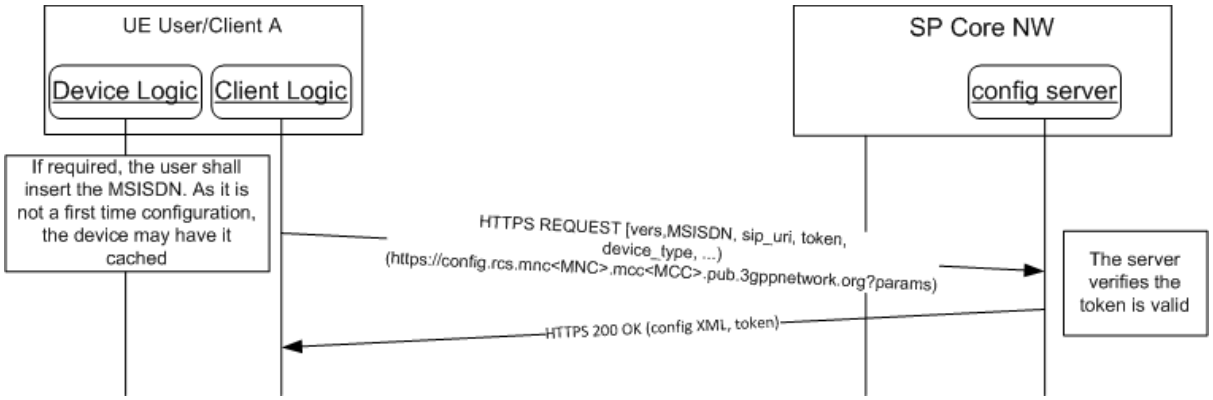

Figure 12: HTTP configuration for additional devices: Subsequent attempts

If the token and/or the MSISDN are not available (for example the device is reset), then the client shall start a first time configuration as described in section 2.5.1.

Please note the received token shall be stored with the MSISDN so it is not necessary to repeat this procedure for future requests. These values shall be removed together with the rest of the configuration when the device or client is reset.

2.5.4 Error handling

In the process of performing a subsequent configuration for additional devices, there are three possible error conditions that the client has to be aware of and to handle:

1. The MSISDN used is not valid or it is not authorized to get the configuration/make use of the services to be configured. In this case, the initial request will be answered with an

HTTP 403 FORBIDDEN error and the client shall inform the user of the issue and may offer to retry with a different MSISDN.

2. The token is no longer valid. In this case, the HTTP configuration server replies again with a HTTP 511 NETWORK AUTHENTICATION REQUIRED error. From this moment, the process is equivalent to the first time configuration process after the same error is received.
3. The HTTP server suffers an internal error (HTTP 5XX response coming from the server). In this case, the user shall be informed of the circumstance and offered to retry. If retrying, the client shall start the subsequent configuration attempt procedure from the beginning.

2.5.5 Use cases review

From the use cases presented in section 2.2.4, only the following scenarios apply to the configuration of additional devices sharing the same identity:

1. First detection
2. Version checking
3. Validation process is not OK
4. User asks Customer Care to disable a service

2.5.6 Security considerations

The same security considerations described in section 2.2.5 for the standard HTTP(S) configuration mechanism also apply in this case.

2.6 Configuration of non-Cellular devices with a dedicated identity

To configure clients on devices that do not carry a SIM, but have to function with a dedicated own identity the following generic solution is provided:

1. The user obtains an OTP through means that are out of the scope of this specification (e.g. from an operator website after authentication, delivered together with the device, obtained through an operator's retail outlet, etc.)
2. The user is prompted for the E.164 address or SIP URI to be used by the device and their Service Provider. The account created is always associated with this primary identity that the user has to input into the application.
3. The device performs the HTTPS configuration as presented in section 2.2.1, however, using the following GET parameters instead of the default ones:

Parameter	Description	Mandatory	Format

vers	This is either -3, -2, -1, 0 or a positive integer. 0 indicates that the configuration must be updated (e.g. the configuration is damaged, non-existent or an update is needed following a SIM change). A positive value indicates the version of the static parameters (those which are not user dependent) so the server can evaluate whether an update is required. -1 indicates that the device/client is providing the default behaviour for the services that would be configured and has disabled the autoconfiguration query performed at boot. This may be used by the client/device to inform the SP that the functionality was permanently disabled from the device. -2 Indicates that for the services to be configured the default behaviour needs to be provided (including the disabling of the configuration query at boot), but a configuration query might be triggered on user action.	Y	Int (-3, -2, -1, 0 or a positive integer)
msisdn	E.164 format of the provided identity	N, Mandatory if sip_uri is not provided	E.164 (+447900000001) in international format NOTE: In case that msisdn comes with a plus sign, the client shall provide the msisdn value with the plus sign encoded as per [RFC3986] section 2.1.
sip_uri	SIP URI of the device	N, Mandatory if msisdn is not provided	String (50 max), Case-insensitive
provisioning_version	String that identifies the version of this provisioning specification supported by the client. It shall be set to "*1.0*" (without the quotes) for clients following this specification.	Y	String (4 max), Case-Sensitive

token	If this is the first time the primary device is being configured (or the validity of the token is expired), this should be an empty string. If not, the token obtained in the initial configuration process shall be reused here.	Y	String (24 max), Case-Sensitive
device_type	This indicates the type of device where the client is running.	Y	Possible values: - Tablet - PC - Other
OTP	This is the password provided to the user in step 1. Set to an empty string in case a non-empty token is provided	Y	String (8 Max), Case-Sensitive
friendly_device_name	If provided by the user, a user friendly identification for the device may be passed along that can be used by the network when presenting the user with an overview of their devices. NOTE: this parameter needs to be included only if required for one of the services to be configured. In which case its mandatory character will be documented in the relevant service specific documents.	N, only to be provided if provided by the user	String (30 max before escaping), Case-Sensitive

Table 14: HTTP configuration for non-cellular devices: Initial HTTPS request GET parameters

Please note that the initial HTTP request is not required in this case since the header enrichment requirement is not applicable. Therefore, the device implementation/client will directly perform the HTTPS request as presented in Figure 13.

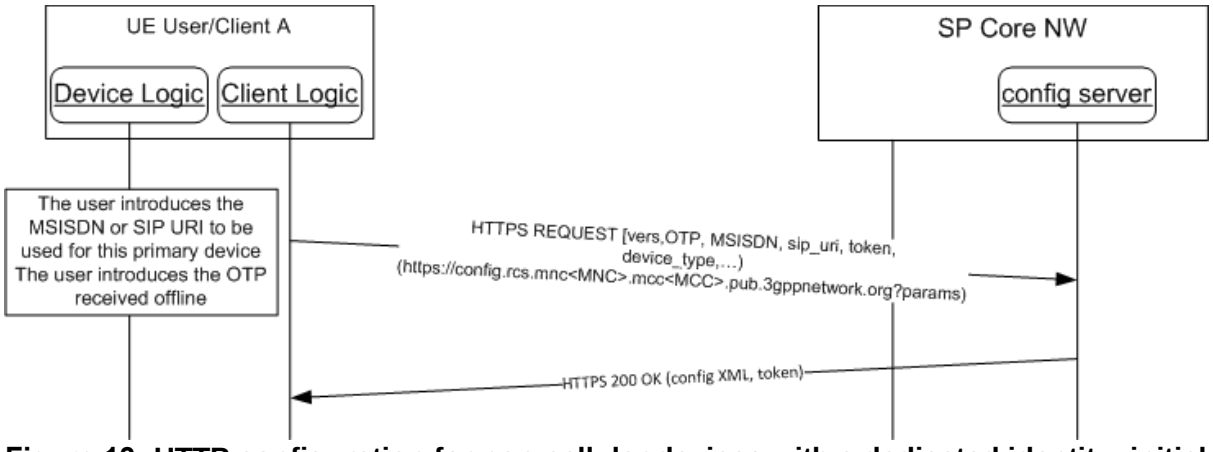

Figure 13: HTTP configuration for non-cellular devices with a dedicated identity: initial request

4. As this is a first time request, the token value is empty; the request is then identified as a first time configuration. In this case, and provided the network allows for configuring

devices using this mechanism, the HTTP server responds with a HTTP 200 OK response carrying a new cookie (Set-Cookie header) to be used in the subsequent HTTP requests

From this point onwards the procedure is identical to the one described in sections 2.2.2 and 2.2.3, however, with the token added as a parameter. If the request is successful, one of the possible 200 OK responses described in section 2.2.2 is provided.

Please note the token shall be stored with the identity so it is not necessary to repeat this procedure for future requests. These values shall be removed together with the rest of the configuration when the device or client is reset.

2.6.1 Subsequent configuration attempts and life cycle

If the client has access to the token and the identity used for the first time configuration, has a value, the initial request is performed

1. An initial request like in the case of the first time configuration of the primary device is made, this time including the token parameter set to the value received on the previous successful configuration attempt
2. If successful, from this point onwards the procedure is identical to the one described in sections 2.2.2 and 2.2.3, however, with the token added as a parameter. If the request is successful, one of the possible 200 OK responses described in section 2.2.2 is provided.

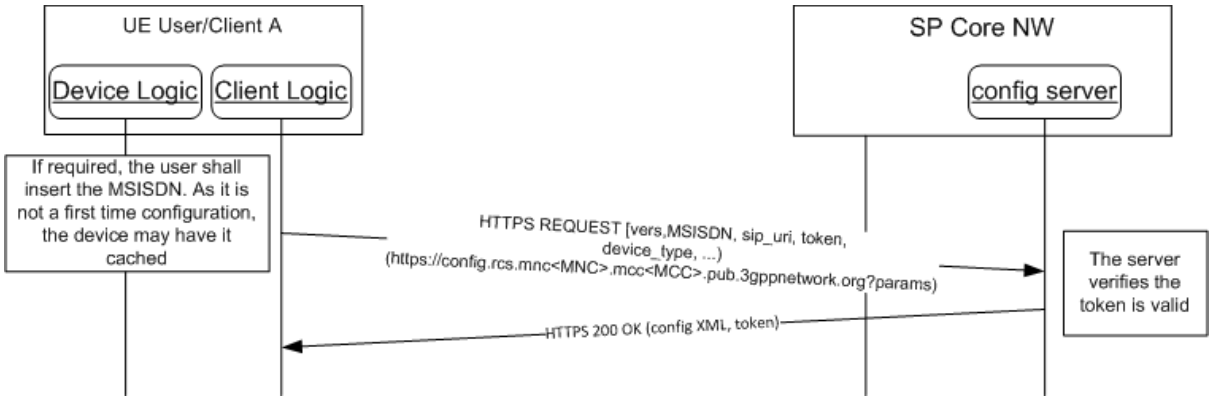

Figure 1: HTTP configuration for non cellular devices: Subsequent attempts

If the token and/or the MSISDN are not available (for example the device is reset), then the client shall start a first time configuration as described in section 2.5.1.

Please note the received token shall be stored with the MSISDN so it is not necessary to repeat this procedure for future requests. These values shall be removed together with the rest of the configuration when the device or client is reset.

2.6.2 Error handling

In the process of performing a subsequent configuration for additional devices, there are three possible error conditions that the client has to be aware of and to handle:

1. The MSISDN used is not valid or it is not authorized to get the configuration/make use of the services to be configured. In this case, the initial request will be answered with an

HTTP 403 FORBIDDEN error and the client shall inform the user of the issue and may offer to retry with a different MSISDN.

2. The token is no longer valid. In this case, the HTTP configuration server replies again with a HTTP 511 NETWORK AUTHENTICATION REQUIRED error. From this moment, the process is equivalent to the first time configuration process after the same error is received.
3. The HTTP server suffers an internal error (HTTP 5XX response coming from the server). In this case, the user shall be informed of the circumstance and offered to retry. If retrying, the client shall start the subsequent configuration attempt procedure from the beginning.

3 Network requested configuration request

There are use cases (e.g. customer support) where forcing a reconfiguration or a first configuration is required.

The present section presents the enhancements that need to be implemented both on the network side and on the client in order to support a network requested reconfiguration.

NOTE: The described mechanisms only cover reconfiguration requests related to primary devices

3.1 First time configuration initiated via SMS

In this option, the mechanism that will trigger the configuration will be a network originated SMS. Please note that this option is only available to platforms and clients that support the application port addressing (UDH header handling as per [3GPP TS 23.040]).

Regarding the SMS format, the following configuration shall be used:

- DataCodingScheme = 08 (UCS2)
- UserDataHeader = 06 05 04 4074 0000

    a) UDHL length fields=06 05 04,
    b) Destination port: 0x9199 (37273 in decimal)
    c) Source Port: 0x0000 (0 in decimal)

- The SMS content shall be the IMSI associated to the SIM plus the word "rcscfg"[3] preceded by the dash symbol i.e. ‘-‘. For example: If the IMSI is *214011001388741*,
  The value in the text shall be
  214011001388741-rcscfg[3]

When the device receives such a request and the IMSI matches the one on the SIM the following actions shall take place:

1. Start the HTTP configuration described in section 2.2 or 2.3 depending on current connectivity

[3] Includes rcs for historic reasons

2. If already active, deactivate service(s) for which the settings have changed as specified for those services.

3.2 Reconfiguration initiated via SMS

In this first option, the mechanism that will trigger the reconfiguration will be a network originated SMS. Please note that this option is only available to platforms and clients that support the application port addressing (UDH header handling as per [3GPP TS 23.040]).

Regarding the SMS format, the following configuration shall be used:

- DataCodingScheme = 08 (UCS2)
- UserDataHeader = 06 05 04 4074 0000

    a) UDHL length fields=06 05 04,
    a) Destination port: 0x9199 (37273 in decimal)
    b) Source Port: 0x0000 (0 in decimal)

- For devices on which no IMS-based service is enabled, the SMS content shall be the IMSI associated to the SIM plus the word "rcscfg"[4] preceded by the dash symbol i.e. '-'. For example: If the IMSI is
  214011001388741,
  The value in the text shall be
  214011001388741-rcscfg[3]
- Otherwise the SMS content shall be the IMS Private User Identity (Private_User_Identity parameter in the XML configuration) plus the word "rcscfg"[3] preceded by the dash symbol i.e. '-'. For example: If the private identity is
  214011001388741@ims.mnc001.mcc214.3gppnetwork.org,
  The value in the text shall be
  214011001388741@ims.mnc001.mcc214.3gppnetwork.org-rcscfg

When the device receives such a request and the IMPI matches the one in the existing configuration, it shall take the following actions:

1. Perform a HTTP configuration (as per section 2.2 or 2.3 depending on current connectivity) setting the version and validity parameters to 0 (i.e. like in the case of a first-time configuration), so it is guaranteed a complete configuration XML is provided by the HTTP configuration server.
2. If active, deactivate service(s) for which the settings have changed as specified for those services and apply configuration.
3. After the configuration process is completed, the client shall activate the services using the received settings.

3.3 Interaction with the user during the network initiated reconfiguration

When performing a network initiated reconfiguration and if the user is making use of the service, he shall be notified that the process is taking place and that consequently the service will not be available until the reconfiguration is completed.

[4] Includes rcs for historic reasons

4 Configuration document formatting

4.1 Configuration Data

The actual configuration data will be represented in an OMA-CP (see [OMA CP Cont]) like XML structure as described in section 4.2. The use of this mechanism for the configuration of a service therefore requires a definition of how its configuration parameters will be represented in this structure. This may be done through the definition of a mapping from existing Management Objects to this XML structure or by including a definition for such representation in newly defined Management Objects specific to the service.

NOTE: An example is provided in [PRD-RCC.15] providing a generic configuration for IMS based services.

4.2 HTTP configuration XML structure

In addition to the parameters and characteristics type provided by the mapping or definition of Management Objects as presented in the previous section, it is necessary to define the following mandatory configuration XML elements[5] where the different applications and their parameters are thus to be specified in service specific documents:

```
<?xml version="1.0"?>
<wap-provisioningdoc version="1.1">
        <characteristic type="VERS">
                <parm name=”version” value=”1”/>
                <parm name=”validity” value=”Z”/>
        </characteristic>
        <characteristic type="TOKEN">
                <parm name=”token” value=”X”/>
                <parm name=”validity” value=”Y”/>
        </characteristic>
        <characteristic type="MSG">                    -- This section is OPTIONAL
                <parm name=”title” value=”Example”/>
                <parm name=”message” value=”Hello world”/>
                <parm name=”Accept_btn” value=”X”/>
                <parm name=”Reject_btn” value=”X”/>
        </characteristic>
        <characteristic type="APPLICATION">
                <parm name=”AppID” value=”apU”/>
                <parm name=”Name” value=”V”/>
                <parm name=”AppRef” value=”W”/>
                        …
        </characteristic>
        <characteristic type="APPLICATION"> -- This is an example
                <parm name=”AppID” value=”apR”/>
                <parm name=”Name” value=”S”/>
                <parm name=”AppRef” value=”T”/>
                <characteristic type=”O”>
                        <parm name=”P” value=”Q”/>
                        …
                </characteristic>
```

[5] Please note the AppID’s used in the example are provided as example only as they have not been reserved.

```
            <characteristic type="M">
                …
            </characteristic>
            …
        </characteristic>
</wap-provisioningdoc>
```

Table 15: Complete RCS HTTP configuration XML structure

4.2.1 Configuration storage on the client

The service configuration (including the token described in sections 2.3.2 and_2.5.1) should be stored securely on the device and should not be accessible to the user.

If any of the required parameters for a service are not configured or configured with an unexpected value, that service functionality should revert to default behaviour and not be presented as such to the user. This default behaviour needs to be defined for the individual services and might for example be to disable the service and its entry points. In this state, the full service functionality can only be restored by completing the first-time configuration procedure (see section 2).

If a configured device is reset, the client should securely back up the configuration in the device together with the associated IMSI prior to the reset. Please note that this also applies in the event of swapping SIM cards. The configuration associated with the old SIM should then be securely backed up before triggering a first time registration.

The motivation behind the configuration backup is to facilitate the scenario where following a reset or after a SIM swap, the original SIM card is re-introduced into the device. In that instance instead of triggering a first-time configuration, the configuration is restored.

In those terminals where, as a consequence of the processes mentioned in the previous paragraphs (reset, SIM card swap), the terminal also deletes the contacts (for example a particular Service Provider is enforcing a policy where a SIM swap causes the deletion of the contacts), the associated Service information (e.g. cached capabilities per contact) should also be removed. In this case, the Service information associated with the contacts is not backed up.

The number of configuration backups stored is left to the device's implementation, but shall be at least 2 (for the currently inserted and a previous SIM).

Clients functioning as secondary clients sharing the SIM identity used in a user's main device (see section 2.5) may offer multiple users the possibility to access the services (e.g. by requiring selecting which user to serve when started). In that case the client shall store a user's configuration and service data (e.g. Chat histories and call logs) in local storage when switching to another account. All private data shall not be accessible to other users. No new configuration requests shall be performed for the stored accounts even when the validity of their configuration expires. A new configuration request shall only be done when a stored configuration is restored because a user has selected to use the account again. If no valid token is available anymore, this may result in a complete first time configuration procedure. When the first time configuration request is successful the stored service data for that user will be made available again.

Annex A Document Management

A.1 Document History

Version	Date	Brief Description of Change	Approval Authority	Editor / Company
1.0	02 February 2015	initial version split of from RCC.07 v5.0 to allow for more generic use	PSMC	Tom Van Pelt / GSMA
2.0	28 February 2015	Token characteristic in response over 3GPP, clarifications in response codes and description of renegotiation of bootstrapped security association	PSMC	Tom Van Pelt / GSMA

A.2 Other Information

Type	Description
Document Owner	GSMA Network2020 Programme IP Communications Global Specification Group
Editor / Company	Tom Van Pelt / GSMA

It is our intention to provide a quality product for your use. If you find any errors or omissions, please contact us with your comments. You may notify us at prd@gsma.com

Your comments or suggestions & questions are always welcome.