Entitlements Protocol Specification
Use Case: Tokyo and Multi-SIM w/GSMA Integration

Version 1.2.5
March 31th, 2017

Table of Content

Table of Figures

Glossary

Term	Definition
4FF	Fourth generation UICC form factor. Refers to the primary device's removable UICC.
ConfirmOrder	ES2+ function to confirm a previously requested download order
DownloadOrder	ES2+ function to instruct the SM-DP+ server of a new profile request
EID	Embedded UICC Identifier
ES2+ Interface	Interface used to order Profile Package preparation for specific eUICCs and delivery of the Profile Package.
eUICC	Embedded Universal Integrated Circuit Card
GSMA	GSM Association
ICCID	Integrated Circuit Card Identifier
MatchingId	Unique identifier of the context of a specific management order generated during Download Initiation procedure (ConfirmOrder)
OneNumber Pairing	Refers to the network process of associating the Primary device's (i.e. iPhone) phone number to the Tokyo device. MT calls to Primary device phone number are forked to both devices and MO calls use the same caller-id, Primary's caller-id.
Primary Device	Refers to the device who owns the phone number in the multi-sim scenario (e.g. iPhone)
RSP	Remote SIM Provisioning
Secondary Device	Refers to companion device sharing the Primary device's phone number (e.g. Tokyo device)
SM-DP+ Server	Enhanced Subscriber Manager Data Preparation Server
URL	Uniform Resource Locator
USIM	Universal Subscriber Identity Module

Overview

This is an update to the Entitlement protocol specification document to support Tokyo and Multi-SIM capability. This builds up on the Arctic Entitlements Specification 1.9.

This document describes related Entitlements changes to the protocol required to support Tokyo and Multi-SIM use case. For simplification purposes, this document will only list deltas to the Arctic Entitlements Specification document v1.9.

Multi-SIM Assumptions

Multi-SIM is used to refer to the ability of the network to map a phone number, usually the iPhone's, to multiple ICCIDs.

All devices multi-SIM paired shared a single phone number, usually the iPhone's.

Any underlying phone number used by individual devices will not be exposed to the user. The network should block services (i.e. voice calls, sms, etc.) to the native phone number.

Cellular Sign-Up and eSIM profile download architecture diagram

The following diagram depicts the cellular signup and eSIM profile download flow and reference architecture during the Tokyo device cellular provision process. This approach requires integration between the Carrier's Network and the GSMA's SM-DP+ server / ES2+ interface according to standard.

Note: the below diagram is a reference architecture implementation. The actual integration network components between the Websheet server and Entitlements server is Carrier / Vendors implementation specific.

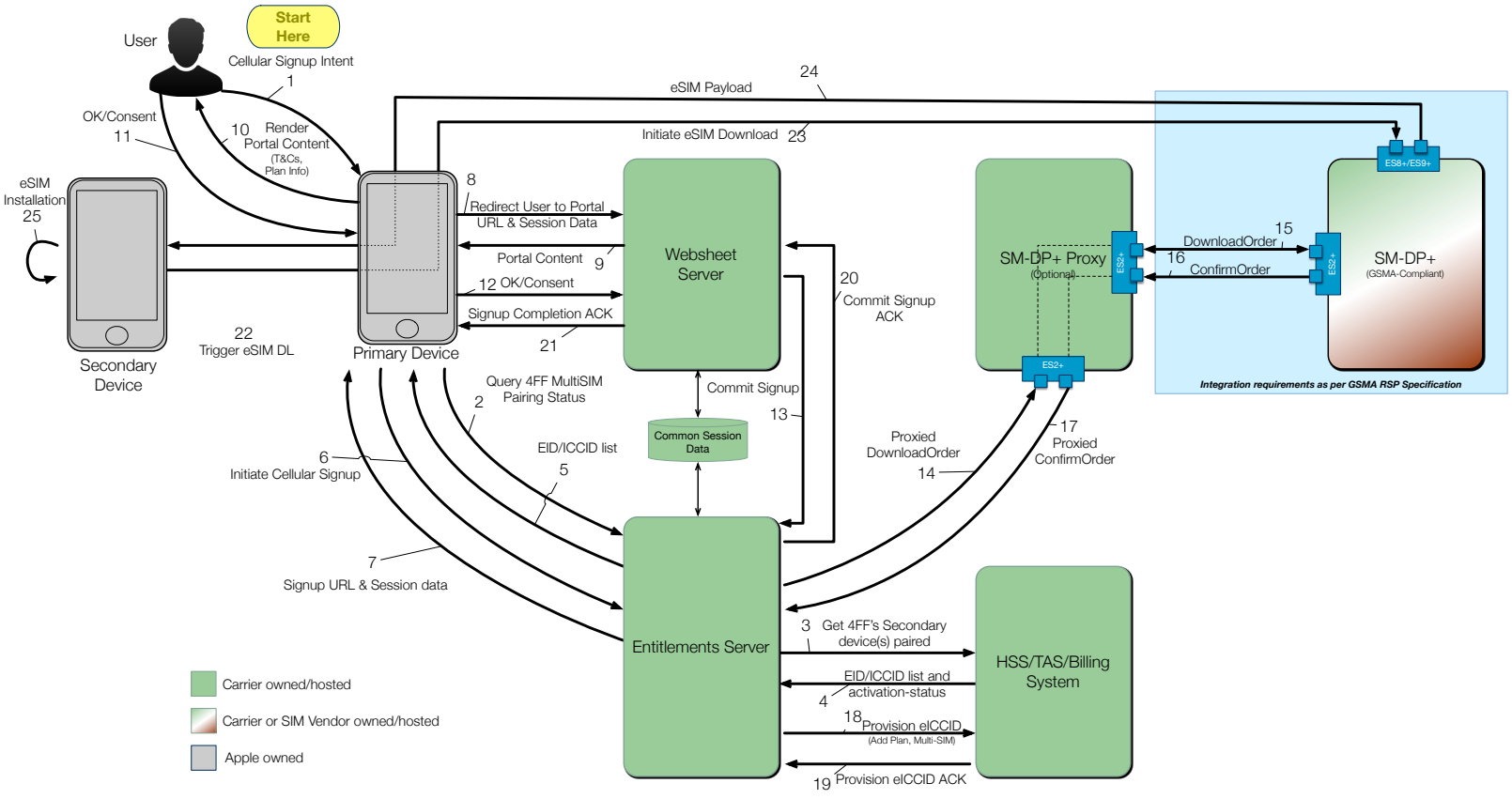

Figure 1 Architecture diagram

Cellular Sign-Up and eSIM profile download call flow

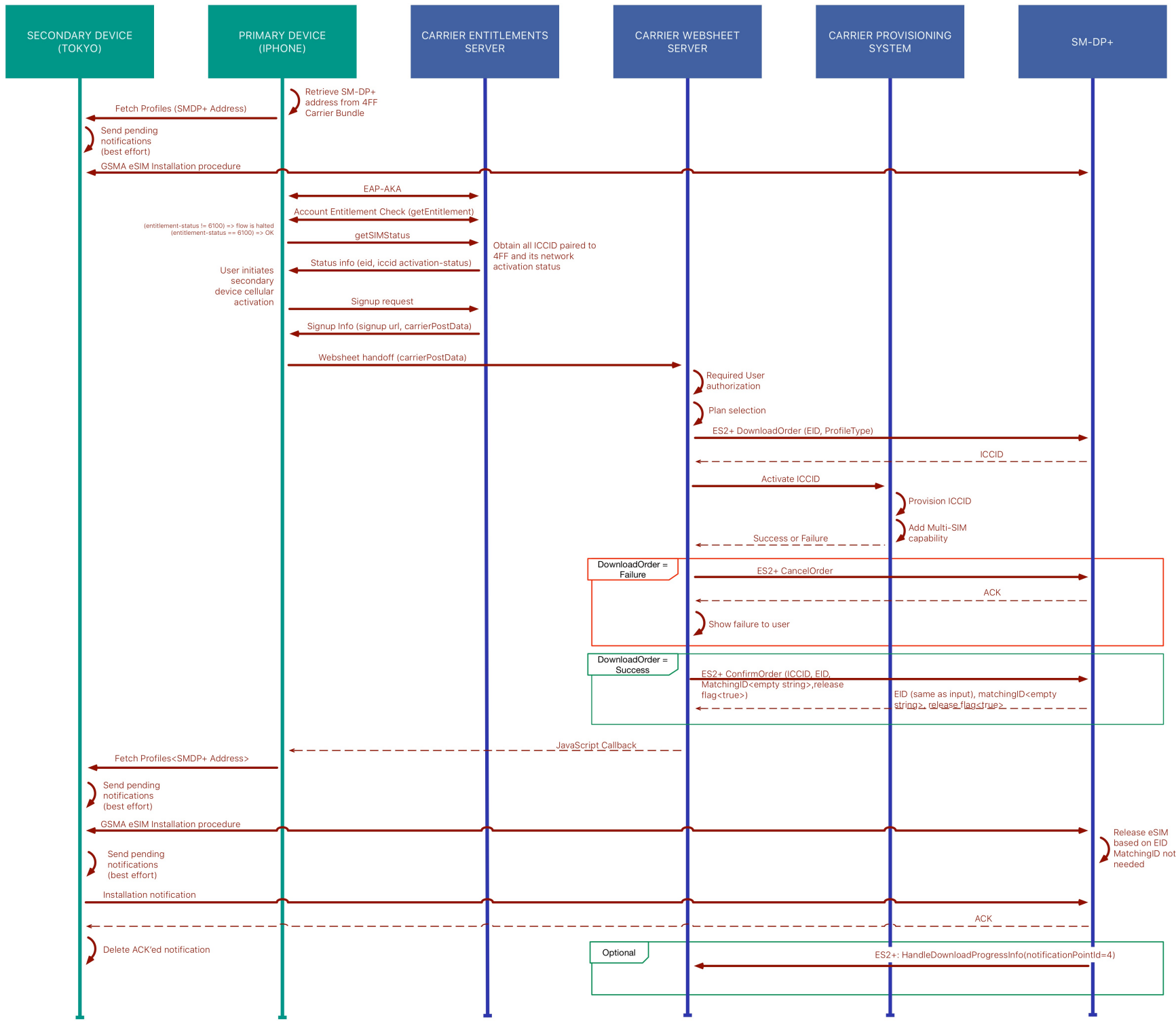

Figure 2 Tokyo Cellular sign-up and eSIM profile download call flow

Entitlements Layer

The following sections discuss the Entitlements layer enhancements required to support Tokyo and Multi-SIM features.

Entitlements Status Codes

Action Level Codes

Code	Constant Name	Description
6000	STATUS_SUCCESS	Action succeeded
6001	STATUS_UNSUPPORTED_ACTION	Action is not recognized
6002	STATUS_DISALLOWED_ACTION	Action is denied by policy
6003	STATUS_UNKNOWN_DEVICE	Device type is not recognized
6004	STATUS_UNKNOWN_SUBSCRIBER	Subscriber is not recognized. Disables Entitlements until device reboot.
6005	STATUS_TEMPORARY_FAILURE	Carrier's network sub-system temporary failure
6006	STATUS_SRVCTOKEN_EXPIRED	service-token expired or it is invalid
6007	STATUS_INVALID_LOCATION	Invalid e911 Address (Not currently used)
6008	STATUS_MAX_DEVICES_REACHED	Maximum number of secondary devices were reached
6009	STATUS_MAX_ICCIDS_REACHED	EID has reached its ICCID limit.

Entitlements Level Codes

Code	Constant Name	Description
6100	STATUS_ENABLED_ENTITLEMENT	Entitlement is enabled for the subscriber
6101	STATUS_DISABLED_ENTITLEMENT	Entitlement is disabled for the subscriber
6102	STATUS_INVALID_ENTITLEMENT	Entitlement is not recognized for the subscriber
6103	STATUS_PROVISIONING_ENTITLEMENT	Entitlement request is still in progress
6104	STATUS_BIZACCT_ENTITLEMENT	Entitlement is not supported for Business Accounts
6105	STATUS_PPACCT_ENTITLEMENT	Entitlement is not supported for Prepaid Accounts
6106	STATUS_INCOMPATIBLE_ENTITLEMENT	Network could not process action due to temporary failure

Entitlements Actions

getAuthentication / postChallenge Actions

Both getAuthentication / postChallenge actions are a fundamental piece to the Entitlements protocol. It enables the network, SIM and device to establish mutual trust between them by means of EAP-AKA authentication procedure. For Tokyo and Multi-SIM, there are not expected changes to the existing getAuthentication / postChallenge actions and therefore the implementation requirements from previous specifications remain the same.

getSIMStatus Action

The device sends this action any time it needs to check the SIM pairing status. In case the account is already sign-up for multi-sim, the Entitlements server should provide a STATUS_SUCCESS status and the list of EID and ICCID paired to the iPhone.

This action is not subject to periodic polling. Events that trigger this action includes but are not limited to: each time the user initiate a sign-up flow, every time the user opens the device's eSIM-related UI, user launches account management websheet, through APNS (multi-sim-signup-changed)

Request

Data Point	Explanation
request-id required	Type: Integer Identifies the particular action. The request-id is unique within a request.
action-name required	Type: Integer Identifies the name of the Entitlements request action Value(s): "getSIMStatus"
primary-iccid required	Type: String Contains the Primary device ICCID.

Response

Data Point	Explanation
response-id required	Type: Integer Must match the request-id for which this response applies. It must be unique within the response
status required	Type: Integer Value(s): STATUS_SUCCESS (6000) STATUS_DISALLOWED_ACTION (6002) STATUS_TEMPORARY_FAILURE (6005)
manage-account-url required	Type: String This is a Websheet hosted by the carrier similar to the signup URL. After signing up the user can use this URL to manage the subscription post-signup.

	Only HTTPS URLs values are supported.
manage-account-url-post-data required	Type: String Carrier-defined payload that is passed as the POST Request body during the Websheet hand off to the manage-account-url. This can be used for user auto-login into the Websheet and correlate the Entitlements request and Websheet request. The data will be passed in the HTTP POST request body, as a URL-encoded key/value pair in the same way that a browser submits an HTML form. The value of this attribute is assigned to "**carrierPostData**" key. Attribute can be ignored if the account has not signup for Multi-SIM.
secondary-devices-paired optional	Type: Array of Objects (See Secondary Devices Paired object definition) List of Secondary devices currently paired to the Primary device's IMSI/MSISDN/MDN. This attribute can be omitted or empty array which is interpreted as not Multi-SIM pairings to the Primary device's account. The attribute is an array given there could be 1+ secondary devices paired simultaneously. This array enables the Primary and Secondary devices during the decision which ICCID must be selected secondary device.

Secondary Devices Paired object definition

Data Point	Explanation
eid Conditional	Type: String Secondary device's EID currently paired to the Parent device's ICCID/Phone Number. EID, IMEI and MEID are mutually exclusive. Only 1 must be returned in the response.
imei Conditional	Type: String Secondary device's IMEI currently paired to the Parent device's ICCID/Phone Number EID, IMEI and MEID are mutually exclusive. Only 1 must be returned in the response.
meid Conditional	Type: String Secondary device's MEID currently paired to the Parent device's ICCID/Phone Number EID, IMEI and MEID are mutually exclusive. Only 1 must be returned in the response.
iccid required	Type: String Seconrady device's ICCIDs currently paired to the Parent device's ICCID/Phone Number
activation-status required	Type: String Represents the activation status of the eid+iccid pair. Enumerated Values: **New**, **Active**, **In-Progress**, **Inactive, Unusable**

	This status is used to provide user feedback about the service status of the ICCID. It is indicate when ICCID ready for cellular camping (Active)
alt-smdp-fqdn optional	Type: String The Fully Qualified Domain Name of an alternate SMDP+ Server. See Supporting Multiple SMDP+ Vendors section.

Expectations:

The network shall maintain the state for users that have sign-up or not for multi-SIM. Similarly, they shall maintain the list of EIDs and ICCIDs that are multi-SIM paired to the Primary's IMSI/MSISDN/MDN.

ICCID Activation lifecycle

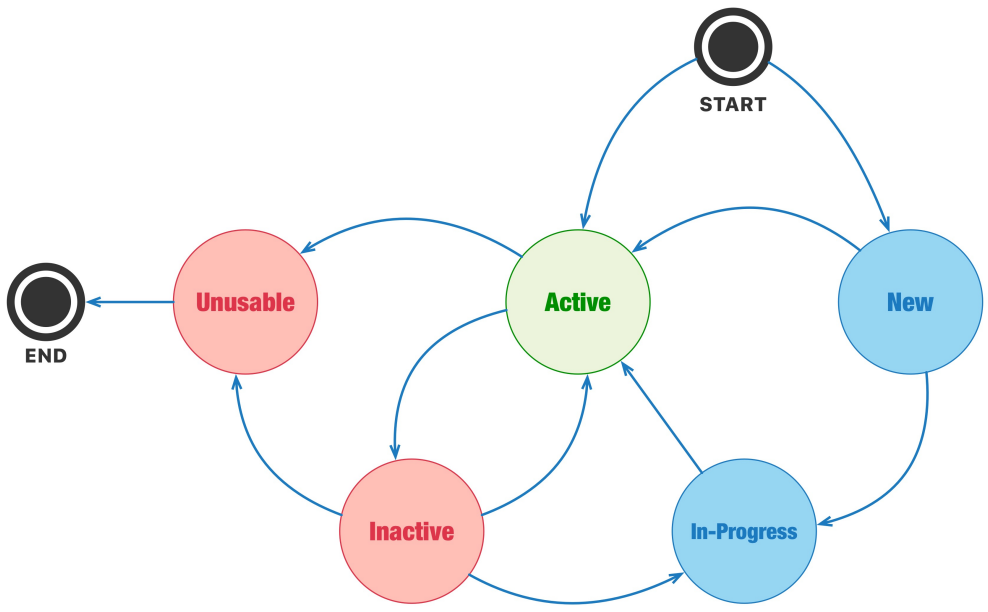

Figure 3 Activation status lifecycle

Figure 3 Activation status lifecycle depicts the different states that a Tokyo's ICCID activation status can transition to:

Active: The ICCID's has an active subscription and is setup for OneNumber forking. Active means the Tokyo device is expected to attach to the cellular network successfully. Unless the status is Active the device will not attempt to camp on the network.

In-Progress: Indicates that the ICCID network provisioning. This state should be used if the Tokyo's ICCID is already associated to the Primary's Phone number but provisioning is still pending.

Inactive: Indicates the ICCID's subscription is either cancel or suspended. Tokyo will not try to use an ICCID in Inactive state. An Inactive Tokyo ICCID can transition to Active.

New: An ICCID associated to the primary 4FF that's not active and has never been activated. This is currently treated equally to Inactive.

Unused: This is handled similar to Inactive with the caveat that the ICCID cannot transition to Active. This is equivalent to a "**dead sim**".

Supporting Mutiple SMDP+ Vendors

The Tokyo device will use the default SMDP+ Server defined in the Primary Device's carrier bundle. This carries overs the limitation that only one SMDP+ vendor can be supported.

To be able to support Multiple SMDP+ Vendors the following has to be supported:

1. During signup and at the time of releasing the profile, the carrier may choose to release the profile from a SMDP+ Vendor different from the Carrier Bundle default vendor. In that case, when the Websheet callback is invoked, the Fully Qualified Domain Name (FQDN) of the alternate SMDP+ must be specified in dataPlanAccountUpdatedWithInfo callback's input paramters dictionary:

   Sample:
   {
   eid: "*xxxxxxxxxxxxxxxxx*",
   iccid: "*yyyyyyyyyyyyyyyy*",
   alternateSmdpFqdn: "*alternate.smdp.domain.com*"
   }

2. The getSIMStatus when queried by the primary device, has to also include the alternate SMDP+ Server FQDN. This is optional and only needed if the ICCID(s) **download state** that was/were released from an alternate SMDP+ is/are still **pending.** That is, Tokyo has not yet downloaded the ICCID from the alternate SMDP+. Once the ICCID is downloaded, it should not longer be included in the getSIMStatus response. Including an alternate SMDP+ FQDN in the getSIMStatus response for a particular ICCID, will be used as a trigger by Tokyo to query the alternate SMDP+ FQDN, therefore it is important to only include only for pending downloads.

   Please work with your SMDP+ Vendor to determine the mechanism they offer/support to track download state.

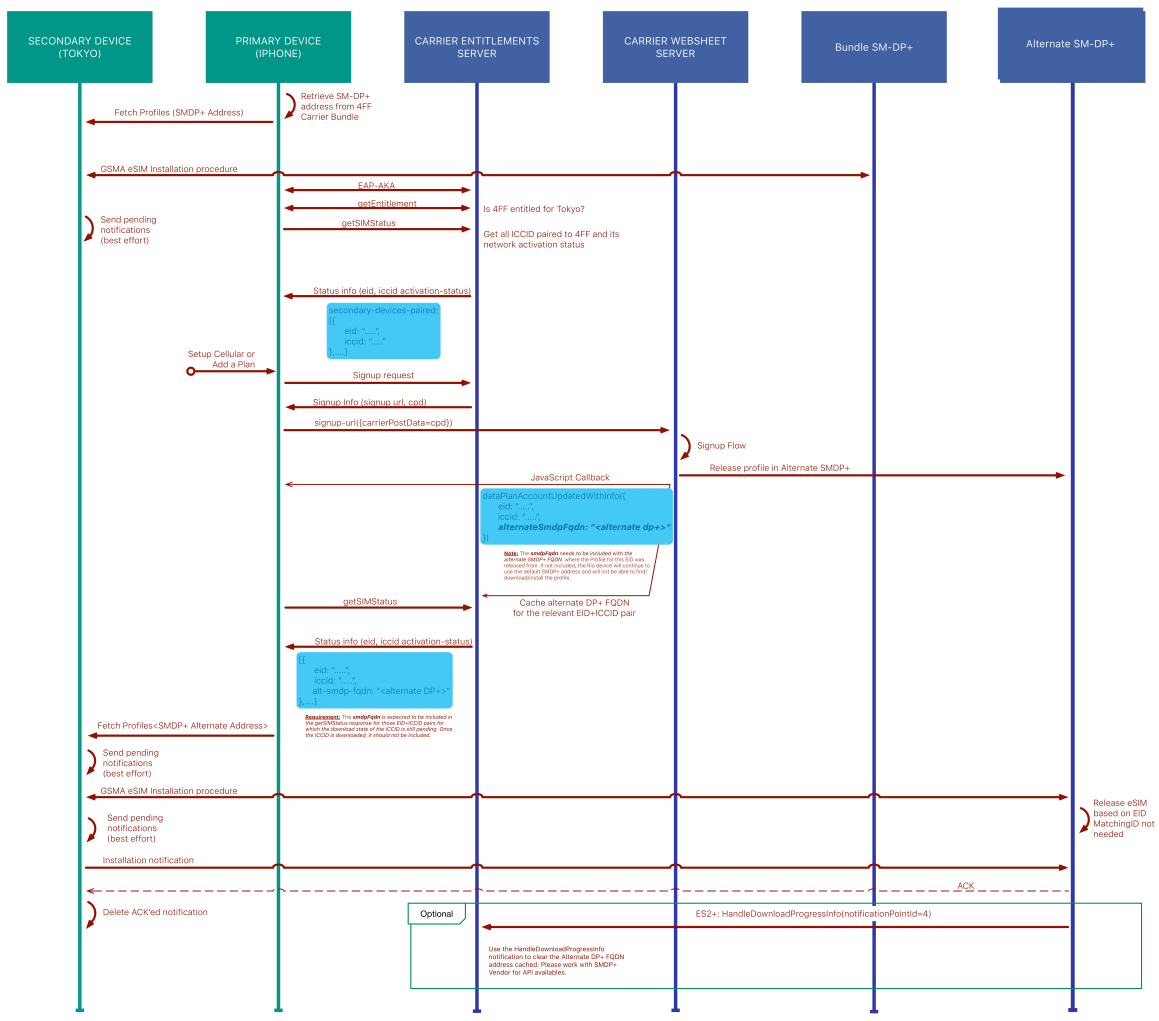

Figure 4. Multiple SMDP+ Vendor support flow

signUpForSIMService Action

This API initiates the Multi-SIM signup process. It encapsulates the following actions:

1) Carrier-managed flows which includes:
    a. Carrier T&Cs consent
    b. Cellular service sign-up
2) ICCID DownloadOrder/ConfirmationOrder for the companion device
3) Activates the ICCID in the carrier's billing system
4) Multi-SIM pairing into the Carrier's provisioning system (i.e. TAS, HSS, etc)
5) ICCID binding to the Companion device's EID

Request

Data Point	Explanation
request-id required	Type: Integer Identifies the particular action. The request-id is unique within a request.
action-name required	Type: Integer Identifies the name of the Entitlements request action Value(s): “signUpForSIMService”
primary-iccid required	Type: String Parent device’s (i.e. iPhone) Integrated Circuit Card Identifier.
eid required	Type: String Companion device’s Electronic Identifier
secondary-device-iccids optional	Type: Array Of String The Secondary device’s ICCIDs if available. See How the List of Secondary Device’s ICCIDs should be handled section for more information on this attribute. If this is an empty array or the attribute is not included (optional) it shall be interpreted by the network as the EID does not have any ICCID installed for the specific 4FF carrier.
secondary-device-imei optional	Type: String Secondary device’s International Mobile Equipment Identifier
secondary-device-meid optional	Type: String Secondary device’s Mobile Equipment Identifier. Passed if secondary device has a MEID. For future use
secondary-device-type optional	Type: String The category of device associated with the IMEI, if available. i.e. iPad

Response

Data Point	Explanation
response-id required	Type: Integer Must match the request-id for which this response applies. It must be unique within the response
status required	Type: Integer Value(s): STATUS_SUCCESS (6000)

	STATUS_DISALLOWED_ACTION (6002)
signup-url required	Type: String The URL with the entry point for the carrier-managed flows. The user will be handoff to this URL during the service sign-up process. Only HTTPS URLs values are supported.
signup-url-post-data required	Type: String Carrier-defined payload that is passed as the POST Request body during the Websheet hand off to the signup-url. This can be used for user auto-login into the Websheet and correlate the Entitlements request and Websheet request. The data will be passed in the HTTP POST request body, as a URL-encoded key/value pair in the same way that a browser submits an HTML form. The value of this attribute is assigned to "**carrierPostData**" key.

Expectations:

The Network should establish a limit of ICCID that can be reserved per user account+eid combination. Once such combination reaches its max. ICCID limit, the network should reject subsequent attempts to reserve new ICCIDs for that same combination.

How the List of Secondary Device's ICCIDs should be handled:

The Tokyo device can maintain 1 or more ICCIDs installed.

During sign-up, the Primary device will send the list of ICCIDs. All ICCIDs will match the Primary device's carrier.

The carrier shall try to re-use/activate any of the Secondary Device's ICCIDs provided.

If any of the Seconadry Device's ICCID is re-used/activated, the ICCID is confirmed in the DataActivationController.dataPlanAccountUpdatedWithInfo callback. This is how the Primary's device knows which ICCID was activated.
If the Entitlements server is not able to re-use/activate any of the Secondary Device's ICCIDs provided by the Primary's device, the network shall proceed to reserve, bind and activate a new ICCID from the eSIM Server. The new ICCID shall be passed back to the device in the DataActivationController.dataPlanAccountUpdatedWithInfo

getEntitlement Action

This entitlement is used by the device to verify if user is eligible to signup for Multi-SIM feature. If either DISABLED_ENTITLEMENT or INCOMPATIBLE_ENTITLEMENT status are returned to the device, it will be interpreted as the account cannot activate Tokyo's cellular service, halting the provisioning flow.

Request

Data Point	Explanation
request-id required	Type: Integer Identifies the particular action. The request-id is unique within a request.
action-name required	Type: Integer Identifies the name of the Entitlements request action Value(s): “getEntitlement”
entitlement-names required	Type: Array of Strings Enumerated value(s): “Multi-SIM”

Response

Data Point	Explanation
response-id required	Type: Integer Must match the request-id for which this response applies. It must be unique within the response
status required	Type: Integer Possible Value(s): STATUS_SUCCESS STATUS_DISALLOWED_ACTION STATUS_TEMPORARY_FAILURE
response required	Type: Array of Objects (see Response object definition) Contains the state of each of the entitlement-names queried

Response Object Definition

Data Point	Explanation
entitlement-name required	Type: String Enumerated value(s): “Multi-SIM”
entitlement-status required	Type: Integer Possible Value(s): STATUS_ENABLED_ENTITLEMENT STATUS_DISABLED_ENTITLEMENT STATUS_INCOMPATIBLE_ENTITLEMENT STATUS_PROVISIONING_ENTITLEMENT

enablePushNotification Action

Used by the device to send its APNS token.

The response should return STATUS_ENABLED_ENTITLEMENT unless it was unable to process or store the APNS token.

Request

Data Point	Explanation
request-id required	Type: Integer Identifies the particular action. The request-id is unique within a request.
action-name required	Type: Integer Identifies the name of the Entitlements request action Value(s): “enablePushNotification”
notifications required	Type: Array of Objects (See Notification Object definition) Array of objects containing the service for which the push notification must be enabled and the associated apns-token for the device to enable it for.

Response

Data Point	Explanation
response-id required	Type: Integer Must match the request-id for which this response applies. It must be unique within the response
status required	Type: Integer Possible Value(s): STATUS_SUCCESS STATUS_DISALLOWED_ACTION STATUS_TEMPORARY_FAILURE
notifications-response required	Type: Array of Objects (see Notification Response object definition) Array of objects. The server must provide an status code for each service sent in the request.

Notification Object (Request) Definition

Data Point	Explanation
notification-name required	Type: String References Push notification for Multi-SIM related notifications. Enumerated value(s): “Multi-SIM”
old-apns-token optional	Type: String Attribute is present if the device apns-token is updated. This will contain the previous apns-token and can be used by the network to update the previous record
apns-token required	Type: String base64-encoded by array with the device push token.

Notification-response Object (Response) Definition

Data Point	Explanation
notification-name required	Type: String References the push notification based on the request. Enumerated value(s): “Multi-SIM”
notification-status optional	Type: Integer Used by the device to determine whether the APNS token was processed and stored successfully by the Entitlements server. Enumerated value(s): STATUS_ENABLED_ENTITLEMENT – If apns-token was stored successfully STATUS_DISABLED_ENTITLEMENT – If an error occurred storing the apns-token

Entitlements APNS integration

The Entitlement client supports APNS notifications as a trigger mechanism for certain network events. APNS support for Tokyo use case is optional.

Push Messages Format

The push message format is a JSON payload that adheres to RFC 4627 with the following structure:

```
{
  “entitlement-update” : {
            “timestamp”: <string: ISO 8601 formatted date>,
            “trigger-actions”: [<action1>, <action2>,…,<actionN>]
  }
}
```

Each push message must have a timestamp specifying when the push message was generated by the server and an array of strings with the actions the device should perform. If the timestamp does not include the time zone offset, it will be assumed the time is expressed GMT. You should use the ISO 8601 complete date plus hours, minutes and seconds format (e.g. 2013-02-28T19:20:30-08:00). The ISO 8601 format that includes a decimal fraction of a second is not currently supported.

New trigger-actions for Tokyo and Multi-SIM

The following ‘trigger-actions’ have been added:

“multi-sim-signup-changed”

This trigger-action is used to notify the primary device that eICCID and Multi-SIM network provisioning has been completed. The iPhone upon receiving this notification it will query the Entitlements server with getSIMStatus to get status updates and iccid paring.

Example push message:

```
{
  “entitlement-update” : {
            “timestamp”: “2014-08-05T11:15:31-08:00”,
            “trigger-actions”: [“multi-sim-signup-changed”]
  }
}
```

“entitlements-changed”

This trigger-action can be used still be used in the context of Tokyo project. It causes the device query getEntitlement which will include multi-sim entitlement.

Websheet Integration

For the Carrier-managed flows a Websheet approach will be used. The entry point to the carrier-managed flow or URL is provided by the Entitlements server through the signupForSIMService's signup-url or getSIMStatus' manage-account-url.

At the time of the handoff, the device will send a HTTP(S) POST request to the URL provided and pass the signup-url-post-data or manage-account-url-post-data as the body as a JSON payload (see below Carrier POST Data JSON format).

Websheet Carrier-Managed Flow

While in the carrier flow, the carrier is responsible for presenting Terms and Conditions (T&Cs), soliciting information required to sign up for MultiSIM, getting and activating the eSIM.

Profile Inventory Management

Note that once a profile has been bound to a EID, it will be downloaded the next time the device checks for available profiles whether it has been activated or not. In most cases, this will occur by entering monitor mode (described below) at the conclusion of the activation flow. However, there are other events that will trigger the device to check for available eSIMs, such as a device or baseband reboots. If any eSIMs are bound but not activated, they will still be downloaded.

For this reason, it is important that a profile be unbound and checked in if a user does not finish the web sheet flow after a profile has been bound but before the profile has been activated. This might include situations such as a failed payment followed by a session timeouts, an explicit cancellation. If no cleanup occurs before the profile is downloaded, profile inventory may be "burned" and unusable in the future. Apple recommends that a profile not be bound until as late as possible in the activation flow; preferably, not until immediately before it will be activated.

Websheet flow completion

After the eSIM has been successfully bound and activated, the device needs to be told that the carrier flow is completed, and that the eSIM is ready to download. This is done by entering "monitoring mode" using the following JavaScript calls from within the carrier flow:

DataActivationController.dataPlanAccountUpdatedWithInfo(Dictionary)

This callback method shall be invoked if an ICCID for the Secondary Device is activated successfully. It will cause the device to contact the the SM-DP+ Server to download / install any available eSIM profile onto the eUICC.

@params: Receives a Hash input param with the following keys:

eid : The eid of the Secondary Device for which the iccid has been released
imei : The IMEI of the secondary device for which the iccid has been released
meid : The MEID of the secondary device for which the iccid has been released
iccid : The ICCID that was activated for the EID's provided during service signup

alternateSmdpFqdn: Optional FQDN of the SMDP+ server where the ICCID was released from. Only required if the SMDP+ is different from the default specified in the Carrier Bundle. See Supporting Multiple SMDP+ Vendors section

@return: none

The ~~dataPlanAccountUpdatedWithInfo~~ dataPlanAccountCancelled can be used from the Manage Account websheet when any subscription change is performed from the Manage Account websheet. For instance, if the carrier supports the ability to cancel the Tokyo subscription from the Manage Account Websheet, the ~~dataPlanAccountUpdatedWithInfo~~ dataPlanAccountCancelled callback must be invoked everytime an user cancel the Tokyo subscription from the Manage Account Websheet, this is required because potentially a ICCID activation-status will transition from **Active** to **Inactive**. In such case, the ~~dataPlanAccountUpdatedWithInfo~~ dataPlanAccountCancelled callback must be invoked.

In this callback parameters, the EID, IMEI and MEID properties are mutually exclusive. Only one (1) must be provided.

Sample Invocations:

```
/* Paramters using an EID */
var paramsWithEid    = {
   eid: “00000000000000000”,
   iccid: “1111111111111111111”
};  // valid

/* Paramters using an IMEI */
var paramsWithImei  = {
   imei: “100000000000000000”,
   iccid: “1111111111111111111”
};  // valid

/* Paramters using an MEID. Assumes Tokyo has MEID */
var paramsWithMeid = {
   meid: “200000000000000000”,
   iccid: “1111111111111111111”
}; // valid

DataActivationController.dataPlanAccountUpdatedWithInfo(paramsWithEid);
DataActivationController.dataPlanAccountUpdatedWithInfo(paramsWithImei);
DataActivationController.dataPlanAccountUpdatedWithInfo(paramsWithMeid);
```

DataActivationController.doneSelected()

This callback causes the websheet to be dismissed. This can be use at the end of the signup flow either success or failures, to allow the user close it through HTML UI.

@params: none
@return: none

DataActivationController. dataPlanAccountCancelled()

This callback is intended to be use in the Manage Account websheet. It should be used any time the user as part of the interaction with the Manage Account Websheet, changes the state of the Tokyo's ICCID. For instance, if the user is cancels the Tokyo subscription from the Manage Account Websheet, this callback must be invoked. Invoking this callback, causes the primary device to send the getSIMStatus in order to get the update state.

@params: none
@return: none

JavaScript Callbacks

The following callbacks are provided to help the Websheet user experience aspects of the carrier-managed flow. Adoption of the following callbacks are optional.

DataActivationController.hideCancelButtonSelected()

Hides the Websheet's native cancel button. This can be used during the signup process. Hiding the cancel button might render the user unable to go back. Note the user will still be able to exit by pressing the Home button.

@params none
@return none

DataActivationController.showCancelButtonSelected()

Makes the native cancel button visible again if hideCancelButtonSelected() was invoked.

@params none
@return none

DataActivationController.showVerifyingIndicator()

Displays an Activity Indicator (a.k.a. spinning gear) in the Websheet navigation bar.

@params none
@return none

DataActivationController.dismissKeyboard()

The Keyboard view is pinned to the Websheet when is brought up the first time an input field gains focus, and remains visible until the Websheet is closed. This callback allows the websheet implementation to programmatically hide the keyboard.

@params none
@return none

Carrier POST Data JSON Format

The HTTP POST Request for Websheets now uses JSON format as opposed to URL-encoded key/value pairs. JSON provides more flexibility.

JSON Format specification:

```
{
        “carrierPostData” : <String>,
        “secondary-device-active” :  {
                “eid” : <String>,
                “iccids” : <Array of String>,
                “display-name” : <String>,
                 “device-imei”: <String>,
                “device-meid”: <String>,
                “device-type”: <String>,
                “primary-device-iccid” : <String>

        }
}
```

carrierPostData: will contain the value of either signup-url-post-data or manage-account-url-post-data received through the Entitlements server.

secondary-device-active: is a JSON object containing the EID, ICCID and display-name of the secondary device active at the moment of launching the websheet. This is to aid the Websheet server.

eid: Tokyo’s Embedded UICC

iccids: The Secondary device’s installed ICCIDs. This only include the ICCIDs matching the iPhone’s 4FF carrier. If the attribute is missing or it’s an empty string, it should be interpreted as the device has 0 ICCIDs installed for the specific carrier.

display-name: The user-defined Tokyo’s name located in Settings / General / About. This is not unique across devices.

device-imei: Secondary device’s International Mobile Equipment Identifier

device-meid: Secondary device’s Mobile Equipment Identifier. Only if hardware supports it.

device-type: The category of device associated with the IMEI, if available. i.e. iPad

primary-device-iccid: Primary device’s (i.e. iPhone) Integrated Circuit Card Identifier.

Sample:

```
{
        “carrierPostData” :  “dG9rZW49MTIzNDU2Njc4OTBmZmZmO3Q9MTttPTEwCg==”,
        “secondary-device-active” :  {
                “eid” : “60010002000b071448ca3a00008e3210”,
                “iccids” : [“8901xxxxxxxxxxxxxxx”],
```

```
                "display-name" : "John's iPad",

                 "device-imei": "3591xxxxxx8005602",

                "device-meid": "3591xxxxxx80056",

                "device-type": "iPad",

                "primary-device-iccid" :  "8901xxxxxxxxxxxxxxx"


        }

}
```

HTTP Headers:

The HTTP Content-Type header will be set to application/json to indicate the format of the POST data body.

Provisioning Error Handling Criteria

For network-side provisioning errors the recommendation is to use the websheet to handle them and provide the necessary feedback to the user about the issue and possibly the action to recover from it.

The following are instances of errors where websheet should be used to provide UI feedback to users about the failure:

1. Irrecoverable DownloadOrder / ConfirmOrder failures
2. ICCID network provisioning failures
3. EID has reach any carrier-defined ICCID limit
4. Etc.

Manage Account Websheet

The Manage account websheet (manage-account-url) can be used for post-signup management tasks (i.e. update emergency address for secondary device(s)). The options offered are carrier-dependent with one exception. The Manage Account websheet **must** support at least the option to initiate a signup flow, similar to the signup flow from the signup-url websheet. This is required for instances where the ICCID is installed and the user wants to re-enable cellular. In that case, the Manage Account websheet will be used instead of the signup websheet.

eSIM Server Layer

The Tokyo project expect the use of a GSMA-complaint eSIM server. For details on the eSIM orders interfaces for eSIM allocation and download refer to the latest **GSMA RSP Specification SGP.22 V2.0a** or talk to your RSP Vendor.

GSMA Specification Deltas

Matching ID

For Matching ID, in GSMA SGP.22 Section 4.1.1, it is specified that “The MatchingID is mandatory information (but MAY be zero-length)”. While in other places in the spec, empty string is used. The expected DER encoded value should be the tag followed by 0x00.

Appendix A: Common Provisioning Use Cases

The following section depicts the flows for most common provisioning use case scenarios.

1. Use Case: Provisioning First Tokyo device

This covers the use cases where the user is provisioning the first Tokyo device paired to its iPhone.

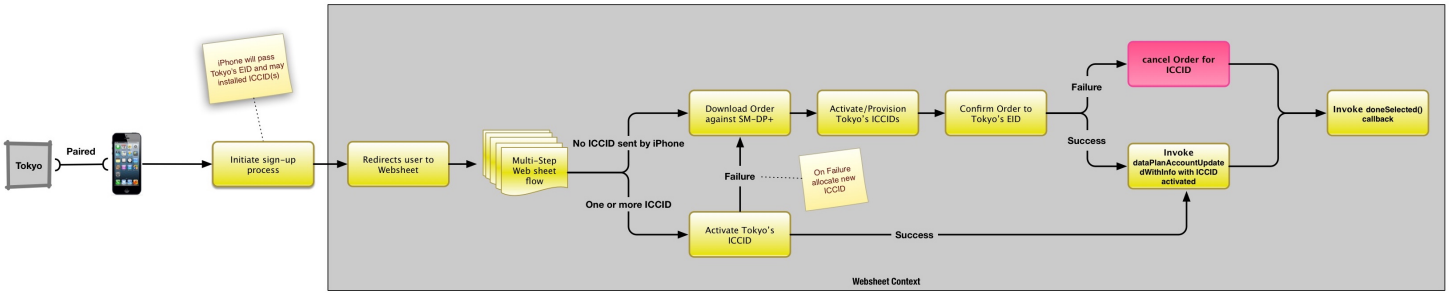

Figure 5 Provisioning First Tokyo device use case

2. Use Case: Pairing second Tokyo device

This use case is when the user is trying to pair an additional Tokyo device to its iPhone.

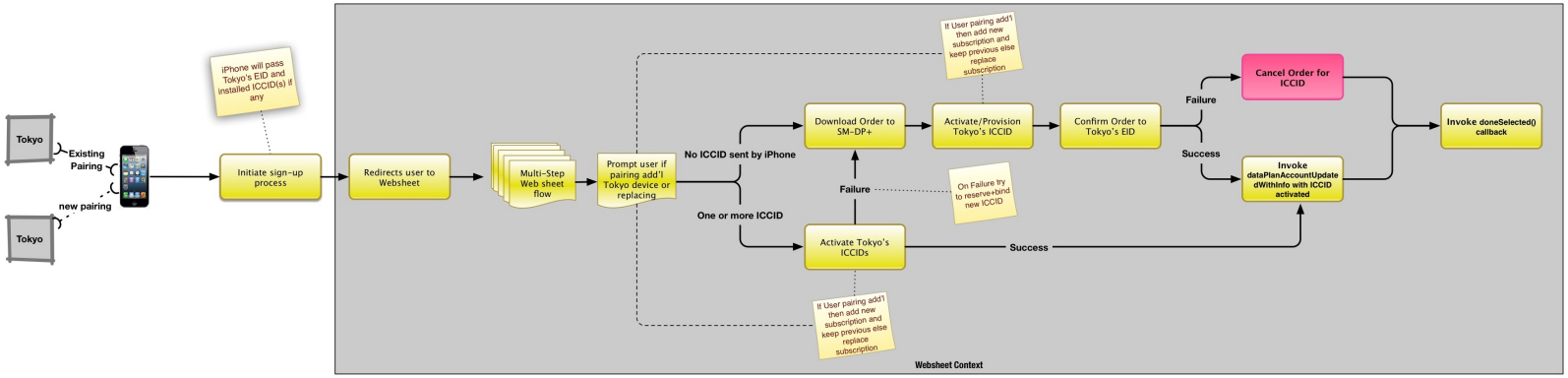

Figure 6 Pairing a second Tokyo device use case

Appendix B: Entitlements Actions matrix

Entitlements Action	*Devices supporting*	*Frequency triggered*
getAuthentication	iPhone	At boot, Frequency polling
postChallenge	iPhone	On AKA-Challenge
getEntitlements	iPhone	On boot, On Push, Recurring
getSIMStatus	iPhone	On boot, On Push, On User Action
signUpForSIMService	iPhone	On User Action
enablePushNotification	iPhone	On boot, On demand

Appendix C: Wi-Fi Calling and Emergency Address Considerations

Certain countries require wireless operators to maintain a user designated address for emergency purposes as a pre-requisite to provide Wi-Fi Calling services. This section describes the approach for project Tokyo on how to deal with such requirements when needed.

Note: The process described below is optional. It can be enabled in locations / carriers where it is required per local regulations.

Tokyo's Cellular Service Sign-up

When the user is sign-up for Tokyo's Cellular service, as part of the Tokyo's Websheet flow there must be the necessary business logic to verify if there's a valid Emergency Address already registered for the particular account.

If there is not a valid emergency address available, the Tokyo's Websheet implementation must provide ways for the user to specify one. It is recommended to provide a default address to the user for a better experience. See Figure 7 Emergency address during sign-up reference flow

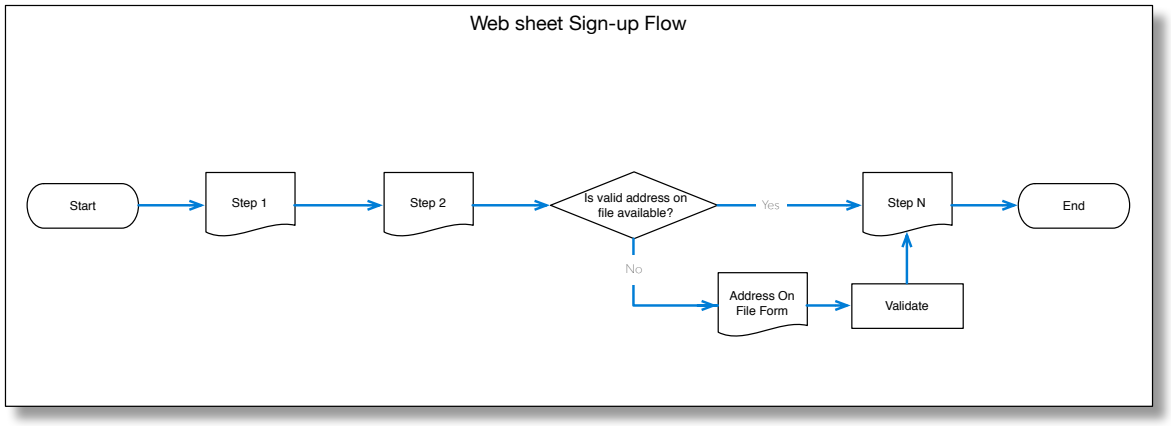

Figure 7 Emergency address during sign-up reference flow

Periodic Emergency Address Validation

In addition to verifying the availability of an emergency address during sign-up and collecting it in case it is not available. There may be the need to periodically validate the emergency address condition is met. In such case, the device can periodically check for VoWiFi Entitlement and getPhoneServicesAccountStatus action.

In addition, the network can request the device to check at any time using a APNS push (reg-loc-update).

This process is consistent with the existing Wi-Fi Calling Entitlement implementation. See Figure 8 Emergency address periodic check reference flow

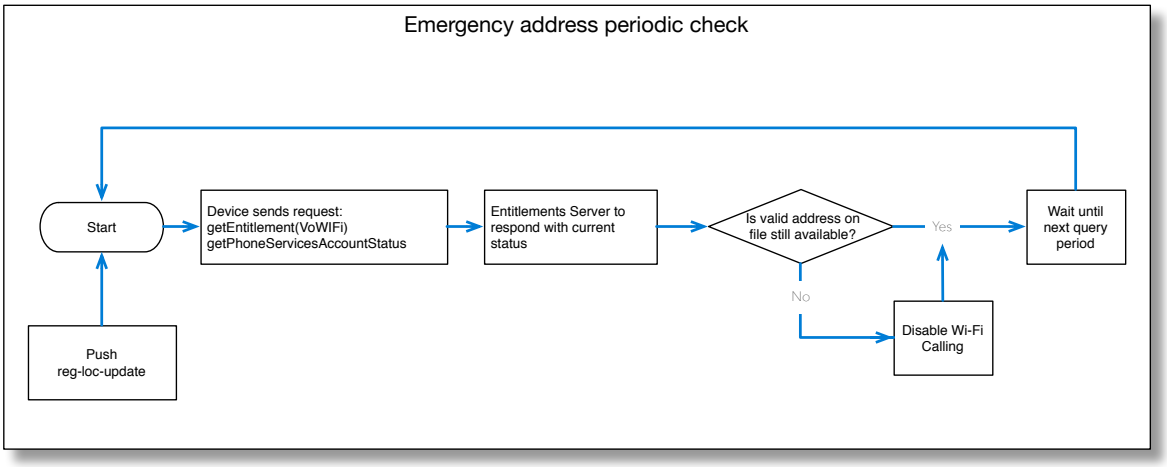

Figure 8 Emergency address periodic check reference flow

Please refer to the latest Wi-Fi Calling Entitlements specification for more details on the implementation requirements for getEntitlement(VoWiFi) and getPhoneServicesAccountStatus

Revision History

Version 1.2.5

Date: March 31th, 2017

- Adds new JS Callback **dataPlanAccountCancelled** for Manage Account Websheet. This callback replaces previous approach to invoke **dataPlanAccountUpdatedWithInfo** without parameters (Ref: 31235010)
- Adds new JS Callback **dismissKeyboard** to hide or unpin the Keyboard view programmatically
- Fixes a typo for enablePushNotification action response where incorrectly the property was named 'notification-response' and the correct name is 'notifications-response'.

Version 1.2.4

Date: March 3rd, 2017

- Updates **getSIMStatus** response to support IMEI and MEID, in addition to EID.
- Updates **dataPlanAccountUpdatedWithInfo** callback to support IMEI and MEID, in addition to EID.

Version 1.2.3

Date: January 27th, 2017

- Adds support for multiple SMDP+ Vendors. Includes updates to getSIMStatus and JavaScript callbacks.
- Adds clarification of how to use the dataPlanAccountUpdatedWithInfo callback in the Manage Account Websheet.
- Updates Appendix B.

Version 1.2.2

Date: September 21st, 2016

- Enhanced the Websheet Post data JSON object. The object mirrors the same attributes found in the signUpForSIMService's API request. This is intended to be use for Cellular Sign-up initiated from Manage Account Websheet.

Version 1.2.1

Date: September 7th, 2016

Minor edits

Version 1.2

Date: August 17th, 2016

As the Tokyo project moves toward the integration with GSMA SM-DP+ server for eSIM profiles management as opposed to the previous plan to use Apple's eSIM Server solution, this document has been updated to adapt the changes as consequence of the adoption of GSMA RSP solution. The following are a summary of changes part of this update:

- Removes support for non-GSMA eSIM Server layer interfaces
- Changes to support GSMA RSP Server integration
- Removes support for eSIM download at user's intent (a.k.a. auto-download)
- Added provisioning errors handling strategy section

- Added requirement to support option to initiate signup flow from Manage Account Websheet URL
- Added ES2+ MatchingId empty requirement for On-device entitlements flow
- Added glossary section
- Updated Appendix A with point of sales use case
- All references to CSN attribute has be renamed to EID

Version 1.1

Date: February 3rd, 2016

- Added Appendix C with emergency address considerations

Version 1.0.2

Date: January 25th, 2016

- Renamed Profile Version to Profile Type in Reserve ICCID API
- Updated Entitlements actions matrix in Appendix B

Version 1.0

Date: November 20th, 2015

- Removed DRAFT watermark
- Changed definition to refer to iPhone as “primary device” instead of “parent device”. Attributes were renamed accordingly.
- Changed definition to refer to Tokyo as “secondary device” instead of “companion device”. Attributes were renamed accordingly.
- getSIMStatus action:
    - Renamed attributes parent-* and companion-* attributes to primary-* and secondary-* respectively
    - Removed imsi attribute from secondary-devices-paired object
    - Added activation-status values in secondary-devices-paired
    - Added manage-account-url and manage-account-url-post-data to support a websheet for user’s account management
    - Added optional esim-server-url to override default value defined in Carrier Bundle
- signUpForSIMService:
    - Removed iPhone’s imsi from the request to avoid duplication. The iPhone’s IMSI is determined via EAPAKA Authentication (getAuthentication action)
    - Marked ‘eid’ attribute as required instead of optional
- getSubscriptionAndUsageStatus
    - Enhanced getSubscriptionAndUsageStatus action to query for Tokyo’s plan information and usage stastics
    - Added new “extended-usage-info” key in the request.
    - Added new “extended-usage-info” structure in the response to carry over all Tokyo’s related information
- Websheet
    - Renamed callback dataPlanAccountUpdatedWithIccid to dataPlanAccountUpdatedWithInfo
    - Changed input dataPlanAccountUpdatedWithIccid input parameter definition
    - Added new callback signupFailure for deployments using ICCID download at user’s signup intention

    - The *-post-data received from Entitlements is now sent assigned to “carrierPostData” key in the body of the HTTP POST request.
- eSIM Server Layer
    - Added support for JSON schema for to eSIM Server APIs for consistency purposes.

Version 0.8.1

Date: October 21th, 2015

- Added optional eSIM Server’s eICCID Deletion notification API.

Version 0.8

Date: October 16th, 2015

- Added Provisioning Approaches section with the 2 possible provisioining implementations.
- eSIM Server API are now optional based on the provisioning approach
- Added enablePushNotification action now with support for Multi-SIM APNS token (new topic)
- Added companion-iccid attribute to signUpForSIMService action

Version 0.6

Date: September 18th, 2015

- Updated Entitlements Status code table
- Updated provisioning sequence diagram

Version 0.5

Date: September 17th, 2015

- Added new attribute to getSIMStatus
- Added new attributes to signUpForSIMService API

Version 0.1

Date: September 16th, 2015

- Initial draft