Mobile Connect Core Technical Requirements

Version 1.2

27 April 2022

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

Security Classification: Non-confidential

Access to and distribution of this document is restricted to the persons permitted by the security classification. This document is confidential to the Association and is subject to copyright protection. This document is to be used only for the purposes for which it has been supplied and information contained in it must not be disclosed or in any other way made available, in whole or in part, to persons other than those permitted under the security classification without the prior written approval of the Association.

Copyright Notice

Disclaimer

The GSM Association ("Association") makes no representation, warranty or undertaking (express or implied) with respect to and does not accept any responsibility for, and hereby disclaims liability for the accuracy or completeness or timeliness of the information contained in this document. The information contained in this document may be subject to change without prior notice.

Antitrust Notice

The information contain herein is in full compliance with the GSM Association's antitrust compliance policy.

Table of Contents

1 Introduction

This document in conjunction with PDATA.01 OpenID Connect Mobile Connect Profile [1] provides a set of mandatory requirements that must be implemented when deploying Mobile Connect.

Depending on the individual products the Operator is implementing, they should also refer to and abide by the relevant product Technical Requirements:

- PDATA.02 Mobile Connect Authorisation Technical Requirements [3]
- PDATA.08 Mobile Connect Identity Services Technical Requirements [4]

Note that a common requirement across the majority of the Mobile Connect product portfolio is to authenticate the user. This operation may be carried out in isolation (Mobile Connect Authenticate products) or combined implicitly with another action such as seeking user authorisation for a transaction (Mobile Connect Authorise products) or seeking user permission to share attributes with a requesting party (Mobile Connect Identity Service). Hence the capability of authenticating the user underpins much of the Mobile Connect framework of services and therefore all requirements relating to authentication are included within this document (rather than being split out as a separate product-specific Technical Requirements document as for the other products).

For information on the Mobile Connect Authenticate products please refer to the PDATA.27 Mobile Connect Product Definition [9].

For further information on the Mobile Connect framework please see Mobile Connect Architecture [2]. For guidance on implementation and best practise please see the Mobile Connect Wiki accessed from the GSMA InfoCentre.

1.1 Scope

The scope of this document is to provide mandatory requirements that must be implemented by Operators to implement Mobile Connect in accordance with Release 2. The scope covers:

- Identity Gateway (ID GW) implementation.
- Authenticator selection.
- Pseudonymous Customer Reference (CR) format and association.
- Server-based invocation
- OpenID Connect (OIDC) implementation.
- Security requirements.
- Backwards compatibility of Mobile Connect.

1.2 References

Ref	Doc Number	Title
[1]	PDATA.01	Mobile Connect Profile V1.2.
[2]	PDATA.13	Mobile Connect Core Technical Requirements V1.0
[3]	PDATA.02	Mobile Connect Authorisation Technical Requirements V1.0
[4]	PDATA.08	Mobile Connect Identity Services Technical Requirements V1.0
[5]	PDATA.41	Mobile Connect Technical Reference V1.0
[6]	PDATA.17	Mobile Connect Technical Architecture V2.0
[7]	PDATA.28	Mobile Connect Lifecycle Events V1.2
[8]	PDATA.40	Mobile Connect Lifecycle Technical Solutions V1.0
[9]	PDATA.27	Mobile Connect Product definition V2.2
[10]	PDATA.18	1AP.06 API Exchange System Architecture V1.0
[11]	PDATA.19	1AP.03 API Exchange Business Process Guide V1.0
[12]	PDATA.24	EH.V3 Discovery API Specification
[13]	PDATA.03	Mobile Connect Authenticator options CPAS4 V1.2
[14]	PDATA.04	Mobile Connect SIM applet authenticator CPAS8 V0.1
[15]	PDATA.09	Mobile Connect Smartphone Application Authenticator V1.0
[16]	PDATA.43	Mobile Connect Release 2 Technical Overview (MNOs) V0.2
[17]		Mobile Connect Brand Communication Guidelines_v2_06_15
[18]		Mobile Connect User Flow Design kit V1.2
[19]	PDATA.01	Mobile Connect Profile V1.2.

1.2.1 International Standards references

Ref	Doc Number	Title
[30]	OpenID Connect	"An interoperable authentication protocol based on the OAuth 2.0 family of specifications" available at http://openid.net/specs/openid-connect-core-1_0.html https://openid.net/specs/openid-connect-basic-1_0.html
[1]	RFC 2119	"Keywords for use in RFCs to Indicate Requirement Levels," S. Bradner, March 1997. Available at http://www.ietf.org/rfc2119.txt
[2]	RFC 2616	"Hypertext Transfer Protocol (HTTP) an application level protocol," J Gettys, J. Mogul, L. Masinter, P. Leach, T. Berners-Lee June 1999. Available at http://www.ietf.org/rfc/rfc2616.txt
[3]	RFC 6749	"The OAuth 2.0 Authorization Framework," D. Hardt, Ed. October 2012 available at http://www.ietf.org/rfc/rfc6749.txt
[4]	RFC 4122	A Universally Unique Identifier (UUID) URN Namespace. https://www.ietf.org/rfc/rfc4122.txt
[5]	RFC 2246	Dierks, T. and C. Allen, "The TLS Protocol Version 1.0," RFC 2246, January 1999
[6]	RFC 3339	Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps," RFC 3339, July 2002
[7]	RFC 3986	Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax," STD 66, RFC 3986, January 2005

[8]	RFC 4627	Crockford, D., "The application/JSON Media Type for JavaScript Object Notation (JSON)," RFC 4627, July 2006
[9]	RFC 5246	Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246, August 2008
[10]	RFC 5322	Resnick, P., Ed., "Internet Message Format," RFC 5322, October 2008
[11]	RFC 5646	Phillips, A. and M. Davis, "Tags for Identifying Languages," BCP 47, RFC 5646, September 2009
[12]	RFC 6750	Jones, M. and D. Hardt, "The OAuth 2.0 Authorization Framework: Bearer Token Usage," RFC 6750, October 2012
[13]	RFC 6819	Lodderstedt, T., McGloin, M., and P. Hunt, "OAuth 2.0 Threat Model and Security Considerations," RFC 6819, January 2013 (TXT).
[14]	RFC 7519	M. Jones, J Bradley, N. Sakimura "JSON Web Token (JWT)", RFC 7519, May 2015
[15]	ISO 29115	International Organization for Standardization, "ISO/IEC 29115:2013 -- Information technology - Security techniques - Entity authentication assurance framework," ISO/IEC 29115, March 2013
[16]	ISO 3166-01	International Organization for Standardization, "ISO 3166-1:1997. Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes," 1997
[17]	ISO 639-1	International Organization for Standardization, "ISO 639-1:2002. Codes for the representation of names of languages -- Part 1: Alpha-2 code," 2002
[18]	ISO 8601-2004	International Organization for Standardization, "ISO 8601:2004. Data elements and interchange formats - Information interchange - Representation of dates and times," 2004

1.3 Abbreviations

Term	Description
ACR	Anonymous Customer Reference
CPAS	Core Products and Solutions
CRUD	Create, Read, Update and Delete
DTBS	Data to be signed
ID GW	Identity Gateway
IDP	Identity Provider
IMSI	International Mobile Subscriber Identity
JWT	JSON Web Token
LDAP	Lightweight Directory Access Protocol
LoA	Level of Assurance
MCC	Mobile Country Code
MLS	Message Layer Security
MNC	Mobile Network Code
MSISDN	Mobile Station International Subscriber Directory Number
OIDC	OpenID Connect
OTA	Over-the-air
PCR	Pseudonymous Customer Reference

Term	Description
RDBMS	Rational Database Management System
SDK	Software Development Kit
SMSC	Simple Message Service Centre
TLS	Transport Layer Security
UI	User Interface
URI	Universal Resource Identifier

1.4 Technical Documentation Map

The Mobile Connect architecture, technical specifications and implementation guidelines are encompassed by a set of documentation as laid out below:

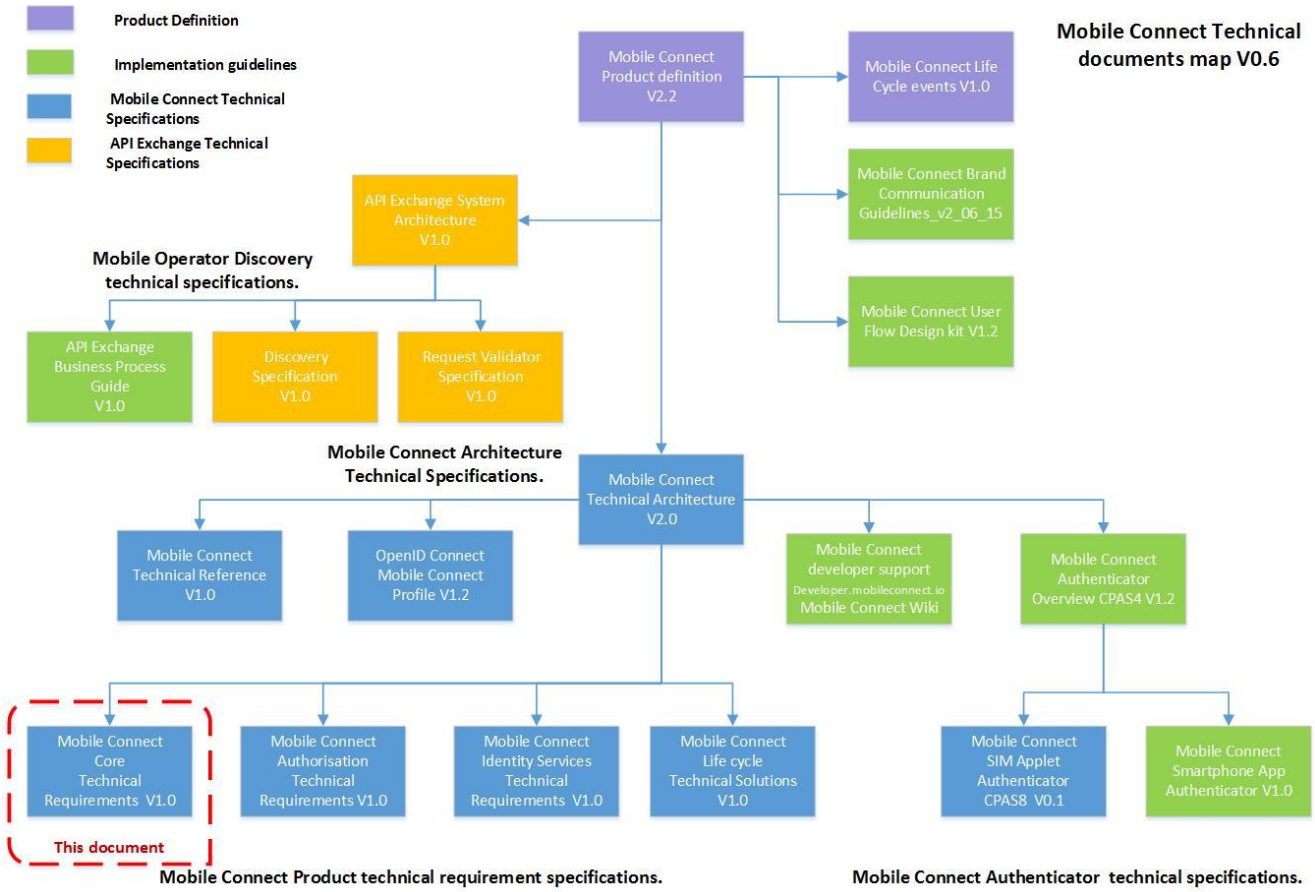

Figure 1: Mobile Connect technical documentation map

2 Mobile Connect Actors

Mobile Connect is a global mobile industry solution that provides simple, secure and convenient tools for end-users to access online services and to authorise digital transactions by using the end-user's unique mobile number to verify and authenticate their identity.

The end-users access Mobile Connect via Service Providers who have integrated one or more of the Mobile Connect portfolio of products. The Service Providers request these products by using different parameters in an Open ID Connect (OIDC) request. The following diagram illustrates the interactions between different Mobile Connect actors for the Mobile Connect authentication products. PCR

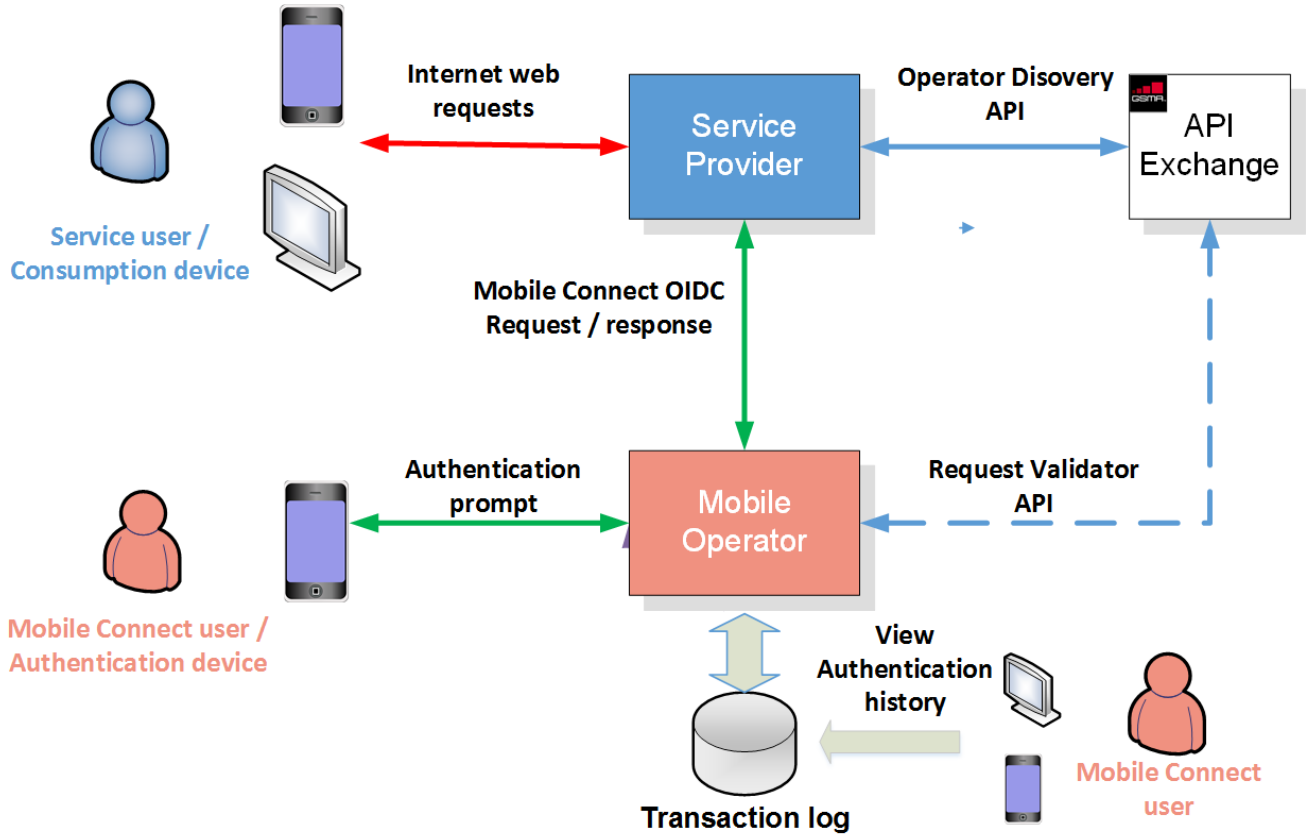

Figure 2: Mobile Connect actors

2.1.1 Service Provider

A Service Provider that has integrated Mobile Connect and uses the Mobile Connect products as part of their service to the end-user. Their Service Users use Mobile Connect products as part of the SP user flow; e.g., to authenticate (login etc.) or authorise a transaction.

2.1.2 Service User

The Service User consumes services from the Service Provider. As part of the user flow, the Service User may need to identify themselves to the Discovery Service so that the SP knows to which Operator to submit their Mobile Connect request.

2.1.3 Discovery

Discovery is used to identify the Operator for a specific Mobile Connect User and the relevant API endpoints supporting the Mobile Connect services. For security/privacy reasons Discovery does not disclose any Mobile Connect user personal information to the SP. See [13] for Mobile Connect Discovery API information.

2.1.4 Operator Identity Gateway (ID GW)

The Operator's Identity Gateway is the provider of Mobile Connect services. It implements Mobile Connect OpenID Connect (OIDC) flows and interacts with the Mobile Connect User.

The main characteristics of the ID GW are:

1. Main logical component called by the SP to consume the Mobile Connect service.
2. Provides an abstraction layer between the SP and the Mobile Connect system components.
3. Implements OpenID Connect Mobile Connect Profile endpoints for the SP to call. The Discovery service returns the ID GW endpoints to the SP.
4. Optionally, invokes the Validation API of the API Exchange to validate the credentials passed by the SP in an OIDC request.

2.1.5 Authentication Device

The device authenticating an SP-requested transaction is always a mobile device that is addressed via the MSISDN of the user. It receives authentication requests from the Operator and is used by the Mobile Connect User to approve or reject authentication requests.

2.1.6 Mobile Connect User

The Mobile Connect User interacts with the Operator using their mobile device to authenticate or authorise transactions requested by the SP.

2.1.7 Consumption Device

The device used to consume a SP's services, by the Service User. Consumption of SP services can be through different channels including directly via mobile and desktop/tablet browsers and mobile applications, and indirectly via voice systems, interaction with Customer Care agents, etc.

Any of these channels can integrate with Mobile Connect to use its services.

3 High Level Requirements

3.1 Core Requirements

The following core requirements have been derived from the Mobile Connect Product Definition document [9].

Requirement	Description
MC_RQ01.1 Common SP deployment	The Mobile Connect deployment MUST expose a common API towards Service Providers.
MC_RQ01.2 Authenticator support	Operator must support LoA2 (OK) and may support LoA3 (PIN) Level of Assurance. Note the choice of authenticator to meet these levels is up to the Operator depending on local market conditions. If possible, local Operators within a market should use the same authenticator to provide a consistent user experience.

MC_RQ01.3 Operation while roaming	The Mobile Connect deployment MUST be able to operate while the user is roaming on to another Mobile network.
MC_RQ01.4 Support of contract and pre-paid customers	The Mobile Connect deployment must support both contract and pre-paid Operator customers although it is recognised that for pre-paid customers it may not be possible to provide Identity Services if the Operator has no information on that particular user
MC_RQ01.5 User experience	As one of the main design goals of Mobile Connect is to remove customer friction from services the Mobile Connect deployment SHOULD target the minimum number of end user steps in order to maximise service uptake and usage.
MC_RQ01.6 Mobile device support	The Mobile Connect deployment SHOULD support all mobile device types within a local market to maximise the addressable market.
MC_RQ01.7 Support for MVNO	Some MVNOs use the same MCC/MNC as the parent Mobile Network the implementation must be able to detect this use case and take the action defined by the Operator.
MC_RQ01.8 High availability SLA	The Mobile Connect deployment should provide high availability/reliability.
MC_RQ01.9 Self-care portal	A self-care portal should be provided to allow the user to manage their Mobile Connect account. For example to reset a PIN.
MC_RQ01.10 Service Provider billing	The Mobile Connect deployment must generate Mobile Connect transaction records sufficient for Service Provider billing.

Table 1: Core product requirements

3.2 Authentication Product Requirements

In addition to the core requirements defined above, implementation of the Authentication products will introduce the following additional requirements in accordance with the Mobile Connect Product Definition document [1]:

Requirement	Description
MC_RQ01.2.1 Authentication request	The Authentication products must offer a mechanism through which the SP can request a user to Authenticate or reject a transaction (e.g. Login).
MC_RQ01.2.2 Authentication Levels	The Authentication product category should support single (LoA2) and two factor (LoA3) Authentication. Note product names LoA2 = Mobile Connect Authenticate LoA3 = Mobile Connect Authenticate Plus
MC_RQ01.2.3 Authentication prompt	The Mobile Connect Authentication product must present a prompt to the user that includes SP short name (from the SP, 16 bytes max) Note: The SP can only provide a pre-registered application short name, using the `client_name` parameter in the OIDC request. The ID GW must check whether the incoming `client_name` OIDC request parameter matches one of the pre-registered application short names[1] for that SP, by comparing with its registry database. If it doesn't match, the ID GW must

[1] In rel2, only one application short name per service provider is allowed. In the future releases, multiple application short names are allowed and SP can choose one of the pre-registered application short names, according to its application needs.

Requirement	Description
	return an `invalid_request` error as per the Mobile Connect Technical Reference [5].
MC_RQ01.2.4 Supported authenticators	For both Mobile Connect Authenticate products valid authenticators include: • SMS + URL (Mobile Authenticate only) • USSD (Mobile Authenticate only[2]) • SIM Applet • Smartphone app authenticator
MC_RQ01.2.5 Authentication requests	Mobile Connect Authentication should support server initiated as well as client initiated Authentication requests
MC_RQ01.2.6 Mobile Connect Authentication result	Mobile Connect must return one of the following: • a positive result • a negative result with an appropriate error code.
MC_RQ01.2.7 First and third party Authentication request.	The Mobile Connect Authentication products should support both first party and third party Authentication use cases (please see the Mobile Connect Product Definitions [9] for more details).

Table 1 : Authentication product requirements

[2] Support for LoA3 (Mobile Authenticate Plus) is not recommended given that the USSD mechanism relays the PIN in plain text

4 Mobile Connect Architecture Overview

The Mobile Connect logical architecture reuses many of the Operator assets and introduces new key components to deliver Mobile Connect services in accordance with the guiding principles outlined below:

- OpenID Connect as the protocol interface towards the Service Provider.
- The Service Provider only has to implement the OIDC protocol to connect to any serving Operator's ID GW.
- ID GW acts as a single point to maintain security, throttling, auditing, reporting, etc.
- Single point of contact for Operator Discovery using the API Exchange.
- Support for a variety of authenticators (pluggable approach); standard definition of LoA
- Authenticator selection based on the SP configuration during registration or based on the context of the request (e.g., LoA, SP client_Id + Policy Routing etc.)

The following diagram illustrates the key logical components that will be provided for, or impacted by, the deployment of Mobile Connect services. It should be noted that this is a logical architecture identifying the functional components. The actual implementation choices (e.g., mapping of functionality to physical components) is left to the Operator.

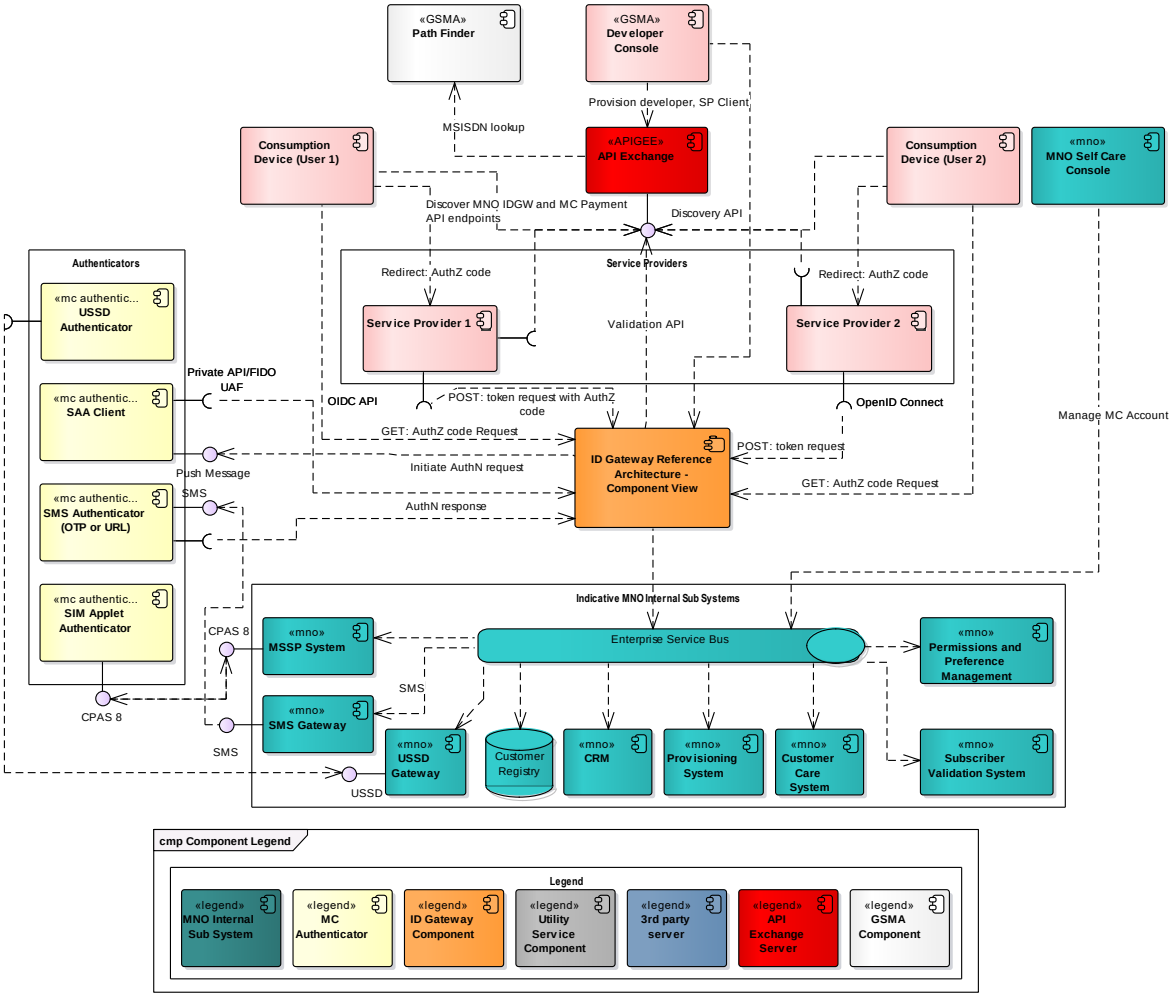

Figure 3: Mobile Connect architecture

Note: components such as Permissions and Preference Management are included for completeness and may not be an existing asset within the Operator deployment architecture.

Operator Onboarding to API Exchange

Mobile Connect uses the API Exchange as a central component to provide an SP with the endpoints of the serving Operator for a given user. The API Exchange exposes a simple REST based Discovery API ultimately returning “Discovered” resources in a JSON object.

The API Exchange utilises the mobile context to provide the Discovery service; i.e., it uses MCC/MNC or network IP addresses to discover the serving Operator. If the client initiates the discovery off net (i.e., not over the mobile network), the API presents the user a form to select the home Operator or enter their MSISDN. If an MSISDN is entered at the API Exchange, it is passed through the SP to the serving Operator in encrypted form to ensure privacy for the user.

The Operator needs to provide the MCC/MNC, IP Address range, OIDC endpoints, and certificate details to the API Exchange during the onboarding process.

Please find more details on the API Exchange Discovery process in the Discovery API specification [13]and in the implementation Guidelines Wiki.

5 Technical Requirements (ID GW)

The ID GW provides the following functionality:

- Acts as the entry point for Mobile Connect interactions.
- Exposes OpenID Connect and acts as the OpenID Connect provider.
- Provides ability to implement asymmetric decryption to decrypt the MSISDN, if present in the request.
- Manages interaction with the authenticators.
- Manages policy-based routing into authenticators.
- Manages the protocol mediation between the northbound OpenID Connect and the authenticator specific protocols using adaptors.
- Manages the multi-variant throttling of the incoming requests based on declarative policies.
- Manages the logging and auditing of interactions and operations.
- Manages the identification, authentication and authorisation of the OpenID Connect clients (Service Providers).
- Supports access to the UserInfo/PremiumInfo data sets from the data gateway or other sources.
- Provides external support for credential management with the API Exchange.

5.1 Overview of the end-end technical flow

The following diagram provides a technical walkthrough of the process of issuing an authentication request and returning a response to the requesting SP:

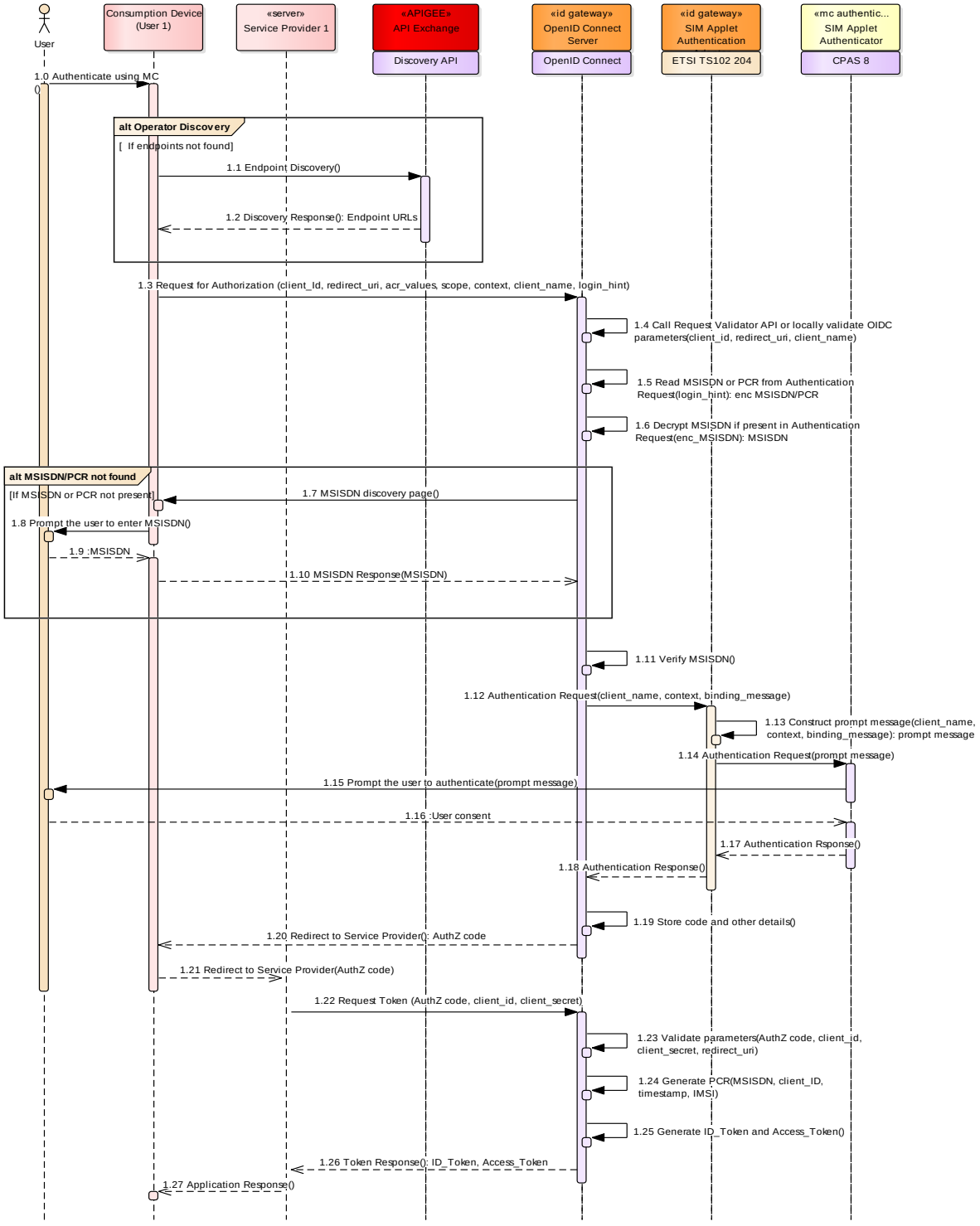

Figure 4: Technical flow - Authentication between the SP and ID GW

1. The user is consuming a service from the SP and needs to authenticate using Mobile Connect on their consumption device.

2. The consumption device's user agent initiates a discovery call with API Exchange to discover Operator's ID GW endpoint URL, encrypted MSISDN and SP client credentials.
3. The consumption device's user agent initiates OIDC authorization request with the Operator's ID GW, passing the user's identity (PCR/Encrypted MSISDN), required LoA (acr_values), SP client_id, SP short name (client_name) and optional binding message in the request object.
4. The ID GW optionally validates the incoming parameters locally or by invoking the Request Validator API in the API Exchange.
5. The ID GW retrieves the encrypted MSISDN or PCR from the `login_hint` attribute. [The ID GW prompts the user to enter the MSISDN if it is not present in the request].
6. The ID GW selects the appropriate authenticator for the LoA, user's device capability and Operator's policy.
7. The ID GW initiates user authentication challenge with the selected authenticator through the authenticator adapter, passing SP short name and optional binding message.
8. The authentication adapter constructs an appropriate authentication prompt message depending on the display capability of the underlying authenticator. Note the prompt action text is hardcoded and varies depending on the authenticator.
9. The authentication adapter invokes the appropriate API of the underlying authenticator to initiate user authentication passing the prompt message.
10. The authenticator prompts the user to authenticate by displaying the prompt message.
11. User authenticates.
12. The authenticator constructs appropriate assertion object and returns the response back to the ID GW via the authenticator adapter.
13. The ID GW stores the assertion and generates a new Authorization Code, returns the Authorization Code back to the SP server as a HTTP 302-redirect response through the consumption device's user agent.
14. The SP server extracts the Authorization Code from the redirected response.
15. The SP server sends a token request to the token endpoint at the ID GW passing the Authorization Code along with SP client_id and SP client_secret.
16. The ID GW validates the Authorization Code and returns the Access Token along with the ID Token JWT – containing the authentication context and PCR

5.2 User Registration

The Operator should provide a registration portal where the user can register for Mobile Connect. The ID GW must store the user's details in its repository in hashed format using a secure hash algorithm such as SHA256. The Operator should also provide support for full user lifecycle management where the user can update their personal details, including the PIN for example.

A standard repository can be used for data storage. Some examples that can be used are Lightweight Directory Access Protocol (LDAP), Rational Database Management System (RDBMS), Name/Value DB or non-relational databases such as MongoDB.

The following diagram represents the flow during MSISDN validation and user registration.

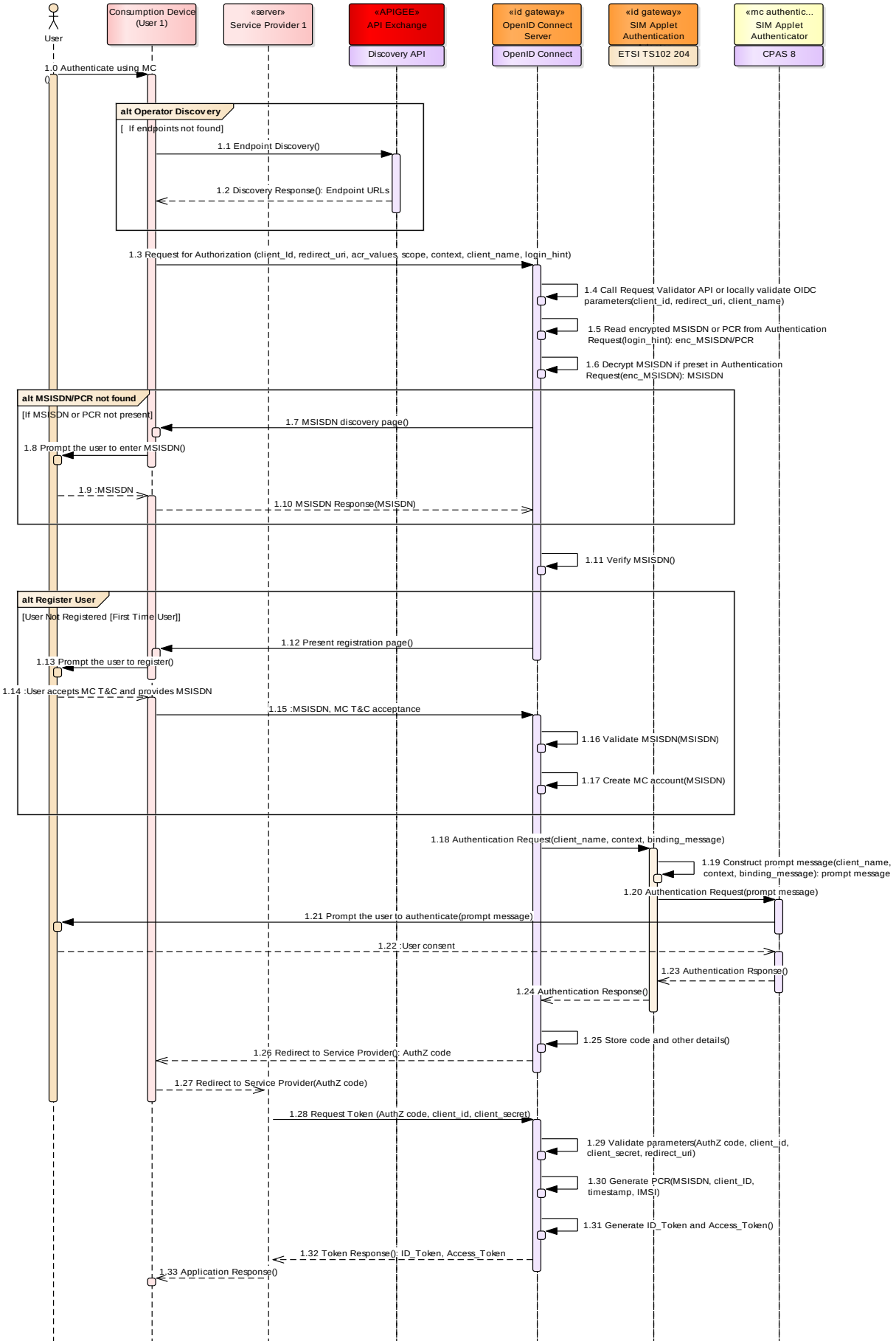
User
Consumption Device (User 1)
«server» Service Provider 1
«APIGEE» API Exchange
Discovery API
«id gateway» OpenID Connect Server
OpenID Connect
«id gateway» SIM Applet Authentication
ETSI TS102 204
«mc authentic... SIM Applet Authenticator
CPAS 8
1.0 Authenticate using MC ()
alt Operator Discovery
[If endpoints not found]
1.1 Endpoint Discovery()
1.2 Discovery Response(): Endpoint URLs
1.3 Request for Authorization (client_Id, redirect_uri, acr_values, scope, context, client_name, login_hint)
1.4 Call Request Validator API or locally validate OIDC parameters(client_id, redirect_uri, client_name)
1.5 Read encrypted MSISDN or PCR from Authentication Request(login_hint): enc_MSISDN/PCR
1.6 Decrypt MSISDN if preset in Authentication Request(enc_MSISDN): MSISDN
alt MSISDN/PCR not found
[If MSISDN or PCR not present]
1.7 MSISDN discovery page()
1.8 Prompt the user to enter MSISDN()
1.9 :MSISDN
1.10 MSISDN Response(MSISDN)
1.11 Verify MSISDN()
alt Register User
[User Not Registered [First Time User]]
1.12 Present registration page()
1.13 Prompt the user to register()
1.14 :User accepts MC T&C and provides MSISDN
1.15 :MSISDN, MC T&C acceptance
1.16 Validate MSISDN(MSISDN)
1.17 Create MC account(MSISDN)
1.18 Authentication Request(client_name, context, binding_message)
1.19 Construct prompt message(client_name, context, binding_message): prompt message
1.20 Authentication Request(prompt message)
1.21 Prompt the user to authenticate(prompt message)
1.22 :User consent
1.23 Authentication Rsponse()
1.24 Authentication Response()
1.25 Store code and other details()
1.26 Redirect to Service Provider(): AuthZ code
1.27 Redirect to Service Provider(AuthZ code)
1.28 Request Token (AuthZ code, client_id, client_secret)
1.29 Validate parameters(AuthZ code, client_id, client_secret, redirect_uri)
1.30 Generate PCR(MSISDN, client_ID, timestamp, IMSI)
1.31 Generate ID_Token and Access_Token()
1.32 Token Response(): ID_Token, Access_Token
1.33 Application Response()

Figure 5: Technical Flow - User registration

1. The user is consuming a service from the SP and needs to authenticate using Mobile Connect on their consumption device.
2. The consumption device's user agent initiates a discovery call with API Exchange to discover Operator's ID GW endpoint URL, encrypted MSISDN and SP client credentials.
3. The ID GW optionally validates the incoming parameters locally or by invoking the Request Validator API in the API Exchange.
4. The ID GW retrieves the encrypted MSISDN or PCR from the login_hint attribute. [The ID GW prompts the user to enter the MSISDN if it is not present in the request].
5. The ID GW validates the MSISDN and checks for the valid token for the corresponding MSISDN or PCR and registration status.
6. The ID GW prompts the user to register the MSISDN by accepting Mobile Connect T&Cs.
7. The ID GW validates the MSISDN before creating a MC account and storing the details securely.
8. The ID GW completes the authentication process by presenting authentication challenge to the user through the selected authenticator sub-system.
9. On successful authentication, the ID GW generates an Authorization Code and returns it as a query parameter in the redirect response to SP server.
10. The SP server can exchange the Authorization Code at the ID GW token endpoint for an ID Token.

5.3 Mobile Connect API

Mobile Connect utilises OpenID Connect as the primary API exposed to SPs for requesting Mobile Connect services. Operators should implement OpenID Connect in accordance with Mobile Connect Profile v1.2 [1].

Mobile Connect utilises the Authorization Code flow for SP requests as defined in the OpenID Connect specification [30]. The Authorization Code flow is a two-step process to obtain the Access Token and the ID Token.

Requirement	Description
MC_RQ02.2.1 API version	Mobile Connect Profile version 1.2 [1] MUST be used for the Mobile Connect Release 2 implementations and deployments
MC_RQ02.2.2 API Version parameter	The optional version parameter is introduced to keep track of different APIs. For more details, please refer to the OpenID Connect Mobile Connect profile [1].
MC_RQ02.2.3 Backward Compatibility	Existing ID GWs MUST retain support for OIDC requests based on OpenID Connect Mobile Connect Profile 1.1.

Table 2: OIDC API: technical requirements

5.3.1 Scopes

The Service Provider requests different Mobile Connect products by using the <scope> and <acr> parameters in the OIDC request:

Mobile Connect product	Scope value	LoA (acr_values)
Mobile Connect Authenticate	mc_authn	2
Mobile Connect Authenticate Plus	mc_authn	3
Mobile Connect Authorise	mc_authz	2
Mobile Connect Authorise Plus	mc_authz	3
Mobile Connect Phone Number	mc_identity_phonenumber	N/A
Mobile Connect Signup	mc_identity_signup	N/A
Mobile Connect National ID	mc_identity_nationalid	N/A

Table 3: Mobile Connect product scopes

Requirement	Description
MC_RQ02.2.4 OIDC scopes	The ID Gateway MUST support the following scope values for Mobile Connect Authentication: • "openid": as the mandatory scope for OpenID Connect • "openid mc_authn": as the explicit scope for Mobile Connect Authenticate products ○ If no explicit product scope is mentioned, then mc_authn is considered as the default scope value.
MC_RQ02.2.5 Authenticate product	The ID Gateway MUST identify the Mobile Connect Authenticate product with the scope ="openid mc_authn" + acr_values = "2" (LoA2) OR scope="openid" + acr_values = "2" (LoA2)
MC_RQ02.2.6 Authenticate Plus	The ID Gateway MUST identify the Mobile Connect Authenticate Plus product with the scope ="openid mc_authn" + acr_values = "3" (LoA3) OR scope="openid" + acr_values = "3" (LoA3)
MC_RQ02.2.7 Acr usage	The ID GW must only accept acr_values in the SP request for Authenticate, Authenticate Plus, Authorise and the Authorise Plus products; for the Identity Products, the acr_values [LoA] are implicit and managed at the ID GW as a policy

Table 4: OIDC scope: technical requirements

5.3.2 Authorisation Endpoint

The Service Provider prepares the authorisation request as per OpenID Connect Mobile Connect Profile v1.2 [1] and sends it to the Operator authorisation endpoint.

5.3.3 Client Credential Validation

The IDP must validate the client credential as well as the redirect_uri, scopes before authenticating the end user and returning the ID Token and Access Token. There are three ways the Operator can validate the SP details.

SP Register Locally to Operator

The Service Provider can register their application on the registration portal provided by the Operator. The Operator can store the Service Provider's details including the client_id, client_secret, registered scopes, redirect_uri, etc. in its local repository.

The Operator can use these details from its local repository to validate and authenticate the client during the authorisation and token request.

Operator Receives Client Details Via Offline Process from API Exchange

The Service Provider can integrate the Operator with the API Exchange and the API Exchange publish the Service Provider details to the Operator via an offline batch process. The Operator stores these details in its database and uses them to validate the client credentials and other attributes.

Operator Uses "Request Validator" API from Exchange

The Operator can also use the "Request Validator" API provided by the API Exchange to validate client credentials. The Operator can pass the client_id and client_secret received from the SP request to make a call on "Request Validation API". The "Request Validator" API returns other metadata including redirect_uri, the registered scopes, etc. in the response. The Operator can use these details to verify the client's redirect_uri, scope etc. and store it locally for future reference.

In summary:

- The Operator must read the client credential from the HTTP Header for client authentication.
- The behaviour will be different (undefined) from Operator to Operator if credentials are passed in HTTP Header and payload.
- The Operator may reject or process the request based on their own implementations if the payload contains credentials or any other parameters out of specification
- The Operator must reject the request if there are no client credentials in the HTTP Header

Once the user has authenticated (and optionally authorised a transaction or provided consent for attribute sharing), an Authorisation Code is returned to the URI value specified by the SP in the `redirect_uri`; response parameters are included as query parameters encoded using application/x-www-form-urlencoded.

If user authentication fails or the user does not provide consent, the Operator should return an authentication error in the response as per OAuth 2.0 using the codes defined in the Mobile Connect Technical Reference [5].

Login_hint_token is an optional parameter to transport the encrypted MSISDN. The mandatory parameter to transport the MSISDN or encrypted MSISDN is login_hint.

Redirect_uri Validation

The Service Provider registered with the Operator provides details including the `redirect_uri` that will be used to send the Authorization Code and Tokens. The Service Provider passes the `redirect_uri` as a parameter in the Authorization request and Token request. The Operator needs to validate this `redirect_uri` before authenticating the end user or returning the token.

The API Exchange can pass all the Service Provider details including the `redirect_uri`, scopes, etc. via an offline process to all participating Operators or it can return the `redirect_uri` in the response from the "Request Validator" API. The Operator should reject any authentication or token request if the `redirect_uri` does not match with the registered one. The Operator should also validate the `redirect_uri` in the token request and match it with the `redirect_uri` passed during the authorisation request.

The `redirect_uri` must match exactly with one of the `redirect_uri` pre-registered with the Operator with the matching performed as simple string comparison. The `redirect_uri` should use the HTTPS flow (although it may use the HTTP url as well).

The redirection endpoint should require the use of TLS when the requested response type is a code or token. If the TLS is not available, the authorisation server should warn the resource owner about the insecure redirection.

5.3.4 Token Endpoint

The ID GW will generate an OIDC response once a user is successfully authenticated and has provided consent for the request. The OIDC response contains an `Access_Token` and an `ID_Token`.

The ID Token (created and returned by the Operator to the SP) is an extension to the OAuth 2.0 token (Access Token) to provide the claims for Authentication Context/Event, represented as a JWT (http://tools.ietf.org/html/draft-ietf-oauth-json-web-token-14). For more details on the claims included within the ID Token please see the OpenID Connect Mobile Connect Profile v1.2 [1].

The Operator must implement a token endpoint that will return the ID Token and Access Token. Communication with the Token endpoint must use TLS (HTTPS). The request encoding used is application/x-www-form-urlencoded.

The Service Provider must pass the Authorization Code along with client credentials to the token endpoint url to obtain the ID Token and Access Token.

The Operator must validate the client credentials before returning the Access Token and ID Token to the SP.

The token response should be in accordance with OAuth 2.0 and should be encoded in UTF-8. Further details on the response parameters can be found in the OpenID Connect Mobile Connect Profile v1.2 [1].

ID Token

The `ID_Token` is a security token that contains claims related to an authenticated user set by an authorisation server. The authorisation server within the ID GW digitally signs the `ID_Token` and passes the JSON Web Signature (JWS) header.

The JWS represents digitally signed or MACed content using JSON data structure and base64url encoding. The JWS represents these logical values.

- JWS Header: JSON object containing the parameters describing the cryptographic operation and the parameter employed.
- JWS Payload: Message content to be secured.
- JWS Signature: Digital signature over the JWS header and payload.

The JSON Header consists of algorithms used to sign the JSON object that may be selected by the Operator based on the available infrastructure. The client has to validate the `ID_Token` before granting access. The `alg` header parameter contains the algorithm used in the JWS. The Operator may validate the ID Token based on their own implementation. This will not be part of the Software Development Kit (SDK) provided by Mobile Connect.

To indicate the details of the authenticator, the `amr` parameter can be used. The `amr` will also be used to provide other authenticator details.

5.4 Authenticator Selection

One of the key architecture principles of Mobile Connect is the Pluggable Authenticator principle. To achieve this, the authenticators must be selected dynamically based on the configured policies using inputs such as the LoA indicated in the OIDC request by the Service Provider, the available authenticator, the user/device eligibility, etc.

Some key points to implement the authenticator selection mechanism:

- The authenticators should be integrated with the ID GW based on an adaptor pattern.
- The specific information of the authenticator implementation should be confined within the authenticator adaptor as much as possible.
  - This provides a loose coupling between the authenticator implementation and the Mobile Connect system and drives the Pluggable Authenticator principle.
- The policy engine should route the authentication request to the adaptor.
  - The adaptor may then use authenticator specific interaction model to invoke and communicate.
  - The adaptor may use more than one call to invoke the authenticator (composition).
- The various factors (inputs) used for the authentication selection should be stored in a configuration database.
  - It should be possible to manage and administer the lifecycle of the factors.
  - Some of the inputs that will be used for the authenticator selection can be LoA, context request, SP `client_id`, user profiles, etc.
- The authenticator selection mechanism should validate and ensure that the request contains enough information to invoke the authenticator and all authenticator specific details are available.

- The authenticator selection mechanism should convert and add any specific information needed for the authenticator.

The following are some examples to implement the authenticator selection policy:

- The SP identifier can be used to map one LoA with different types of authenticators: e.g., Client ID 1 + LoA3 can be mapped with Authenticator 1 and Client ID 2 + LoA3 can be mapped to Authenticator 2. These configuration policies will be stored in the Operator's repository.
- The authenticator can be selected based on the service context request, SP preference, user profiles, etc.
- A fallback mechanism should be configured and authenticators can be selected based on the LoA order. The Service Provider may pass multiple LoA in `acr_values` parameter in order of authenticator preference.
- The Operator may authenticate with a lower LoA if the requested LoA is not available. The authenticator used for authentication will be passed in the `amr` parameter in the OIDC response. The Service Provider will take the decision to provide the service or show an error message to the user based on their policy. For example, a Service Provider needs LoA3 SIM applet authentication and the user is authenticated with LoA2 USSD. The Operator will pass the USSD authenticator in the `amr` parameter. The Service Provider will decide whether to grant the full service, limited service or error to the user.

The Service Provider can use the amr attribute to signal their preferred authenticator type; e.g. the Service Provider can pass amr=SIM-OK in the authentication request. The Operator will read this attribute and will select the SIM-OK authenticator if available for the target user.

The ID GW policy engine must have a configuration to map the LoA to the appropriate authenticator. This configuration should be stored in a repository and it should be possible to manage the CRUD (Create, Read Update, and Delete) lifecycle of the mapping configuration. For performance reasons, the configuration can be cached in memory to optimise the reading of the mapping by the policy execution engine.

Authenticator	LoA
Seamless authenticators	LoA 2
HTTP Header enrichment based authenticator	LoA 2
MO SMS based authenticator	LoA 2
Device agent/library based authenticator	LoA 2
NI USSD based authenticator	LoA 2 [OK][3]
SIM applet authenticator	LoA 2 [OK], LoA 3 [PIN]
SMS + URL based authenticator	LoA 2

[3] LoA3 is possible but as the PIN would be transported over the mobile network in plain text, use of NI USSD for LoA3 is not recommended from a security perspective

Smartphone App Authenticator	LoA 2 [OK], LoA 3 [PIN or biometrics]

Table 2: Mapping of authenticators to LoA indicated

5.5 Pseudonymous Customer Reference (PCR)

The ID GW will need to generate a Pseudonymous Customer Reference to be included in the OIDC response that identifies an authenticated user uniquely to a Service Provider. The following principles relate to the creation and use of PCRs.

- The PCR must be unique per user/SP application group pairing inside the specific ID GW perimeter.
- A PCR is dependent on the end user's current Operator (Operator 1). The creation is specific to an Operator (to support uniqueness).
- The implementation guide will recommend the size of the PCR to ensure that a PCR can be imported into a Mobile Connect account.
- PCRs are used by Service Providers to link an SP user account with a Mobile Connect account inside Operator ID GW perimeter; data cannot be extracted from a PCR.
- If an end user changes the MSISDN or the International Mobile Subscriber Identity (IMSI), the PCRs inside Operator ID GW perimeter should not be regenerated.
  - In other words, if the MSISDN is not anymore allocated to the end user for any reason (end of contract, same contract owner but new end user…), link between end user and MSISDN must be deleted. At the same time, the PCR may still exist and is still be attached to the end user. The PCR can be used later, for the same end user, with a further MSISDN or with others MSISDNs.
  -

The specific technical requirements for implementation of the PCR are as follows:

Requirement	Description
MC_RQ02.2.8 New PCR	The ID GW MUST generate the new PCR format for Mobile Connect Release 2, as a Globally Unique Identifier (GUID) inside its perimeter.
MC_RQ02.2.9 PCR format	The GUID MAY be UUID format as specified in RFC 4122[9] with these Specifics: • The byte order of GUID MAY be in the format 8-4-4-12 digits hex string (e.g. **5f90512d-972d-4def-bf90-9ef0ef2e5d2d)**. • The layout and Byte order MAY match section 4.1.2 in RFC 4122. The GUID MAY be version-4 UUID, which is randomly or pseudo-randomly generated.
MC_RQ02.2.10 PCR association	ID GW MUST associate the PCR with the Sector Identifier (HOST part of the redirect_uri registered by the Service Provider) and Mobile Connect account id (MSISDN)
MC_RQ02.2.11 PCR consistency	For a given Mobile Connect user (MSISDN), ID GW MUST generate the same PCR if the Sector ID is same, as registered by the Service Provider
MC_RQ02.2.12 SP PCR	The Service Provider must be able to acquire a PCR through a successful Mobile Connect Authentication.
MC_RQ02.2.13 Untrusted SP MSISDN	Untrusted Service Providers must be able to request using encrypted MSISDN or using PCR.

	And must not be able to request Mobile Connect Authentication or Authorisation using plain MSISDN.

Table 5: PCR: technical requirements

5.6 Trusted Service Providers

Trusted Service Providers (TSP) are allowed to pass the MSISDN or Encrypted MSISDN in the OIDC Authorisation request. The SP can be marked as a TSP in the ID GW based on a business process.

Requirement	Description
MC_RQ02.2.14 Trusted/Normal Service Provider	The ID GW must support two different types of Service Providers. • Trusted Service Provider • Normal Service Provider The ID GW must maintain a Service Provider type to distinguish between Normal, Trusted and any future classification of Service Provider (i.e., the field must not be Boolean for simply flagging Trusted status).
MC_RQ02.2.15 SP trust status	The Service Provider must be able to request Mobile Connect Authorisation using login_hint (OR) login_hint_token parameter based on their "trust" status. These parameters are used to specify MSISDN/ encrypted MSISDN or PCR. For more information on the login_hint and login_hint_token parameters please see the Mobile Connect Profile [1].
MC_RQ02.2.16 Trusted SP MSISDN	Trusted Service Providers can request Mobile Connect Authorisation using plain MSISDN or encrypted MSISDN in the login_hint (Or) login_hint_token parameter
MC_RQ02.2.17 Unregistered SP	If the SP is not marked as a TSP at the ID GW and an MSISDN is supplied in the login_hint, the request must be rejected and error should be returned.

Table 6: Trusted Service Providers: technical requirements

5.7 MSISDN Decryption

The API Exchange can prompt the user to provide their MSISDN to perform Discovery if it does not find the MCC/MNC or IP address information in the request. The API Exchange will encrypt this MSISDN using the serving Operator public key and pass the encrypted MSISDN to the Service Provider to use in their Mobile Connect API request towards the serving Operator.

Please refer to the Implementation Guidelines Wiki for more information on MSISDN encryption.

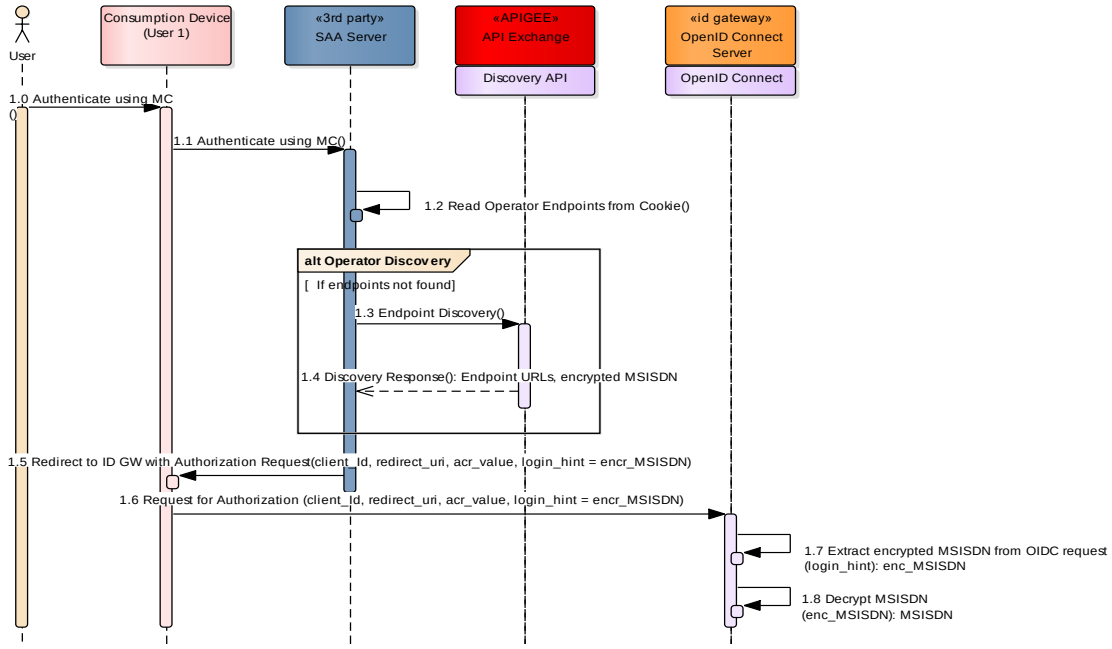

Figure 6: Technical flow - MSISDN decryption

MSISDN decryption steps

1. The Service Provider passes the encrypted MSISDN in the OIDC request within the `login_hint` parameter. The parameter can be a URL encoded.
2. Following is an example of the encrypted MSISDN being passed:
   - **in the OIDC request**: login_hint=ENCR_MSISDN:5be…..a11c
   - **or with URL encoding**: login_hint=ENCR_MSISDN%3A0b…..5a2e
3. The serving Operator recognises the input of the encrypted MSISDN and decodes the base64 encoded data.
4. The serving Operator applies their private key to decode the RSA coded data.
5. The serving Operator then extracts the initial (numeric) portion of the decrypted data as the MSISDN separated by (|) pipe and uses this for any relevant purpose in API services/user sign-in.

5.8 Device Interlock Using a Binding Message

Mobile Connect can be accessed in an off channel mode when the consumption device is a PC and the authentication device is a mobile device. To mitigate the risk of phishing attacks, Mobile Connect can display an alphanumeric binding message on both devices. The user does not have to transfer this binding message, it is just for comparison on both devices to reassure the user that the approval they are providing via the authentication device corresponds to the action they are invoking on the consumption device.

The binding message is displayed to the user in the authentication/authorisation prompt and returned as part of the OIDC response within the `displayed_data` claim for auditing purposes:

displayed_data:

Text that is displayed in the authentication/authorisation prompt and is returned back to the Service Provider as an ID Token claim within OpenID Connect Mobile Connect Profile 1.2

The displayed data / prompt contains the following information:

`client_name`[4] + `context`[5] + `binding_message`

5.9 Server-initiated calls

Requirement	Description
MC_RQ02.2.18 Server-based invocation support	The ID GW must implement support for server-based invocation of Mobile Connect services. The SP Server, which is initiating the Mobile Connect request to the ID GW MUST possess the MSISDN or the PCR of a Mobile Connect user.
MC_RQ02.2.19 Server- based invocation	If prompt= "mobile" is included in the OIDC Authorisation Request, the ID GW MUST treat the request as a server-initiated request without a user agent.
MC_RQ02.2.20 Server- based invocation HTTP stack	The ID GW MUST use a non-blocking I/O HTTP stack to support server-based invocation. It is assumed that the SP Server component has an HTTP stack which can support HTTP redirects and that the timeout for the Authorisation HTTP request is set at a high value
MC_RQ02.2.21 Server- based invocation holding pages	The ID GW MUST not return any HTML page in response to the Authorisation request, e.g. the holding page as this is a server initiated request
MC_RQ02.2.22 Use of TLS	Communication with the ID Gateway endpoints MUST use TLS.
MC_RQ02.2.23 GET and POST support	The ID GW MUST accept the Server initiated OIDC request either through GET or POST.
MC_RQ02.2.24 Redirect	The ID GW MUST return an Authorization code to the Server using the redirect at the registered redirect_uri.

Table 7: Server-initiated calls: technical requirements

5.10 Security Requirements

As identified by the GSMA SFRA group, (Security & Fraud Risk Assessment), Mobile Connect may include various threats on the Operator side such as DDoS attack or data leak. The user may suffer social media attacks, OS incompatible bugs, malware, spam, etc. A Service Provider may also suffer from DDoS and

[4] SP short name

[5] For Mobile Connect Authorization products

phishing attacks. This section provides more details on Operator Side Security and Fraud Mitigation.

DDoS Attack

The Operator should provide anti-DDoS solutions, such as an IPS system, to protect the ID GW and other components. The Operator should also consider DDoS mitigation solutions to ensure resilience of service. This should be applied on all the in-scopes systems and should be coupled with effective incident response capabilities.

Data Leak

The Operator should provide data protection such as data encryption and access control mechanisms to keep the user's personal information safe. Tools such as an Intrusion Detection System and/or Intrusion Prevention should also be considered along with effective monitoring, detection and incident management.

Mass Spam and Target Spam

The Operator should provide antispam solutions by using second attributes such as location, or using alias input such as MSISDN. Meanwhile the Identity Gateway should have the ability to detect abnormal patterns.

SIM Cloning

A SIM card may be cloned and despite off network countermeasures this may in itself allow for fraudulent registration of the service. The Operator should implement SIM cloning detection capabilities such as the use of volume, value and velocity checking within their fraud management system.

MSISDN Recycling

An abuse of MSISDN recycle/purge processes may create an opportunity for fraudulent registration services. The Operator should implement an internal audit process to tackle such issues.

OTA

The Operator may use OTA campaigns to distribute updated SIM applications. A specific fraud risk includes the possibility to download the application to SIM profiles with known OTA vulnerabilities. OTA campaigns must be constructed to identify and reject downloads to SIMs with known vulnerable profiles.

SMS Gateway and SMSC

An attacker sends spoofed SMS to customers using SMS Gateways or SMCS as part of fake authentication processes to socially engineer the customer to believe they have authenticated to legitimate sites. The Operator should take precautions to unambiguously identify the SMS source.

5.11 ID GW requirements summary

The following table summarises all the requirements identified in the previous subsections:

Requirement	Description
MC_RQ02.2.1 API version	Mobile Connect Profile version 1.2 [1] MUST be used for the Mobile Connect Release 2 implementations and deployments
MC_RQ02.2.2 API Version parameter	The optional version parameter is introduced to keep track of different APIs. For more details, please refer to the OpenID Connect Mobile Connect profile [1].
MC_RQ02.2.3 Backward Compatibility	Existing ID GWs MUST retain support for OIDC requests based on OpenID Connect Mobile Connect Profile 1.1.
MC_RQ02.2.4 OIDC scopes	The ID Gateway MUST support the following scope values for Mobile Connect Authentication: • "openid": as the mandatory scope for OpenID Connect • "openid mc_authn": as the explicit scope for Mobile Connect Authenticate products ○ If no explicit product scope is mentioned, then mc_authn is considered as the default scope value.
MC_RQ02.2.5 Authenticate product	The ID Gateway MUST identify the Mobile Connect Authenticate product with the scope ="openid mc_authn" + acr_values = "2" (LoA2) OR scope="openid" + acr_values = "2" (LoA2)
MC_RQ02.2.6 Authenticate Plus	The ID Gateway MUST identify the Mobile Connect Authenticate Plus product with the scope ="openid mc_authn" + acr_values = "3" (LoA3) OR scope="openid" + acr_values = "3" (LoA3)
MC_RQ02.2.7 Acr usage	The ID GW must only accept acr_values in the SP request for Authenticate, Authenticate Plus, Authorise and the Authorise Plus products; for the Identity Products, the acr_values [LoA] are implicit and managed at the ID GW as a policy
MC_RQ02.2.8 New PCR	The ID GW MUST generate the new PCR format for Mobile Connect Release 2, as a Globally Unique Identifier (GUID) inside its parameter.
MC_RQ02.2.9 PCR format	The GUID MAY be UUID format as specified in RFC 4122[9] including these Specifics: • The byte order of GUID MAY be in the format 8-4-4-12 digits hex string (e.g. **5f90512d-972d-4def-bf90-9ef0ef2e5d2d)**. • The layout and Byte order MAY match section 4.1.2 in RFC 4122. The GUID MAY be version-4 UUID, which is randomly or pseudo-randomly generated.
MC_RQ02.2.10 PCR association	ID GW MUST associate the PCR with the Sector Identifier (HOST part of the redirect_uri registered by the Service Provider) and Mobile Connect account id (MSISDN)
MC_RQ02.2.11 PCR consistency	For a given Mobile Connect user (MSISDN), ID GW MUST generate the same PCR if the Sector ID is same, as registered by the Service Provider
MC_RQ02.2.12 SP PCR	The Service Provider must be able to acquire a PCR through a successful Mobile Connect Authentication.
MC_RQ02.2.13 Untrusted SP MSISDN	Untrusted Service Providers must be able to request using encrypted MSISDN or using PCR. And must not be able to request Mobile Connect Authentication or Authorisation using plain MSISDN.
MC_RQ02.2.14 Trusted/Normal Service Provider	The ID GW must support two different types of Service Providers. • Trusted Service Provider

	• Normal Service Provider The ID GW must maintain a Service Provider type to distinguish between Normal, Trusted and any future classification of Service Provider (i.e., the field must not be Boolean for simply flagging Trusted status).
MC_RQ02.2.15 SP trust status	The Service Provider must be able to request Mobile Connect Authorisation using login_hint (OR) login_hint_token parameter based on their "trust" status. These parameters are used to specify MSISDN/ encrypted MSISDN or PCR. For more information on the login_hint and login_hint_token parameters please see the Mobile Connect Profile [1].
MC_RQ02.2.16 Trusted SP MSISDN	Trusted Service Providers can request Mobile Connect Authorisation using plain MSISDN or encrypted MSISDN in the login_hint (Or) login_hint_token parameter
MC_RQ02.2.17 Unregistered SP	If the SP is not marked as a TSP at the ID GW and an MSISDN is supplied in the login_hint, the request must be rejected and error should be returned.
MC_RQ02.2.18 Server-based invocation support	The ID GW must implement support for server-based invocation of Mobile Connect services. The SP Server, which is initiating the Mobile Connect request to the ID GW MUST possess the MSISDN or the PCR of a Mobile Connect user.
MC_RQ02.2.19 Server- based invocation	If prompt= "mobile" is included in the OIDC Authorisation Request, the ID GW MUST treat the request as a server-initiated request without a user agent.
MC_RQ02.2.20 Server- based invocation HTTP stack	The ID GW MUST use a non-blocking I/O HTTP stack to support server-based invocation. It is assumed that the SP Server component has an HTTP stack which can support HTTP redirects and that the timeout for the Authorisation HTTP request is set at a high value
MC_RQ02.2.21 Server- based invocation holding pages	The ID GW MUST not return any HTML page in response to the Authorisation request, e.g. the holding page as this is a server initiated request
MC_RQ02.2.22 Use of TLS	Communication with the ID Gateway endpoints MUST use TLS.
MC_RQ02.2.23 GET and POST support	The ID GW MUST accept the Server initiated OIDC request either through GET or POST.
MC_RQ02.2.24 Redirect	The ID GW MUST return an Authorization code to the Server using the redirect at the registered redirect_uri.
MC_RQ02.2.25 Profile V1.2 support	The ID GW must support the <version> parameter received within the OIDC Authorisation request. It must support both mc_v1.2 and mc_v1.1
MC_RQ02.2.26 Request parameter checking	The ID GW must check the combination of <version> parameter (if exists), <scope> and <acr> values to identify the required Mobile Connect Services.
MC_RQ02.2.27 Mobile Connect product SP check	The ID GW must check that the SP is subscribed to the requested Mobile Connect product service (by checking the requested scope value1) before serving the request
MC_RQ02.2.28 Unknown scopes	The ID GW must ignore any <scope> values which are unknown.
MC_RQ02.2.29 User state	The ID GW must maintain various Mobile Connect account states as mentioned in [11].
MC_RQ02.2.30 Active state	The ID GW must serve OIDC requests if and only if the Mobile Connect user's account is in an "active" state.
MC_RQ02.2.31 Reject states	The ID GW must reject Mobile Connect Authorisation OIDC requests for all other Mobile Connect account states (suspended, deleted, not available).

Table 8: ID GW: technical requirements summary

5.12 Operator Requirements Summary

Requirement	Description
MC_RQ02.3.1 OIDC endpoints	Operators need to expose the OIDC endpoint from their ID GW components of Mobile Connect.
MC_RQ02.3.2 API exchange registration	Operators should register with the API Exchange and provide all details including the OIDC endpoint URLs, the IP address range, certificates or Mobile Country Code (MCC)/Mobile Network Code (MNC) details.
MC_RQ02.3.3 Pluggable Authenticators	Operators have to implement a pluggable authenticator mechanism to support various authenticators as per requirements.
MC_RQ02.3.4 SIM applet authenticators	For those Operators deploying a SIM applet authenticator, they must develop an authenticator adapter from the ID GW to their MSSP compliant with ETSI TS 102. Operators may also need to provide over-the-air (OTA) capabilities for distributing the SIM applet.
MC_RQ02.3.5 Security	Operators should take care of all security aspects during the ID GW implementation, including Transportation Layer Security (TLS) or Message Layer Security (MLS), as per the Mobile Connect architecture.
MC_RQ02.3.6 Mobile Connect portal	Operators should also provide a portal for end users to register and accept the Terms and Conditions for Mobile Connect.
MC_RQ02.3.7 Account Status	Operators should integrate their ID GW to their CRM system to synchronise the customer account status as described in the Mobile Connect Lifecycle events [7] and Mobile Connect Life Cycle Technical Solutions [12]

Table 9: Operator requirements summary

Annex A Document Management

A.1 Document History

Version	Date	Brief Description of Change	Approval Authority	Editor / Company
1.0	25/5/2016	New Document	PDATA/PSMC	David Pollington/GSMA
1.1	12/05/2017	Transfer of PRD from Personal Data		Nick Cheung/GSMA
1.2	27/04/2022	New minor version	Internet Group	Yolanda Sanz, GSMA

A.2 Other Information

Type	Description
Document Owner	IDG
Editor / Company	Yolanda Sanz/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.