IDY.25 Mobile Connect Authorise PKI Definition and Technical Requirements

Version 1.0

06 December 2022

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

Security Classification: Non-confidential

Copyright Notice

Disclaimer

Antitrust Notice

Review Log (to be completed by GSMA Support Staff)			
Workflow Step	***Document Review Comments***	***GSMA Support Staff Name***	***Comments Date***
Step 1: Change Request Creation (no comments required)			
Step 2: Document Quality and/or Legal Review			
Document Quality Team	Comments and amendments for review.	Donna Mackay/GSMA	28/032019
Legal Review	INSERT COMMENTS HERE Please enter details for the Legal Review Confirm Legal feedback Record any issues, actions and key decisions	GSMA Support Staff Name	DD/MM/YY
Step 3: Formal Review			
Group(s)/Project(s) Review(s) Comments and Feedback	Approved offline CPAS	Gautam Hazari	17/12/2019
Step 4: Formal Approval(s)			
Group(s)/Project(s) Approval(s) Comments and Feedback	Submitted to TG for final approval	Gautam Hazari	17/12/2019

Table of Contents

1 Introduction

1.1 Overview

Mobile Connect is a worldwide initiative by mobile operators to bring a wide portfolio of identity services to market that enable SPs and Users to transact with one-another more securely through strong authentication, authorisation and exchange of attributes, subject to User consent.

The Mobile Connect architecture consists of a Core framework around which additional components can be added to support the different Mobile Connect services. The Core framework is based upon the OpenID Connect (OIDC) protocol [1] and allows Users to be identified by their MSISDN (or a related Pseudonymous Customer Reference) to enable authorisation on their mobile device.

This document details the Mobile Connect Authorise PKI service which offers the ability for a User to authorise or approve transactions presented to them by a SP via the User's mobile device and to enable the authorisation response[1] to be digitally signed by the User's private key for non-repudiation purposes. In doing so, the service combines two-factor authorisation i.e. "*Something I have*" (device) and "*Something I know or am*" (PIN or biometric) as per the Mobile Connect Authorise service with the addition of PKI thus offering an authorisation service that is compatible with eIDAS[2] High level and based on ISO 29115 LoA4.

The Mobile Connect Authorise PKI service addresses customer demand across several countries for an authorisation service that is underpinned by mobile-based PKI, hence bringing additional convenience compared to existing PKI solutions using smartcards & smartcard readers, for example.

1.2 Scope of the document

In Scope	Out of Scope
• Definition & specification of an authorisation service that combines two factor authentication i.e. "*Something I have*" (device) and "*Something I know*"' (PIN) with PKI and User-signed responses for non-repudiation purposes	• Detailed Privacy and Trust Principles • UI/UX guidelines • Service provider / developer implementation guidelines

1.3 Audience

The target audience of this document are the product managers and service/technical departments and Operators who are considering deploying the Mobile Connect Authorise PKI service.

Readers of this document are expected to have familiarity with Mobile Connect and some knowledge of the technical architecture and Mobile Connect Core framework technical requirements.

[1] The signed response could be the data that is displayed on the mobile device or a SP provided data to be signed.

[2] EU Regulation 910/2014 of 23 July 2014 on electronic identification

1.4 Relationship to Other Mobile Connect Documentation

This document details the Mobile Connect Authorise PKI service and its usage including requirements (building on the Mobile Connect Core framework) and the relevant technical parameters for the service.

The Mobile Connect Technical Architecture and Core Requirements document [5] describes the Mobile Connect Architecture in more detail and includes the core requirements and the specification of elements for Mobile Connect that are generic to all Mobile Connect services and modes of operation.

The Mobile Connect Device-Initiated OIDC Profile [6] and the Mobile Connect Server-Initiated OIDC Profile [7] specify the Mobile Connect APIs which provide details for OIDC Authorization Requests and Responses, and Token retrieval including examples and error codes.

The Mobile Connect Technical Architecture and Core Requirements document along with the Mobile Connect Device-Initiated OIDC Profile and Server-Initiated OIDC Profile together define the Mobile Connect Core framework upon which all services are built.

The Mobile Connect Technical Overview document [4] provides a high-level description of Mobile Connect and how it works. It also includes a master list of abbreviations and terminology used within the Mobile Connect Documentation set and a map of that documentation set. It serves as a starting point for understanding how Mobile Connect works and also references the relevant documents for the reader to obtain further detail.

1.5 Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [3].

1.6 Terminology & Definitions

Mobile Connect specifications and related documents make use of terms that are defined by the OpenID Connect Core Specification [1] and extended in the OIDF CIBA (Client Initiated Backchannel Authentication Flow) [2].

The Mobile Connect Technical Overview document [4] defines relevant terms that are used within the Mobile Connect Specifications and interprets terminology from source standards in Mobile Connect terms. It also includes a list of abbreviations.

Due to potential confusion with OIDC and OAuth 2.0 terminology, the initial Mobile Connect service request which underpins Mobile Connect Authentication, Authorisation and User consent associated with attribute services, is referred to as an OIDC Authorization Request[3] (spelled with a ‘z’) throughout this document.

[3] In OAuth2.0 the initial request is referred to as an “Authorization Request”, whereas in OIDC it is referred to as an “Authentication Request”. Mobile Connect offers several services including MC

1.7 Abbreviations

Abbreviation	Description
CA	Certificate Authority
DTBD	Data To Be Displayed
DTBS	Data To Be Signed
GDPR	General Data Protection Regulation
ID GW	Identity Gateway
ID Token	Identity Token
MC	Mobile Connect
MSISDN	Mobile Station International Subscriber Directory Number
MSSP	Mobile Signature Service Provider
OIDC	Open ID Connect Protocol
OTA	Over The Air
PKI	Public Key Infrastructure
RA	Registration Authority
SIM	Subscriber Identity Module
SP	Service Provider
TEE	Trusted Execution Environment
wPKI	Wireless Public Key Infrastructure

1.8 References

Ref	Doc Number	Title
[1]	OpenID Connect Core Specification	"An interoperable authentication protocol based on the OAuth 2.0 family of specifications" available at https://openid.net/specs/openid-connect-core-1_0.html
[2]	OIDF CIBA	OpenID Connect MODRNA Client Initiated Backchannel Authentication Flow 1.0 https://openid.net/specs/openid-connect-modrna-client-initiated-backchannel-authentication-1_0.html
[3]	RFC 2119	"Keywords for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997. Available at https://tools.ietf.org/html/rfc2119
[4]	IDY.05	Mobile Connect Technical Overview
[5]	IDY.04	Mobile Connect Technical Architecture and Core Requirements
[6]	IDY.01	Mobile Connect Device-Initiated OIDC Profile
[7]	IDY.02	Mobile Connect Server-Initiated OIDC Profile

Authentication and MC Authorisation, hence MC specifications have adopted the term "OIDC Authorization Request" to describe this initial service request in the protocol flow.

2 Mobile Connect Authorise PKI service overview & functional requirements

Mobile Connect Authorise PKI combines two-factor authorisation[4] with the introduction of PKI and User's digital certificates [X.509] to increase robustness and provide non-repudiation (the User signing their response with their digital certificate) hence ensuring that no party (User, MNO or SP) can later deny the existence or execution of that Mobile Connect service transaction; in essence, it ensures that Users can be held accountable for their actions.

2.1 Use Case Examples

Mobile Connect Authorise PKI supports a range of use cases by supporting non-repudiation between SPs and Users. A few examples are listed below.

Service	Example Use Cases
Mobile Connect Authorise PKI	• Accessing a website through login, with non-repudiation • Verification of high value or risky transactions with non-repudiation • Secure Health record transfer approval with non-repudiation • Legal document signing using User's signature

Table 1: Use Case Examples

2.2 MC Authorise PKI Variants

The Mobile Connect (MC) Authorise PKI service has been designed to offer a number of different modes to support different use cases and deployment models.

2.2.1 Variant 1: PKI Authorisation without returning signed response

- The User's response is signed using the User's private key on the authentication device but the ID GW only returns the result of whether or not the User authorised the SPs request as per the MC Authorise/Plus service.
- The data to be signed uses the SP provided text in the initial Mobile Connect service request.
- The User's signed response MUST be logged so that it can be queried separately within a business process; e.g., for auditing and dispute resolution purposes

2.2.2 Variant 2: PKI Authorisation with signed response returned to SP

- The SP submits a request for authorisation and the specific 'data' that they would like to be signed by the User.

[4] i.e., authentication with two-factors "Something I have" (device) and "Something I know or am"' (PIN or Biometric) before User authorisation of the SP request

- The data to be signed can either be the SP authorisation request text, a digest[5], a hash[6] of data/document or whatever is relevant for the target use case.
- The data is signed using the User's private key on the authentication device to provide an electronic signature.
- The service returns the result of whether or not the User authorised the SPs request along with the signed data; three subtypes are included to support different methods of validating the User signature by the SP.

2.2.2.1 Variant 2a:

The SP already has the User's digital certificate/public key or can obtain it separately from the issuing Certificate Authority if they wish to validate the User's signature (neither the digital certificate or public key are returned to the SP by the ID GW)

2.2.2.2 Variant 2b:

The service includes the User's digital certificate (or digital certificate URL) in the response to the SP, hence providing the true identity of the User via the User's digital certificate

2.2.2.3 Variant 2c:

The service returns only the public key in the response to the SP; as a result, no User information is made available to the SP hence this Variant could be used where the use case requires anonymity

2.2.3 Summary of the four Variants

	SP request		ID GW Response		
	Authorisation request	**'Data'**	**Authorisation response**	**Signed 'Data'**	**User's Certificate/URI or Public Key**
Variant 1	✓		✓		
Variant 2a	✓	✓	✓	✓	
Variant 2b	✓	✓	✓	✓	Digital certificate (or URL)
Variant 2c	✓	✓	✓	✓	Public key

Table 2: Variants

Note: That in ALL Variants, a response signed with the User's private key is generated but in Variant 1 the signed response is not returned to the SP as part of the service (i.e., Variant 1 provides LoA4 authorisation only)

- A document signing service would typically use Variant 2b to return the signed document data and the true identity of the individual who signed it.

[5] Digest terminology is used when mc authorize PKI is used for login

[6] Hash of data or document terminology is used when MC Authorise PKI is used for used for approval or document signing.

- Variants 1 and 2c might be important for ensuring GDPR compliance for some use cases (as an example).
- Variant 2a would typically be used where the SP can identify the User and is already in possession of either the User's public key or User's digital certificate.

Note: That the Mobile Connect Authorise PKI service provides a mechanism for SPs to request a User to authorise a transaction. Mobile Connect will process the request, log the User's signed response and if requested by the SP return the User's signed response with the User's public key or User's digital certificate to the SP. It is up to the SP to process this response and decide the final outcome of the transaction.

2.3 Service Registration

Before a User is able to use the Mobile Connect Authorise PKI service, they will first need to register for the service with their Operator, be provisioned with an associated User's digital certificate (and PKI key pair[7]) and set up an appropriate authenticator (e.g. SIM applet).

The registration process needs to be sufficiently robust to ensure that the certificates and keys are issued to the right individual. Essentially there are 3 steps:

1. ID proofing: User presents their national ID card and/or passport or other proofs to their Operator acting as a Registration Authority (RA)[8]
2. ID verification: the Operator/RA verifies the authenticity of the proofs received
3. Certificate issuance: the Operator/RA provides User details to a Certificate Authority (e.g., User's name and National ID[9]) which then issues the User's digital certificate (bound to the User details provided by the Operator). A private key is paired with the certificate using a standardised certificate signing request and is securely stored, typically within the Mobile Connect authenticator on the User's device[10].

In Mobile Connect, the User's digital certificate must be based on X.509 Certificate standards i.e. RFC 5280, 6818, 8398, 8399. Further details on Registration and Certification Authorities are provided in 6.2.4.

2.4 Mobile Connect Authorise PKI Flow

The flow for the MC Authorise PKI service mirrors that used for MC Authorise Plus in obtaining authorisation for a SP-provided request - if the User approves the request, the Operator will return a successful authorisation response. An appropriate error is returned if the User cancels the request or fails to authenticate (e.g., via PIN or biometric).

[7] In some cases Operators may choose to issue one key pair for each specific use case, this is left for the Operator to define.

[8] The Operator can also choose to outsource this to an external RA

[9] Note that this is left down to local Operator/legislation requirements

[10] Split key and 'PKI in the Cloud' deployment options may also be applicable

Depending on the particular MC Authorise PKI service Variant being used, the signed response and associated User public key or User's digital certificate may also be provided to the SP.

The following diagram shows a simplified MC Authorise PKI service flow illustrating how the service is presented to the User.

Note: That this illustrates a Device-Initiated request [6], but a Server-Initiated request [7] can also be used. Mobile Connect Technical Architecture and Core Requirements [5] provides more detailed sequence diagrams illustrating the flows for Device-Initiated and Server-Initiated modes.

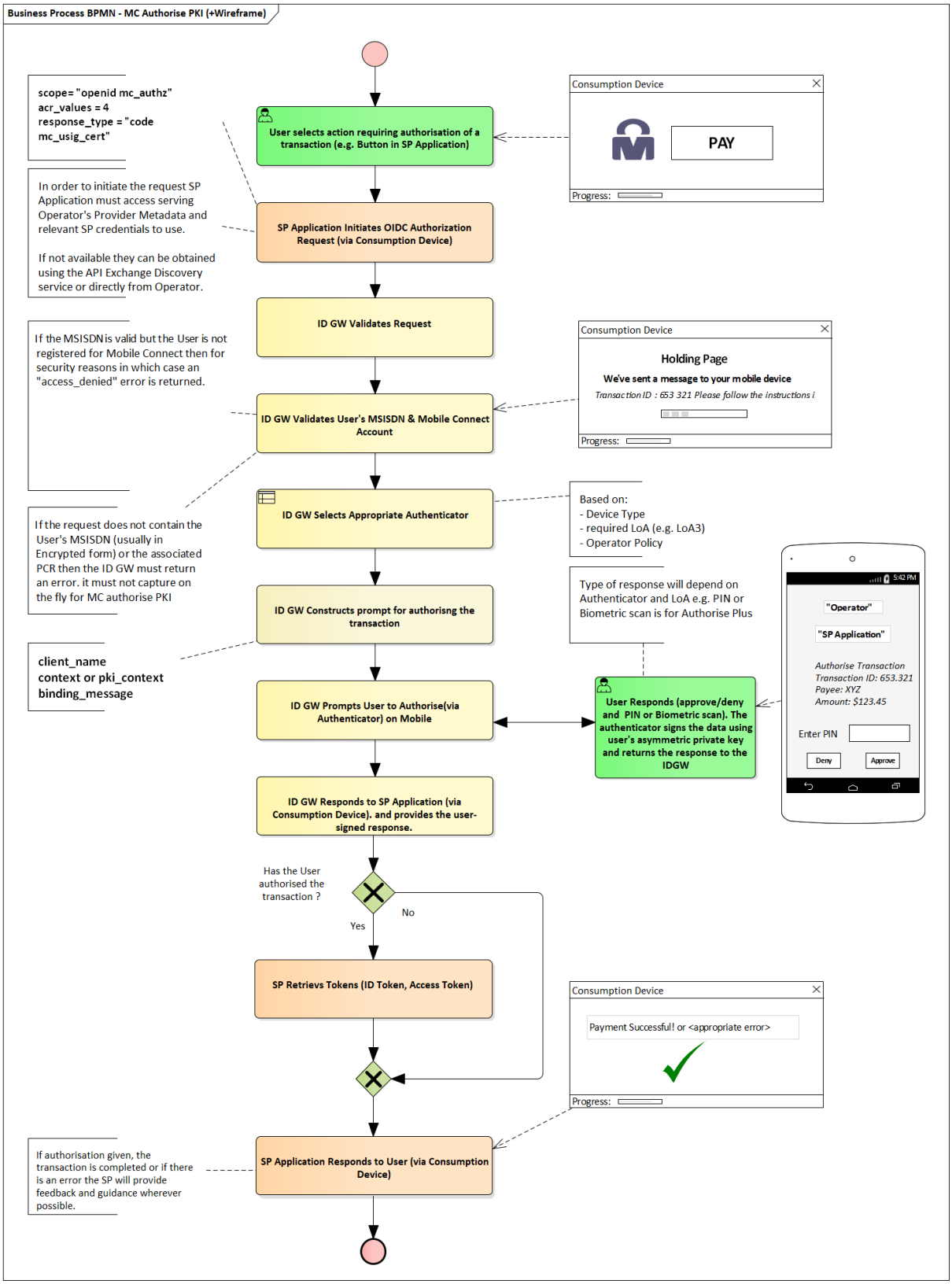

Figure 1: Mobile Connect PKI Authorise Service Flow

- The User is accessing a service provided by a SP either through a native app or through a browser on the Consumption Device (e.g. a laptop) and within the SP

application there is an option to authorise a payment, for example, using Mobile Connect.

- The SP application initiates the MC Authorise PKI service request (OIDC Authorization Request) to the Operator's ID GW Authorize endpoint by specifying the `scope` parameter and required LoA using the `acr_values` parameter in the request (see :)

  The following parameters are also supplied by the SP in the request:

  - `client_name` (required) specifies the SPs short name for their application.
  - `context` or `pki_context` (depending on Variant) provides the required information for the authorisation (i.e. what the User is being asked to authorise and the data that should be signed). See section 3.2.3.
  - `binding_message` allows the same message to be displayed on both the Consumption Device (e.g. on the holding page) and on the Authentication Device so that the User can link what is seen on both devices.
  - `response_type` identifies the MC Authorise PKI Variant being requested by the SP.

- As per Device-Initiated mode, the User is redirected to the ID GW holding page which prompts the User to check their mobile device.
- The Operator's ID GW validates the request (i.e. that the SP has been registered with the Operator for the MC Authorise PKI service requested and that the required parameters are included in the correct format)
- The Operator ID GW checks the MSISDN and whether the User is registered for the Mobile Connect Authorise PKI service
- The authorisation prompt is then constructed using the `client_name` and `context` or `pki_context` and `binding_message`[11]. Section 2.6 provides more details on the authorisation prompt.
- Assuming the User successfully authorises the transaction on their mobile device, the data-to-be-signed[12] is signed with the User's private key. Depending on the MC Authorise PKI Variant requested by the SP (via the `response_type`), the signed data along with the User's public key or digital certificate is returned inside the ID Token.
- The ID Token and Access Token[13] are retrieved by the SP. The method for retrieving the ID Token differs depending on whether Device-initiated or Server-Initiated

[11] Note "prompt action text" relating to the requested action depending upon LoA is inserted into the authorisation prompt by the selected Authenticator

[12] As explained in the previous sections, the data to be signed with the User's private key depends on the Variant type selected.

[13] Depending on the SP implementation the SP can complete the transaction by receiving ID token and access token OR if SP would like to submit tokens to their internal systems they can use Access Token and ID Token. Implementation details are out of scope.

requests were made. For Device-Initiated mode, the ID GW issues an Authorization Code and control is transferred to the SP server to initiate a token request using that code to retrieve the ID Token and Access Token (Token Response).
- Receipt of the ID Token provides the details of the successful authorisation thereby allowing the SP to proceed with the transaction.

A SP can also choose to use a Server-Initiated MC Authorisation service request where the request is initiated directly from the SPs server rather than via the User's Consumption Device and can be initiated without the User having to be online or initiating a transaction via pressing a button in an application. An example of the use of a Server-Initiated request could be where a bank detects the need to get an explicit approval from the User for a payment request initiated from an unusual location. The bank will use the MSISDN on file for the User to initiate the authorisation request.

This process is broadly the same as described in **Error! Reference source not found.** but with the following differences:

- The format of the request (defined in the Mobile Connect OIDC Server-Initiated OIDC Profile [7]) is different but the same `scope` values are used. The User must be identified using their MSISDN (via the `login_hint` or `login_hint_token` parameter). The mechanism to obtain the ID Token and the Access Token also differs.
- The User is not interacting with the SP application via a Consumption Device so no holding pages can be presented. The User will always be presented with the authorisation prompt and will respond on the Authentication Device.

Mobile Connect Technical Architecture and Core Requirements [5] provides more detailed sequence diagrams illustrating the flow for Device-Initiated mode and the Server-Initiated mode.

2.5 Authenticators for MC Authorise PKI

As per all other Mobile Connect services, the Operator deploying this service has a choice of which authenticators to use for obtaining User authorisation.

Two authenticators are applicable to MC Authorise PKI:

- SIM applet
- Smartphone App Authenticator (SAA)

If the Operator chooses to utilise SIM applet as the authenticator then the deployment architecture will need to include an MSSP, wPKI SIM etc. and if the choice is to use SAA then there may be a dependency for a TEE[14] on the device, SAA Authenticator server and usage of an existing push messaging platform.

If the SIM applet authenticator is used in the deployment of this service, the Operator will need to consider the impact of SIM applet distribution and potentially a need for distribution of new SIMs (given the dependencies of this service on SIMs with wPKI capability and the

[14] Trusted Execution Environment

SIM applet size for PKI mitigating the ability to deliver this applet OTA). In addition, the Operator will need to review and potentially adjust their existing User enrollment processes.

Note: That an Operator may decide to require/use separate PINs from the User depending on which Mobile Connect service is being used (e.g., MC Authenticate Plus vs MC Authorise Plus).

2.6 Authorisation Prompt

Figure 2 shows an example authorisation prompt on the User's mobile device (Authentication Device). In the case of the SIM applet authenticator, in order to ensure interoperability across different markets requires the maximum length of the prompt to be 220 bytes.

If Variant 1 is being requested, the `context` parameter must be used to construct the prompt action text.

If Variant 2 is being requested, the `dtbd` parameter value within `pki_context` must be used to construct the prompt action text.

This means that: Length(`client_name`) + Length(`context or dtbd`) + Length(`binding_message`) <= 220 bytes.

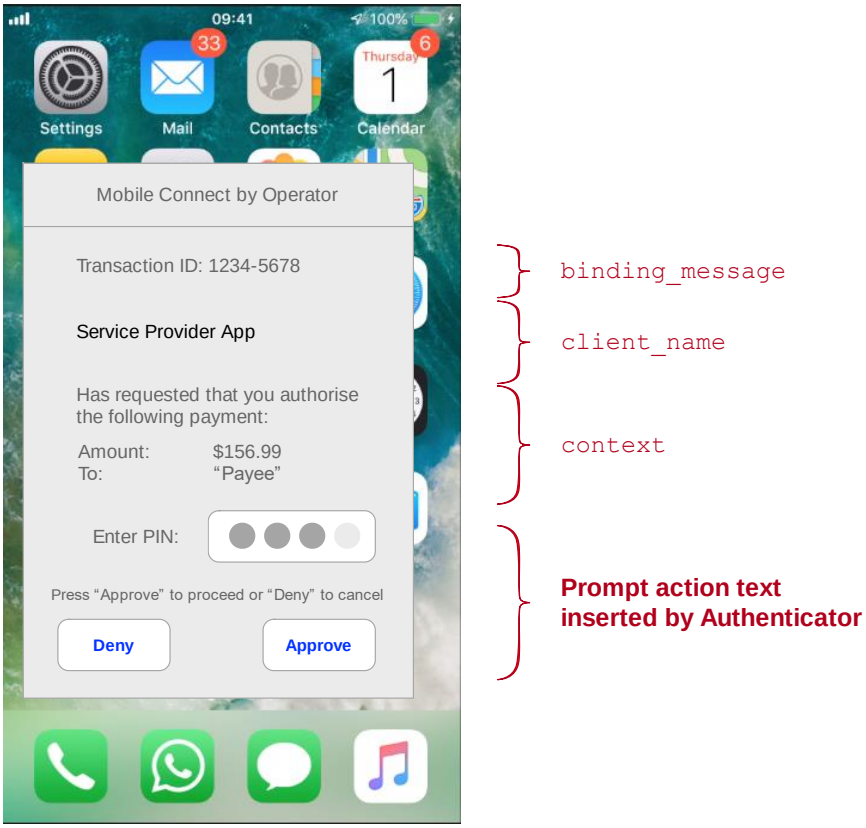

Figure 2: Example Authorisation prompt

For the best user-experience, it is recommended that the prompt text comprising of both authentication ("enter pin") and authorisation ("do you authorise…") are rendered on a single screen. However, it is recognised that this may not always be possible due to limited space

on a single screen where local language text may require more space or where, in certain countries, regulations may require explicit confirmation to be displayed to the User on the Authentication Device.

2.7 Authorisation Response

A successful MC Authorisation results in the return of an ID Token and an Access Token to the SP application. An authorisation failure will be returned if the authorisation was not successful.

Note: That MC Authorisation specific errors are defined in section 6 and generic error codes, applicable to all supported services, are defined in the relevant MC OIDC Profile (Device-Initiated [6] or Server-Initiated [7]).

The ID Token provides confirmation of the successful authorisation and includes a Pseudonymous Customer Reference (PCR) identifying the User which can be used in subsequent Mobile Connect service requests.

Depending on the MC Authorise PKI Variant being requested, the ID token may also contain the User-signed response, User's public key or User's digital certificate along with the `displayed_data` parameter which is constructed by using `client_name, dtbd or context` and `binding_message`.

After a User has authorised a transaction for the first-time using Mobile Connect, the SP Application can store the PCR associated with the User and the serving Operator details (issuer ID (`iss`)).

The ID Token has a period of validity, defined by the Operator and returned within the `exp` value (expiration time) within the ID Token. This must be kept as short as possible by the Operator so that the authorisation is only valid for a single transaction.

2.8 Personal data and consent handling

Depending on the MC Authorise PKI Variant being used, the User's public key or digital certificate or certificate URI may be shared with the SP. In the scenario where the User's digital certificate is shared, this will implicitly provide the SP with access to the User information registered and populated into the certificate when it was issued by the CA. As such, it is important that where such information is shared (or accessible) by the SP, that sufficient User consent has been obtained to permit SP access to this information and any personal data. The underlying principle will be to share as little as needed to support the use case with a SP, hence the reason why the MC Authorise PKI service has been defined with a number of different Variants, Variant 2c ensuring that Mobile Connect does not share any User information.

For Variant 2b where the User's digital certificate or certificate URI is returned to the SP, the release of the digital certificate (and associated User identity) is then down to the CA and outside the scope of the MC Authorise PKI service although the User should be made aware of this within the terms of service when signing up for the MC Authorise PKI service and notified that their digital certificate/User info will be shared when providing their authorisation.

In the case where the digital certificates are issued by and managed by an external nationally-accredited CA (Variant 2a), the SP might use MSISDN in their request to the CA to fetch the required User's digital certificate but this is out of scope of MC.

User consent should be managed in accordance to regulations, with minimal user-experience friction. The following options may be considered:

- During sign up, a User may give consent for their Operator to share their digital certificate with SPs the User engages with.
- For transactions where the Operator is sharing personal data, it may be appropriate to present a consent message on the screen as the User approves the transaction; i.e., either on the consumption device or the authenticator. The consent message should outline what attributes will be shared with the SP but not the actual data.
- There may be instances where the Operator may trust the SP to capture consent and in those instances it is recommended that the ID GW does not generate another consent message due to the negative impact on user-experience.
- Mobile Connect Product Manager's Handbook **Error! Reference source not found.**[15] and Mobile Connect Privacy Principles [5] provide more detailed guidelines on obtaining active consent.

3 Mobile Connect Authorise PKI Service Specification

This section contains the relevant information required by Operators to implement and support the MC Authorise PKI service.

3.1 API Modes Supported

Each MC Authorise PKI Variant can be implemented using either the DI, SI-notification or SI-polling API mode. The Operator MUST publish through the ID GW Provider Metadata configuration the MC Authorise PKI Variants and API modes that they support.

3.2 OIDC Authorization Request Parameters

3.2.1 scope and acr_values

The SP requests the Mobile Connect Authorise PKI service by specifying `scope` and `acr_values` parameters in the Mobile Connect OIDC Authorization Request as described in : .

Mobile Connect Service	`scope` value[16]	LoA (`acr_values`)
Mobile Connect Authorise PKI	`"openid mc_authz"`	`4`

Table 3: Scope and acr_values

[15] Chapter 7 Consent Management

[16] "openid" must be included within the `scope` parameter as a string followed by the relevant Mobile Connect service descriptors separate by spaces

3.2.2 response_type values

As stated previously, Mobile Connect Authorise PKI can be deployed in four different Variants based on particular market and business requirements. The Variant and API mode being requested must be provided via the `response_type` value in the Mobile Connect Authorise PKI's OIDC Authorization Request.

Note: That the following values are valid only if the `scope` and `acr_values` have the above mentioned values (indicating a request for the MC Authorise PKI service), otherwise the ID GW must return an error (see section 6).

`response_type` value	API Mode	Variant	Description
"code"	DI	1	Signed data not returned to SP
"code usig"	DI	2a	Only signed data returned
"code usig_cert"	DI	2b	Signed data and User's digital certificate or URI (whatever is registered for that User) returned
"code usig_pkey"	DI	2c	Signed data and public key returned
"mc_si_async_code"	SI Notification	1	Signed data not returned to SP
"mc_si_async_code usig"	SI Notification	2a	Only signed data returned
"mc_si_async_code usig_cert"	SI Notification	2b	Signed data and User's digital certificate or URI (whatever is registered for that User) returned
"mc_si_async_code usig_pkey"	SI Notification	2c	Signed data and public key returned
"mc_si_polling"	SI Polling	1	Signed data not returned to SP
"mc_si_polling usig"	SI Polling	2a	Only signed data returned
"mc_si_polling usig_cert"	SI Polling	2b	Signed data and User's digital certificate or URI (whatever is registered for that User) returned
"mc_si_polling usig_pkey"	SI Polling	2c	Signed data and public key returned

Table 4: `response_type` values

3.2.3 Additional Request parameters

If MC Authorise PKI Variant 1 is used (`response_type` = "code" or "mc_si_async_code" or "mc_si_polling", the following parameters should be used:

Parameter Name	Usage Category	Description
`context`	REQUIRED	As defined in the MC OIDC Profiles [1] and [2].
`binding_message`	REQUIRED	As defined in the MC OIDC profiles [1] and [2]

Parameter Name	Usage Category	Description
client_name	REQUIRED	As defined in the MC OIDC profiles [1] and [2]

Table 5: MC Authorise PKI Variant 1 request parameters

If MC Authorise PKI Variant 2 is used where a signed response is returned to the SP, the following parameters should be used:

Parameter Name	Usage Category	Description
pki_context	REQUIRED	A JSON object that contains all PKI context parameters as described in **Error! Reference source not found.**
binding_message	REQUIRED	Same as defined in the MC OIDC profiles [1] and [2]
client_name	REQUIRED	Same as defined in the MC OIDC profiles [1] and [2]

Table 6: MC Authorise PKI Variant 2 request parameters

Parameter Name	Usage Category	Description
dtbd	REQUIRED	Data to be displayed. The value must be a string. It identifies the data that needs to be displayed to the User on the Authentication device – i.e., the 'Authorisation text'.
dtbs	REQUIRED	Data to be signed. Either a string or pre-computed data depending on the use case. [e.g., a hash of the document]
sig_type	REQUIRED	The type of the signature to be used, as defined in the ETSI Mobile Signature Services standard: • XML Signature • CMS-Signature (Cryptographic Message Syntax) • PKCS#7 • PKCS#10 The signature types supported by the Operator should be declared in the Operator's OIDC Provider Metadata.

Table 7: MC Authorise PKI Variant 2 `pki_context` JSON request parameters

3.3 MC Authorise PKI ID Token response

If MC Authorise PKI Variant 1 is used, the ID Token response parameters should be the same as for the Mobile Connect Authorise Plus service with the exception that `acr_values` = 4.

If MC Authorise PKI Variant 2 is used where a signed response is returned to the SP, the following parameters must be used in the response.

Parameter	Mandatory	Specification
`dts`	REQUIRED	Data signed – the signature using User-specific private key.
`sig_type_used`	REQUIRED	The signature type used – generally the same value as included in the `sig_type` input parameter but if the ID GW uses a different signature type to that requested, it must indicate the type used via this response parameter.
`dtbs_used`	REQUIRED	The "data to be signed" used – generally the same value as included in the `dtbs` input parameter for ensuring what has been used for the signature creation
`displayed_data`	REQUIRED	The data which was displayed to the User via the Authentication device: `"client_name"` + `"-"` + `"binding_message"` + `"-"` + (`"dtbd"` or "`context`") Note that for Variant 1, `context` parameter must be used and for all other Variants `dtbd` must be used.
`dts_time`	REQUIRED	The time when the signature was created. The format is the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time specified.

`cert_x509`
Or
`cert_x509_uri`
	REQUIRED
	For service Variant 2b and the following `response_type` values:
	`"code usig_cert"`
	`"mc_si_async_code usig_cert"`
	`"mc_si_polling usig_cert"`
		If certificate is returned, then it must be a Base64 encoded X509 certificate or certificate chain.
		If certificate URI is returned then it must be a X509 certificate URI, which should be accessible by the SP.

`public_key`	REQUIRED For service Variant 2c and the following `response_type` values:	Base64 encoded public key

Parameter	Mandatory	Specification
	"code usig_pkey" "mc_si_async_code usig_pkey" "mc_si_polling usig_pkey"	

Table 8: MC Authorise PKI Variant 2 ID Token response

3.4 Request and Response Reference Table

3.4.1 MC Authorise PKI Variants and Authorization Request Parameters

The following table summarises the request parameters required for the different MC Authorise PKI Variants:

Request Parameter	Service Variant / Usage			
	Variant 1	Variant 2a	Variant 2b	Variant 2c
client_name	✓	✓	✓	✓
binding_message	✓	✓	✓	✓
context	✓			
pki_context		✓	✓	✓

Table 9: MC Authorise PKI request parameters summary

3.4.2 MC Authorise PKI Variants and ID Token Response Parameters

The following table lists all PKI specific ID token response parameters (ID token claims)

ID Token Claim Name	Service Variant / Usage			
	Variant 1	Variant 2a	Variant 2b	Variant 2c
dts		✓	✓	✓
sig_type_used		✓	✓	✓
dtbs_used		✓	✓	✓
displayed_data	✓[17]	✓[18]	✓	✓
dts_time		✓	✓	✓
cert_x509 or cert_x509_uri			✓	
public_key				✓

Table 10: MC Authorise PKI ID Token response parameters summary

[17] The contents of the displayed_data must contain client_name, binding_message and context parameter values submitted through the authorization request.

[18] For all 2x Variants the displayed_data must contain client_name, binding_message and dtbd parameter values submitted through the authorization request.

3.5 Service-specific Requirements

The following table provides service-specific requirements relating to Mobile Connect Authorise PKI. These should be used in conjunction with the MC Authorise Plus requirements in the implementation of this Mobile Connect service. Core Requirements are specified in the Mobile Connect Technical Architecture and Core Requirements [5]. Note that these are common to all Mobile Connect services. For terminology and associated specifications please refer to the Mobile Connect Technical Overview [4].

Number	Relating To	Requirement
MC_AUTHZ_PKI_01	Service Registration	Registration and service activation must take account of the additional roles i.e., the RA and CA and obtain a User-specific key/User's digital certificate from the CA.
MC_AUTHZ_PKI_02	Support of Service	User consent for attributes in the User's digital certificate, where shared, must be handled in accordance with regulation and Mobile Connect privacy policies
MC_AUTHZ_PKI_03	Support of Service	ID GW must support one or more MC Authorise PKI service Variants and the `response_type` values stipulated in : .
MC_AUTHZ_PKI_04	Support of Service	ID GW must use the `response_type` parameter to determine the MC Authorise PKI Variant being requested and whether or not to return the User's signed response, public key and User's digital certificate.
MC_AUTHZ_PKI_05	Support of Service	The Mobile Connect Authorise PKI service must support high security multi factor (LoA4) authorisation via a User's mobile device using an appropriate authenticator. Note that an authentication step is implicit within the authorisation
MC_AUTHZ_PKI_06	Service Registration	ID GW must be able to allow a SP (client application / service) to register for the MC Authorise PKI service and be provisioned with the requisite SP-provided parameters dependent on whether the SP intends to use Device-Initiated mode or Server-Initiated mode when requesting the service and what modes are supported by the ID GW.
MC_AUTHZ_PKI_07	Authenticator	Authorisation of a transaction by the User must be via the Authenticator on their Authentication Device (i.e. their mobile device).
MC_AUTHZ_PKI_08	Authenticator	The appropriate Authenticator is selected by the ID GW based upon the requested acr_values (LoA), and which Authenticators are provisioned on the User's mobile device.
MC_AUTHZ_PKI_9	Service Invocation	The SP will specify the required MC Authorise PKI service via the `scope` parameter, `acr_values` parameter and `response_type` within the OIDC Authorization Request. The ID GW must support the use of these parameters for the Mobile Connect Authorise PKI service.
MC_AUTHZ_PKI_10	Service Request - Validation	The ID GW must validate the submitted MC Authorise PKI service request (OIDC Authorization Request) and request parameters as defined in the MC Device-Initiated OIDC Profile or the MC Server-Initiated OIDC Profile, as appropriate. The ID GW must implement all REQUIRED parameters specified in this document to support

Number	Relating To	Requirement
		the MC Authorise PKI service. For MC Authorise PKI, as well as the REQUIRED parameters, the SP must submit the `context` or `pki_context` parameter as appropriate based on the MC Authorise PKI Variant being requested. If the REQUIRED parameters are not provided, an error must be returned.
MC_AUTHZ_PKI_11	Service Request - SP Validation	The ID GW must check that the SP is registered for the requested MC Authorise PKI Service and is registered to use Device-Initiated or Server-Initiated modes as defined in the MC Device-Initiated OIDC Profile or the MC Server-Initiated OIDC Profile, as appropriate.
MC_AUTHZ_PKI_12	Service Request - User Validation	The ID GW must check whether the User is already registered for the MC Authorise PKI service and if not, the SP request must be rejected with an appropriate error code as specified in the MC Device-Initiated OIDC Profile or the MC Server-Initiated OIDC Profile, as appropriate, subject to Operator Policy.
MC_AUTHZ_PKI_13	Service Request - Prompt	The ID GW must present a prompt to the User that includes: - SP provided short application name (`client_name` parameter in OIDC Authorization Request, 16 bytes max) - SP provided context for the transaction/action to be Authenticated: o If Variant 1 is selected then the `context` parameter must be used in constructing the service request prompt. o If Variant 2 is requested then the `dtbd` parameter from `pki_context` must be used to construct the service request prompt. - SP provided message to link the Authentication Device and Consumption Device[19] (binding_message parameter in OIDC Authorization Request). For the best User experience, it is recommended that the prompt is displayed on a single screen (not across a sequence of screens) unless local regulations dictate otherwise.
MC_AUTHZ_PKI_14	Service Request - Prompt	For interoperability purposes, the maximum length of the prompt is <=220 bytes (based on the SIM-Applet Authenticator). Depending on the service Variant, the length of the `context` or `dtbd` message as a result will be determined by: length(`context` or `dtbd`) <= 220 – length(`binding_message`) – length(`client_name`) ID GW Policy may recommend the maximum prompt length that should be used by SPs based on which Authenticators will be used for displaying the prompt and capturing User authorisation. Note - The maximum length of the `binding message` is implementation specific but must be within the limits of the prompt maximum length.

Number	Relating To	Requirement
MC_AUTHZ_PKI_15	Service Response	The Mobile Connect Authorise PKI service must return to the initiating SP application: - a positive result, or - a negative result with an appropriate error code and error description. Note that a positive result will provide an ID Token and an Access Token. The ID Token will include a PCR, which uniquely identifies that User to the SPs client, and details of the User authentication/authorisation as defined in the MC Device-Initiated OIDC Profile or the MC Server-Initiated OIDC Profile and this document, as appropriate.
MC_AUTHZ_PKI_16	Error Responses	Error Responses may be returned at different stages of the processing of a service request as specified in the MC Device-Initiated OIDC Profile and the MC Server-Initiated OIDC Profile and must be supported for the MC Authorise PKI service. These errors are generic to all Mobile Connect services. Service Specific Error Responses are specified in section 6
MC_AUTHZ_PKI_17	Tokens - ID Token	The ID GW must return the `displayed_data` claim in the ID Token which combines the `client_name`, `context` or `dtbd` and `binding_message`. `displayed_data` uses the following format: `client_name` + "–" + `binding_message` + "–" + `context` or `dtbd`[20] Note that "–" is added to differentiate the parameters.
MC_AUTHZ_PKI_18	Tokens	The ID GW must not return a Refresh Token or if it does it must be with a null value.
MC_AUTHZ_PKI_19	Tokens	The ID GW must issue Access Tokens with a zero time-to-live using a very low value (e.g. 10s) for the `expires_in` parameter in the Token Response or by restricting it to a single-use token.
MC_AUTHZ_PKI_20	Transaction Logs	A complete Mobile Connect transaction log must be maintained, archived and accessible to resolve any disputes in line with local data protection laws and the Operator's data retention policy. For MC Authorise PKI this should include: • Date & Time • MSISN, PCR • Service requested (i.e. `scope` + `acr_values` + `response_type`) • User Response (approve, timeout or authorisation failure) • Status (Complete, In-process, error) • `displayed_data` (i.e., prompt that was displayed on Mobile device and returned in the ID Token) • Authenticator type used (as per the returned `amr` value) • Level of Assurance requested and used • Error codes and error description.

[20] As explained in the previous sections, depending on the requested Variant from the SP, the ID GW must construct `displayed_data` using either `context` or `dtbd` as appropriate.

Table 11: Service specific requirements

4 MC Authorise PKI High-Level Flows

This section describes a high-level flow of Mobile Connect Authorise PKI independent of the DI and SI mode protocols. For protocol specific information refer to [3] and [4].

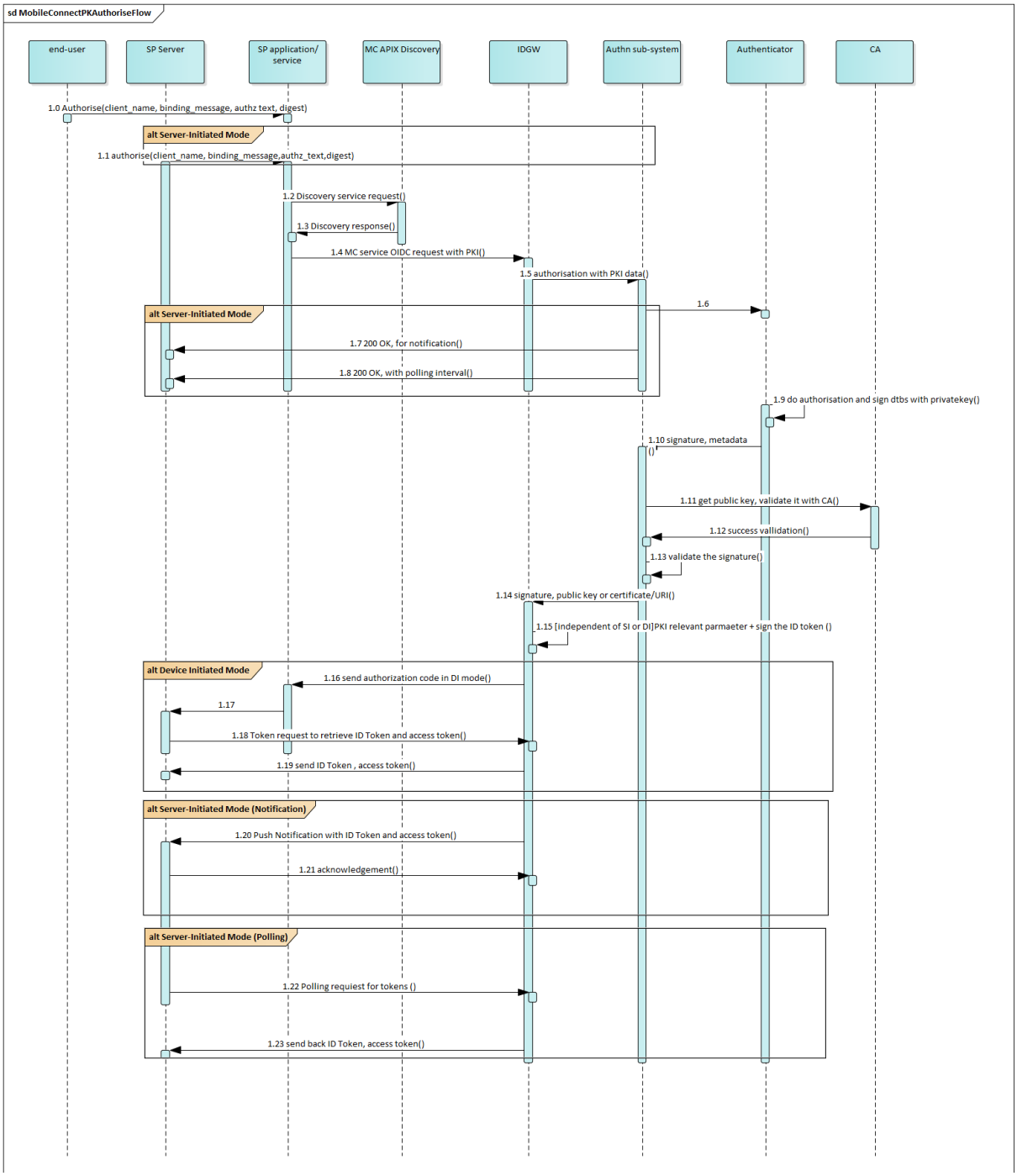

Figure 3: Mobile Connect PKI Authorise Flow

1. SP Application/services [or server] makes a Discovery call for the target MSISDN and receives the Operator ID GW endpoints.
2. SP makes Mobile Connect Authorise PKI service authorization request using either Mobile Connect Device-Initiated or Server-Initiated mode.
3. Operator ID GW validates scope, response_type and acr_values and determines it is a request for the Mobile Connect Authorise PKI service for a specific variant.
4. Operator ID GW validates the PKI-specific input parameters within the authorization request based on the requested variant as listed in the input parameters section of the OIDC authorization request. If Variant 1 is requested, then all input parameter relevant to MC Authorise Plus are considered for processing; if any of the Variant 2x is requested, then pki_context (a JSON object that contains PKI specific input parameters for Mobile Connect Authorise PKI) is taken into consideration for processing the request.
5. Upon successful request validation:

    a) For DI mode, ID GW performs the PKI authorisation by submitting relevant parameters to the authentication subsystem
    b) For SI mode, ID GW simply returns 200 OK to the SP application and for polling along with a polling interval, and then performs PKI based authorisation by submitting relevant parameters to the authentication subsystem.
6. The authorisation prompt displays a message including the client_name, binding_message and the context (for variant 1) or dtbd (from pki_context for all Variant 2x) in the SP service request.
7. Note that based on the authentication subsystem used, the data to be signed parameter will be passed so that it will be signed by the User's private key [that is stored on either the authenticator or implementation specific storage]; User's signed response should be returned to the SP application through the dts parameter.
8. After successful authorisation;

    a) In the DI mode, the ID GW returns authorization code along with redirect 302. And SP fetches the ID token and Access token by making a token request

9. the `dtbs` passed by the SP in the request is signed using the User specific private key. The signature is performed in a location specific to the Authenticator in use (e.g. for SIM Applet the signature is performed in the SIM Applet itself).
10. Signed `dtbs` signature is returned through the `dts` parameter in the final response along with User public key/certificate/or certificate_uri via the ID Token as appropriate based on the MC Authorise PKI Variant requested.
11. The Operator ID GW signs the ID token and sends the response to the SP's application / service.
12. SP application / service receives the ID Token and Access Token and based on the content of the ID Token (User's digital certificate or the User's public key), the SP performs the following :
    a) ID Token signature will be verified by using the ID GW public key from the jwks_uri.
    b) If ID Token has a certificate embedded into it, SP retrieves the public key from the User's digital certificate and verifies the signed data signature (`dts`).

c) If ID Token has a certificate URI, the SP retrieves the User's digital certificate from the URI, then retrieves the public key and verifies the signed data signature (`dts`).
d) If ID Token has a public key (without a User's digital certificate), the SP uses it to verify the signed data signature (`dts`).

13. Upon successful validation, SP accepts the response in DI mode; and accepts and acknowledges in the SI mode.

5 Provider Metadata

This section lists the Provider Metadata entries that are required specific to PKI based services such as MC Authorise PKI.

Parameter name	Usage Category	Description
`mc_pki_enabled`	REQUIRED	Allowed values : `true / false` This parameter must be set to `'true'` if PKI based services are supported otherwise `'false'`. Before making any MC service request, SP application / services must check this parameter to determine whether PKI based MC services are supported by an ID GW.
`mc_pki_service_type_supported`	REQUIRED	If `'mc_pki_enabled'` is true, then this entry must have a value. This is a JSON array of strings. Single or multiple values are allowed in the array. These values will be used within the request parameter response_type parameter. Only one value at a time is allowed. `["usig", "usig_cert", "usig_pkey"]` Example : `response_type = "code usig_cert".`
`mc_pki_sigkey_type`	REQUIRED	If `mc_pki_enabled` is `'true'` , this parameter must have a value. Array of string (s). Single string also must wrap in an array. Allowed values for Mobile Connect : **`"public_key"`** = if the PKI based MC services response contains public key [anonymous User]. **`"cert_x509"`** = if the PKI based MC services response contains a public key wrapped in an X509 certificate that is tied to a specific Mobile Connect User. **`"cert_x509_uri"`** = if the PKI based MC services response contains a reference to a public key wrapped

Parameter name	Usage Category	Description

		in an X509 certificate and tied to a specific Mobile Connect User.
		Multiple key types are allowed. examples :
		`["cert_x509"] [or]`
		`["cert_x509", "cert_x509_uri"] [or]`
		`["cert_x509","cert_x509_uri","public_key"]`

Table 12: Mobile Connect PKI Authorise – Provider Metadata

6 Error Handling

The error handling mechanism for Mobile Connect Authorise PKI is the same as for Mobile Connect Authorise Plus. Refer [1] and [2] for generic guidance on error handling.

6.1 Device-Initiated Mode

6.1.1 Authorization Request

All generic errors that are listed in the Mobile Connect Technical Architecture and Core Requirements [1] must be implemented. Additionally, the following service-specific errors must be supported.

Error Scenario	HTTP mode	Error code	Error Description [RECOMMENDED text]
PKI based authorisation is not supported (ID GW has agreed contract with SP)	Redirect 302	server_error	"service is not available"
PKI based authorization is not supported (ID GW does not have contrac with SP or not implemented)	Redirect 302	invalid_request	"service is not available"
acr_values = 4 and pki_context does not exist	Redirect 302	invalid_request	"pki_context does not exist for PKI based authorisation"
Request is for PKI based authorisation but dtbs parameter does not exist	Redirect 302	invalid_request	"required parameter dtbs does not exist"
Request is for PKI based authorisation and sig_type exists Operator does not support this (ID GW has NOT published the sig_type through provider metadata)	Redirect 302	invalid_request	"required parameter sig_type does not exist.
Request is for PKI based authorisation and sig_type exists Operator does not support this (ID GW has published the sig_type through provider metadata)	Redirect 302	server_error	"Server does not support the requested sig_type"
Dtbs exists but it is empty	Redirect 302	invalid_request	"Required parameter dtbs value is missing"

Table 13: Errors – Device Initiated Mobile Connect PKI Authorise Request

6.1.2 Token Request

Same as defined in the Mobile Connect Device Initiated Profile [5].

6.2 Server-Initiated Mode

For all generic polling and notification errors refer to mobile connect profiles [5] and [6].

6.2.1 Server Initiated Authorization Endpoint

All generic server-initiated errors listed in Mobile Connect Technical Architecture and Core Requirements must be supported. In addition, the following service-specific errors must be implemented for PKI based services.

Error Scenario	HTTP mode	Error code	Error Description [RECOMMENDED text]
PKI based authorisation is not supported (ID GW has agreed contract with SP)	501	sever_error	"service is not available"
PKI based authorization is not supported (ID GW does not have contrac with SP or not implemented)	400	invalid_request	"service is not available)
acr_values = 4 and pki_context does not exist	400	invalid_request	"pki_context does not exist for PKI based authorisation"
Request is for PKI based authorisation but dtbs parameter does not exist	400	invalid_request	"required parameter dtbs does not exist"
Request is for PKI based authorisation but sig_type does not exist	400	invalid_request	"required parameter sig_type does not exist.
Request is for PKI based authorisation and sig_type exists Operator does not support this (ID GW has published the sig_type through provider metadata)	501	server_error	"Server does not support the requested sig_type"
Request is for PKI based authorisation and sig_type exists Operator does not support this (ID GW has NOT published the sig_type through provider metadata)	400	Invalid_request	"Service does not suppot the request sig_type".
dtbs exists but it is empty	400	invalid_request	"Required parameter dtbs value is missing"

Table 14: Errors – Server Initiated Mobile Connect PKI Authorise Request

6.2.2 Notification Error

All generic errors that are listed in the Mobile Connect Technical Architecture and Core Requirements [1].

6.2.3 Notification Acknowledgement errors

The following errors must be logged by an ID GW for auditing purposes.

Note: The ID GW MUST never redirect/process HTTP status codes 3xx received as notification acknowledgement. Hackers can exploit the situation and can introduce dangerous security vulnerabilities.

Error Scenario	HTTP mode	Error code	Error Description [RECOMMENDED text]
dts parameter does not exist	400	`invalid_request`	"unable to verify the signature"
DTS parameter exist but it is not signed with the correct key [OR] Dts validation is failed [OR] it is empty	400	`invalid_request`	"unable to verify the signature"
sig_type_used parameter is empty [OR] value exist but invalid [OR] value is empty	400	`invalid_request`	"unable to verify the signature"
dtbs_used exists but empty [OR] dtbs_used does not exist [OR] dtbs_used exists but it is not equal to dtbs parameter that is sent in the authorization OIDC request	400	`invalid_request`	"data is manipulated, malformed response.
displayed_data parameter does not exist [OR] displayed_data exist, but empty [OR] displayed_data exist but the content are not matching	400	`invalid_request`	"data is manipulated, authentication is executed with malformed data"
Dts_time parameter does not exist [OR] Dts_time parameter exist but has invalid value [OR] Dts_time parameter exist but is empty	400	`invalid_request`	"malformed data, unable to verify the signature"
No public key or User's digital certificate or certificate URI parameters exist	400	`invalid_request`	"required parameter is missing, unable to verify the signature"
Only cert_x509 exists but value is empty [OR] Cert_x509 exists, but the value is invalid	400	`invalid_request`	"corrupted data, unable to verify the signature".
Only cert_x509_uri exists but value is empty [OR] Cert_x509_uri exists, but the value is invalid	400	`invalid_request`	"corrupted data, unable to verify the signature".
Only public_key exists but value is empty [OR] Public_key exists, but the value is invalid	400	`invalid_request`	"corrupted data, unable to verify the signature".

Table 15 : Errors – Server Initiated Mobile Connect PKI Authorise – Notification Acknowledgement

6.2.4 Polling Request Errors

Refer to [6] for all generic polling request errors.

Annex A Registration Authority (RA) and Certification Authority (CA)

A.1 Registration Authority (RA)

- The Registration Authority (RA) performs ID Proofing of a Mobile Connect User in accordance with policy (regulatory, operational), face-to-face or remote(online)[21], based on strong identity documents. The RA verifies the proofs and then captures any identity data needed for the PKI certificate[22] and issues a request for certification on behalf of the Mobile Connect User to the Certification Authority (CA). RA also manages certificate revocation requests.
- It is assumed that the Operator performs the role of Registration Authority (RA) for the MC Authorise PKI service given that the Operator will typically be performing ID proofing (KYC) checks anyway when issuing mobile contracts. An alternative is to register Users online using existing strong eID proofs. In some cases, there may be a second Registration Authority, such as the government, that validates RA authentication against a national ID scheme.

A.2 Certification Authority (CA)

- The Certification Authority (CA) creates and issues the digital certificate (X.509) for each User on the request of the RA and acts on any revocation requests from the RA. One CA can work with several RAs. The contractual relationship between RA and CA, mandates minimum requirements for RAs like security and procedures to perform verification and ID proofing. The CA is responsible for key backup and recovery. It also manages the cross-certification (certificate chain management) and validates the certificates when requested.
- Typically, the CA is responsible for maintaining a PKI repository that holds the certification Revocation Lists (CRLs) and certificates. It is usually an X.500/Light-weight Directory Access Protocol (X.500/LDAP) directory, but it could also be a Website.
- The CA is typically a 3rd party (e.g., VeriSign etc.,) that has been nationally accredited to act as a CA in accordance with local regulation although the Operator can also act as the Certification Authority. The CA may also be part of a hierarchy of CAs, the certificate chain helping to establish a much more scalable infrastructure – as illustrated below. This specification does not mandate 'who must be the CA' and it is left to Operator's implementation and infrastructure choice based on local market needs.

[21] Remote eKYC/eIDV ID proofing may also be possible dependent on local legislation and the requirements of the CA

[22] Depending on local regulation and policies, certificates may contain personal information of the User such as Customer name, National ID (Social security number) and Customer contact details

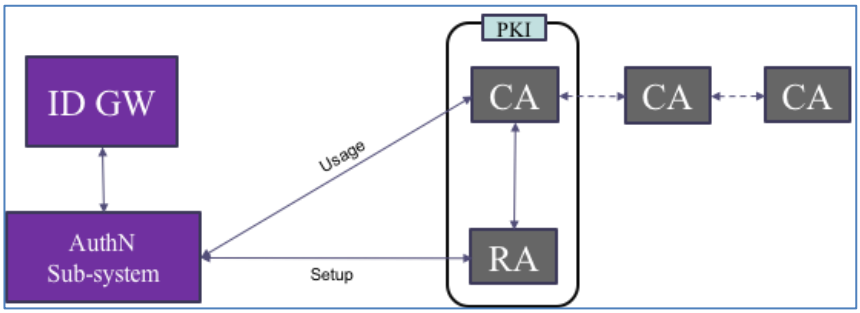

Figure 4. The relationship between Mobile Connect and other players in PKI.

An example setup/registration flow for the MC Authorise PKI service is shown below:

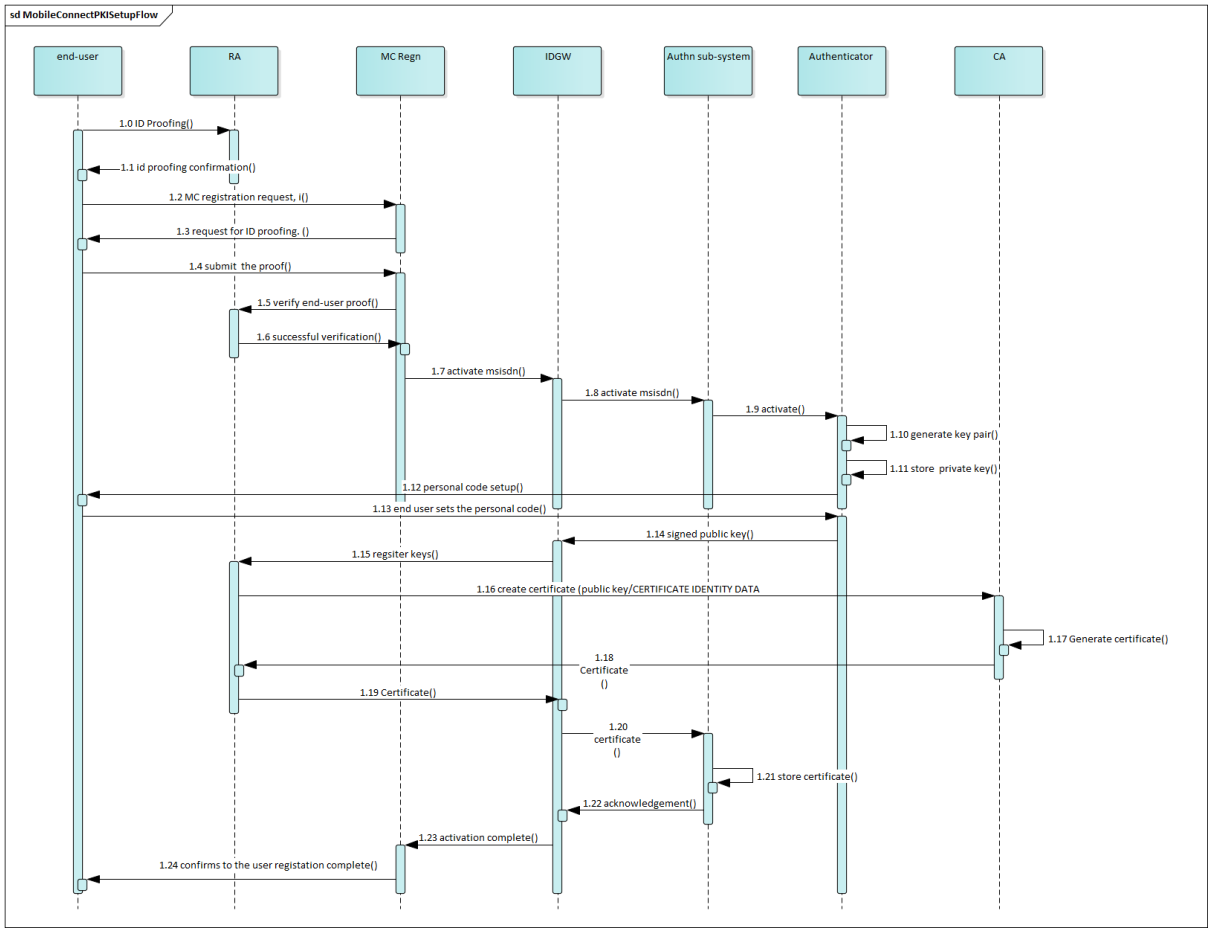

Figure 5: Mobile Connect PKI Authorise – Setup Flow

1. The RA entity conducts ID proofing and verification of the Mobile Connect User.
2. User registers for Mobile Connect by submitting the proof of ID verification received from the RA.
3. Upon successful registration the RA requests the CA to generate a digital certificate specific to the User.
4. After receiving the certificate, the RA sends the certificate to the Mobile Connect ID GW, which stores the certificate and associates with the User's MSISDN.
5. Mobile Connect ID GW registration system informs the User that the PKI based User registration is complete in an agreed communication method.

Annex B Document Management

B.1 Document History

Version	Date	Brief Description of Change	Approval Authority	Editor / Company
1.0	06/03/2019	New document	TG	V Boyalakuntla(Siva)/ GSMA
1.0	06/12/2022	Go throught TG approval	TG	Yolanda Sanz/GSMA

B.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.