European Payments Council

# Interoperability of MSCTs based on NFC or BLE

**EPC287-22 Version 0.18 / Date issued: 10 January 2023**

**Public**

© Copyright European Payments Council (EPC) AISBL
Reproduction for non-commercial purpose is authorised, with acknowledgement of the source

## Table of Contents

## List of tables



## List of figures

## Executive Summary

The ERPB invited the EPC in their Statement (ERPB/2021/028), published in November 2021, to broaden the scope of work on a QR-code standard (making sure to involve relevant stakeholders and standardisation bodies) to include other technologies, starting with Near-Field Communication (NFC) and continuing with Bluetooth Low Energy (BLE).

Subsequently, the EPC requested the Multi-stakeholder Group on Mobile Initiated SEPA (Instant) Credit Transfers (MSG MSCT – see Annex 3) to execute this work. The MSG MSCT established a dedicated work stream for this work in January 2022 following an open call for nominations on the EPC website. Their work-stream on Risk & Security has also been involved in the development of this document for the security related aspects.

For the development of this document the MSG MSCT leveraged the work included in the 2nd release of the *Mobile Initiated SEPA (instant) Credit Transfer Payments and Interoperability Guidance* (MSCT IG [10]) and the document on the *Standardisation of QR-codes for MSCTs* (EPC024-22, [14]).

The document includes illustrative MSCT use cases that employ NFC or BLE as proximity technology for the exchange of the necessary transaction information between the payer and payee to enable the initiation of an MSCT. Note that most of the MSCT use cases that employ BLE are based on the current work on eIDAS2.0. Next, both these proximity technologies are analysed in more detail, their usability for payments including the feedback received from some mobile payment service providers that tried to use these proximity technologies in the market, security aspects and main challenges to be addressed.

One of the major issues that has hindered the market take-up of NFC for mobile account-based payments was the difficulties encountered with the usage of NFC on some mobile platforms as described in section 6.2. It is expected that with the implementation of the newly published Digital Market Act [6], some of these obstacles will disappear over time.

In view of the lack of maturity of the usage of BLE for payments, the chapter on minimal data elements to be exchanged between the payer and the payee only addresses NFC based MSCTs.

The document further includes a dedicated chapter on the security of the data exchanged between the payer and the payee using NFC or BLE as proximity technologies.

Last but not least, the document identifies a number of opportunities towards standardisation of MSCTs based on NFC, subject to sufficient market interest to pursue this technology for account-based mobile payments.

In order to help developing a successful MSCT ecosystem that provides value for all, it is very important to gather industry opinion and market feedback regarding this document. Therefore a 10-week public consultation is launched before a final version of the document will be prepared. This final version will also be included into the third release of the MSCT IG (EPC269-19, [10]).

# 1 Document information

## 1.1 Structure of the document

This document contains a number of chapters and annexes, as follows:

Executive Summary;

Chapter 1 includes the document information;

Chapter 2 provides an introduction to the document;

Chapter 3 briefly discusses the interoperability model for MSCTs;

Chapter 4 describes MSCTs use cases based on NFC;

Chapter 5 describes MSCTs use cases based on BLE;

Chapter 6 analyses the proximity technologies NFC and BLE

Chapter 7 defines the minimum data sets to be exchanged between the payer and the payee using NFC;

Chapter 8 discusses the security aspects of data exchanged between the payer and the payee using NFC or BLE;

Chapter 9 discusses the potential topics to be addressed towards standardisation of MSCTs based on NFC;

Chapter 10 provides the conclusions;

Annex 1 provides a short introduction to eIDAS2.0;

Annex 2 lists the participants to the MSG MSCT Plenary;

Annex 3 lists the participants to the work-stream on interoperability of MSCTs based on NFC or BLE.

## 1.2 References

| N° | Title | Issued by |
|---|---|---|
| **[1]** | EBA/GL/2019/04: EBA Guidelines on ICT and security risk management | EBA |
| **[2]** | PSD2: Directive (EU) 2015/2366 of the European Parliament and of the Council of 25 November 2015 on payment services in the internal market, amending Directives 2002/65/EC, 2009/110/EC and 2013/36/EU and Regulation (EU) No 1093/2010, and repealing Directive 2007/64/EC | EC |
| **[3]** | Commission Delegated Regulation (EU) 2018/389 of 27 November 2017 supplementing Directive (EU) 2015/2366 with regard to regulatory technical standards for strong customer authentication and common and secure open standards of communication (also referred to as "RTS") | EC |
| **[4]** | General Data Protection Regulation (GDPR): Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the | EC |

| | | |
|---|---|---|
| | processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC | |
| **[5]** | eIDAS: Regulation (EU) No 910/2014 of the European parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC | EC |
| **[6]** | Digital Market Act: Regulation (EU) 2022/1925 of the European parliament and of the Council of 14 September 2022 on contestable and fair markets in the digital sector and amending Directives (EU) 2019/1937 and (EU) 2020/1828 | EC |
| **[7]** | EPC125-05: SEPA Credit Transfer Scheme Rulebook | EPC |
| **[8]** | EPC342-08: Guidelines on Cryptographic Algorithms Usage and Key Management | EPC |
| **[9]** | EPC004-16: SEPA Instant Credit Transfer Scheme Rulebook | EPC |
| **[10]** | EPC269-19v2.0 (2nd release): Mobile Initiated SEPA (Instant) Credit Transfer Payments and Technical Interoperability Guidance (MSCT IG) | EPC |
| **[11]** | EPC193-21v1.0: 2021 Payment Threats and Fraud Trends Report | EPC |
| **[12]** | EPC014-20: SEPA Request-to-Pay (SRTP) Scheme Rulebook | EPC |
| **[13]** | MSG MSCT045-21: Business requirements – Consumer selection of preferred payment instrument | EPC |
| **[14]** | EPC024-22: Standardisation of QR-codes for MSCTs | EPC |
| **[15]** | ERPB/2021/028: Statement following the sixteenth meeting of the ERPB held on 25 November 2021 | ERPB |
| **[16]** | ISO 12812: Core banking - Mobile financial services - Parts 1-5 | ISO |
| **[17]** | ISO 13616: Financial services - International Bank account number (IBAN) -- Part 1: Structure of the IBAN | ISO |
| **[18]** | ISO 18092: Information technology - Telecommunications and information exchange between systems -- Near Field Communication - Interface and Protocol (NFCIP-1) | ISO |
| **[19]** | ISO 20022: Financial Services – Universal Financial Industry Message Scheme | ISO |
| **[20]** | ISO TC 68 / SC 2 DIS 5201 : Financial services – Code scanning payment security – under ballot | ISO |
| **[21]** | ISO/IEC 18004: Information technology -- Automatic identification and data capture techniques -- QR-code bar code symbology specification | ISO |

| | | |
|---|---|---|
| **[22]** | ISO/IEC 14443: Identification cards - Contactless integrated circuit(s) cards - Proximity cards – Parts 1-4 | ISO |
| **[23]** | ISO/IEC 15417: Information technology — Automatic identification and data capture techniques — Code 128 bar code symbology specification | ISO |
| **[24]** | NFC Controller Interface (NCI) Specifications NFC Forum | NFC Forum |

Table 1: Bibliography

## 1.3 Terminology

| Term | Definition |
|---|---|
| **Account Servicing Payment Service Provider (ASPSP)** | A PSP providing and maintaining a payment account for a payer (see Article 4 in [2]) or a payee. |
| **Alias** | See Proxy |
| **Beneficiary** | See Payee. |
| **Bluetooth Low Energy (BLE)** | A wireless personal area network technology designed and marketed by the Bluetooth Special Interest Group aimed at novel applications including beacons. Compared to classic Bluetooth, BLE is intended to provide considerably reduced power consumption and cost while maintaining a similar communication range. |
| **Collecting Payment Service Provider (CPSP)** | A payment service provider according to PSD2 that collects the payment transactions on behalf of the merchant (the ultimate beneficiary) and as such is the beneficiary of the IP at POI transaction. |
| **Consumer** | A natural person who, in payment service contracts covered by the PSD2, is acting for purposes other than his or her trade, business or profession (see Article 4 in [2]). |
| **Consumer Device** | An internet capable device used by the consumer to conduct an instant payment. Examples include a mobile device or a personal computer (PC). |
| **Consumer Device UVM (CDUVM)** | A user verification method (UVM) entered by or captured from the consumer (user) on the consumer device (e.g. a mobile device) (see [16]). |
| **Consumer-presented data** | Data provided by the consumer at the merchant's POI. |
| **Countersignature** | These are signatures that are applied one after the other and are used where the order in which the signatures are applied is important. In these situations, the first signature signs the signed document/message. Each additional signature can sign in turn the |

| | |
|---|---|
| | latest previously generated signature, or all the previously generated signatures together with the signed document/message. |
| **Credit transfer** | A payment service for crediting a payee's payment account with a payment transaction or a series of payment transactions from a payer's payment account by the PSP which holds the payer's payment account, based on an instruction given by the payer (see (see Article 4 in [2]). |
| **Credit Transfer instruction** | A payment instruction given by an originator to an originator ASPSP requesting the execution of a credit transfer transaction, comprising such information as is necessary for the execution the credit transfer and is directly or indirectly initiated in accordance with the provisions of [2]. |
| **Credit Transfer Transaction** | An instruction executed by an originator ASPSP by forwarding the transaction to a CSM for forwarding the transaction to the beneficiary ASPSP. |
| **Customer** | A payer or a beneficiary which may be either a consumer or a business (merchant). |
| **CustomerID** | In the context of this document, an identification of the payer (consumer), issued by their ASPSP for access to (a) customer facing user interface(s) (e.g. their on-line banking system), as required in the PSD2 API. |
| **2D barcode** | A two-dimensional barcode is a machine-readable optical label that contains digital information. They are also referred to as matrix barcodes. Examples include QR codes and tag barcodes. |
| **Digital wallet** | A service accessed through a consumer device which allows the wallet holder to securely access, manage and use a variety of services/applications including payments, identification and non-payment applications (e.g., value added services such as loyalty, couponing, etc.). A digital wallet is sometimes also referred to as an e-wallet. |
| **Electronic identification** | The process of using personal identification data in electronic form uniquely representing either a natural or legal person, or a natural person representing a legal person. |
| **Elliptic-Curve Diffie–Hellman (ECDH)** | A key agreement protocol that allows two parties, each having an elliptic-curve asymmetric key pair (consisting of a private and a public key), to establish a shared secret over an insecure channel. This shared secret may be directly used as a key, or to derive another secret key. The secret key, or the derived key, can subsequently be used to encrypt communications using symmetric key cryptography. |

| | |
|---|---|
| **EMVCo** | An LLC formed in 1999 by Europay International, MasterCard International and Visa International to enhance the EMV Integrated Circuit Card Specifications for Payments Systems. It manages, maintains, and enhances the EMV specifications jointly owned by the payment systems. It currently consists of American Express, Discover, JCB, MasterCard, Union Pay and VISA. |
| **Funds** | Cash, scriptural money or electronic money as defined in (see Article 4 in [2]). |
| **HUB** | An infrastructure ensuring connectivity between IP service providers. The term HUB is meant to be agnostic to the way it might be implemented – logically or physically - different models may be possible, but it should at least cover (a kind of) routing service. As an example, this could be a direct connection amongst IP service providers through a dedicated API. |
| **IBAN attribute certificate** | This is a payment means attribute attestation. This payment means attestation contains the IBAN and may contain other information to support payments, such as the Host URI for PSD2 Open Banking or the BIC of the ASPSP that holds the payment account (see also Annex 1). |
| **Instant(ly)** | At once, without delay. |
| **Instant Payment** | Electronic retail payment solutions available 24/7/365 and resulting in the immediate or close-to-immediate interbank clearing of the transaction and crediting of the payee's account with confirmation to the payer (within seconds of payment initiation) (see [9]). |
| **International Bank Account Number (IBAN)** | An internationally agreed system of identifying bank accounts across national borders to facilitate the communication and processing of cross border transactions (see [17]). |
| **Instant Payment (IP) Application** | A set of modules (application software) and/or data (application data) needed to provide functionality for an Instant Payment (IP) as specified by the IP service provider in accordance with the SEPA Instant Credit Transfer scheme. |
| **MSCT Service Provider** | A service provider that offers or facilitates an MSCT service to a payer and/or payee based on an SCT Instant or SCT payment transaction. This may involve the provision of a dedicated MSCT application for download on the payer's device or the provision of dedicated software for the merchant POI. As an example, an MSCT service provider could be a PSP (e.g. an ASPSP or any party acting as a PISP under PSD2) or a technical service provider supporting a PSP. |
| **Merchant** | A beneficiary within a payment scheme for payment of the goods or services purchased by the consumer. The merchant is a customer of their PSP. A merchant may also be referred to as payee. |

| | |
|---|---|
| **Merchant-presented data** | Data provided by the merchant's POI to the consumer. |
| **Mobile code** | An authentication credential used for user verification and entered by the consumer via the keyboard of the mobile device. |
| **Mobile device** | Personal device with mobile communication capabilities such as a telecom network connection, Wi-Fi, Bluetooth, etc.<br>Examples of mobile devices include mobile phones, smart phones, tablets, wearables, car on-board units. |
| **Mobile Network Operator (MNO)** | A mobile phone operator that provides a range of mobile services, potentially including facilitation of NFC services. The MNO ensures connectivity Over the Air (OTA) between the consumer and their PSP using their own or leased network. |
| **Mobile payment service** | A payment service made available by software/hardware through a mobile device. |
| **Mobile service** | A service such as identification, payment, ticketing, loyalty, etc., made available through a mobile device. |
| **Mobile wallet** | A digital wallet accessed through a mobile device. This service may reside on a mobile device owned by the consumer (i.e. the holder of the wallet) or may be remotely hosted on a secured server (or a combination thereof) or on a merchant website. Typically, the so-called mobile wallet issuer provides the wallet functionalities but the usage of the mobile wallet is under the control of the consumer. |
| **Mutual Authentication** | This refers to two parties authenticating each other at the same time using an authentication protocol (also referred to as two-way authentication). |
| **NFC (Near Field Communication)** | A contactless protocol for mobile devices specified by the NFC Forum for multi-market usage. NFC Forum specifications (see [24]) are based on ISO/IEC 18092 [18] but have been extended for harmonisation with EMVCo and interoperability with ISO/IEC 14443 [22] . |
| **Originator** | See Payer. |
| **Payee** | A natural or legal person who is the intended recipient of funds which have been the subject of a payment transaction (see Article 4 in [2]), (examples include merchant, business). |
| **Payee Reference Party** | A person/entity on behalf of or in connection with whom the payee receives a payment. |

| | |
|---|---|
| **Payer** | A natural or legal person who holds a payment account and allows a payment order from that payment account, or, where there is no payment account, a natural or legal person who gives a payment order (see Article 4 in [2]). |
| **Payment account** | An account held in the name of one or more payment service users which is used for the execution of payment transactions (see Article 4 in [2]). |
| **Payment Initiation Service Provider (PISP)** | A payment service provider pursuing business activities as referred to in Annex I.7 of [2]. |
| **Payment Request** | Set of rules and technical elements (including messages) that allow a payee to claim an amount of money from a payer for a specific transaction. As an example, see [12]. |
| **Payment Request message** | Message sent by the payee to the payer, directly or through agents. It is used to request the movement of funds from the payer account to the beneficiary account. |
| **Payment Service Provider (PSP)** | An entity referred to in Article 1(1) of [2] or a natural or legal person benefiting from an exemption pursuant to Article 32 or 33 of [2]. |
| **Payment Service User (PSU)** | A natural or legal person making use of a payment service in the capacity of payer, payee, or both (see Article 4 in [2]). |
| **Payment scheme** | A technical and commercial arrangement (often referred to as the "rules") between parties in the payment value chain, which provides the organisational, legal and operational framework rules necessary to perform a payment transaction. |
| **Payment system** | A funds transfer system with formal and standardised arrangements and common rules for the processing, clearing and/or settlement of payment transactions (see Article 4 in [2]). |
| **Payment transaction** | An act, initiated by the payer or on his/her behalf or by the payee (beneficiary), of placing, transferring or withdrawing funds, irrespective of any underlying obligations between the payer and the payee (see Article 4 in [2]). |
| **Personal data** | Any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person (see [4]). |
| **Person Identification Data (PID) Provider** | A service provider in the context of eIDAS2.0 who verifies the identity of the EUDIW user, maintain an interface to provide PID |

| | |
|---|---|
| | securely to the EUDI Wallet (in a harmonised common format) and make available information for relying parties to verify the validity of the PID, without having an ability to receive any information about the use of the PID. The terms and conditions of these services would be for each EU Member State to determine.<br><br>PID providers may for instance be the governmental bodies which issue today official identity documents, electronic identity means, EUDIW issuers etc. (see also Annex 1).<br><br>PID providers may or may not be the same bodies as the EUDIW issuers. |
| **Physical POI** | A POI that is a physical device and consists of hardware and software, hosted in acceptance equipment to enable a consumer and/or merchant to perform an MCST. The merchant-controlled POI may be attended or unattended. Examples of POI include Point-of-Sale (POS), vending machine. |
| **Point of Interaction (POI)** | The initial point in the merchant's environment (e.g. POS, vending machine, payment page on merchant website, QR-code on a poster, etc.) where data is exchanged with a consumer device (e.g., mobile phone, wearable, etc.) or where consumer data is entered to initiate an instant credit transfer. |
| **Proximity Payment** | A payment where the consumer and the merchant (and/or their equipment) are in the same location and where the communication between the mobile device and the Point of Interaction device takes place through a proximity technology (e.g., NFC, 2D barcodes, BLE, ultrasonic, etc.). |
| **Proxy** | Data required in order to retrieve a payment account identifier (e.g., mobile phone number, e-mail address, etc.). This is sometimes referred to as an "alias". As an example, a proxy could be used to replace an IBAN which will be referred to as IBAN-proxy in this document. |
| **QR-code** | Quick Response-code [21], see also 2D barcode. |
| **RequestID** | This is an identifier that allows a wallet/mobile app to establish a link to the correct relying party request. For example it could be an online checkout session or it could be used to identify the correct POS terminal or even a table in a restaurant. This ID can be ephemeral (one time) or it can be fixed (contained within a printed physical QR-code attached to a table). |
| **Secure Element (SE)** | A tamper-resistant platform (typically a one chip secure microcontroller) capable of securely hosting applications and their confidential and cryptographic data (e.g., key management) in accordance with the rules and security requirements set forth by a set of well-identified trusted authorities. |

| | |
|---|---|
| | There are different form factors of SE including Universal Integrated Circuit Card (UICC), embedded SE (including eUICC and iSE) and microSD. Both the UICC and microSD are removable. |
| **SEPA Credit Transfer** | The SEPA Credit Transfer is the payment instrument governed by the rules of the SEPA Credit Transfer Scheme for making credit transfer payments in euro throughout the SEPA from payment accounts to other payment accounts (see [9]). |
| **SEPA Instant Credit Transfer** | The SEPA Instant Credit Transfer is the payment instrument governed by the rules of the SEPA Instant Credit Transfer Scheme for making instant credit transfer payments in euro throughout the SEPA from payment accounts to other payment accounts (see [9]). |
| **Single Euro Payments Area (SEPA)** | The countries and territories which are part of the jurisdictional scope of the SEPA payment schemes<br>(see https://www.europeanpaymentscouncil.eu/document-library/other/epc-list-sepa-scheme-countries). |
| **Tokenisation** | Process of substituting payment account, PSU identification data or transaction related data with a surrogate value, referred to as a token. |
| **Token** | Tokens can take on a variety of formats across the payments industry. They generally refer to a surrogate value for payment account (e.g., the IBAN), PSU identification data (e.g., CustomerID) or transaction related data. Payment Tokens must not have the same value as or conflict with the real payment account related data. If the token is included in the merchant-presented data it might be referred to as a merchant token; if the token is included in the consumer-presented data it might be referred to as a consumer token. |
| **Token Requestor** | An entity requesting a token to the Token Service |
| **Token Service** | A system comprised of the key functions that facilitate generation and issuance of tokens and maintain the established mapping of tokens to the related data when requested by the token requestor. It may also include the capability to establish the token assurance level to indicate the confidence level of the payment token to the related information binding. The service also provides the capability to support token processing of payment transactions submitted using tokens by de-tokenising the token to obtain the actual related information (see also the definition of Token). |
| **Token Service Provider (TSP)** | An entity that provides a Token Service. |

| | |
|---|---|
| **Trusted certificate** | A trusted certificate (normally a root certificate) available on the EU Trusted List (https://esignature.ec.europa.eu/efda/tl-browser/#/screen/home)<br>A verifier sources the trusted certificate from the EU Trusted List and is then able to verify the full certificate chain of a counterparty certificate. The trusted certificate establishes a root of trust and allows the verifier to trust the counterparty certificate and corresponding private key. |
| **Trusted Execution Environment (TEE)** | A separate execution environment that runs alongside, but isolated from the main operating system. A TEE has security capabilities and meets certain security-related requirements: it protects TEE assets from general software attacks, defines rigid safeguards as to data and functions that a program can access, and resists a set of defined threats. |
| **Trusted Third Party (TTP)** | An entity which facilitates interactions between stakeholders of the ecosystem who all trust this third party (examples are SE provider, common infrastructure manager…). |
| **Ultra-Wide Band (UWB)** | A radio technology that can use a very low energy level for short-range, high-bandwidth communications over a large portion of the radio spectrum. Most recent applications target sensor data collection, precise locating, and tracking. UWB support starts to appear in high-end mobile phones. |
| **Uniform Resource Identifier (URI)** | A unique sequence of characters that identifies a logical or physical resource used by web technologies. |
| **White Box Cryptography (WBC)** | A cryptographic technique that combines methods of encryption and obfuscation to embed secret keys within application code. The goal is to combine code and keys in such a way that the two are indistinguishable to an attacker, and the new "white-box" program can be safely run in an insecure environment. |

Table 2: Terminology

## 1.4 Abbreviations

| Abbreviation | Term |
|---|---|
| **an** | alphanumeric |
| **ASPSP** | Account Servicing PSP |
| **API** | Application Programming Interface |
| **B2B** | Business-to-Business |
| **BLE** | Bluetooth Low Energy |
| **C2B** | Consumer-to-Business |

| | |
|---|---|
| **CDUVM** | Consumer Device UVM |
| **CEN** | European Committee for Standardisation |
| **CPSP** | Collecting Payment Service Provider |
| **CSM** | Clearing and Settlement Mechanism |
| **2D barcode** | Two dimensional barcode |
| **EBA** | European Banking Authority |
| **EC** | European Commission |
| **ECDH** | Elliptic-Curve Diffie–Hellman |
| **ECSG** | European Cards Stakeholders Group |
| **EPC** | European Payments Council |
| **EPI** | European Payments Initiative |
| **EPIF** | European Payment Institutions Federation |
| **ERPB** | Euro Retail Payments Board |
| **ETPPA** | European Third Party Providers Association |
| **GDPR** | General Data Protection Regulation |
| **IBAN** | International Bank Account Number |
| **ID** | Identifier |
| **IP** | Instant Payment |
| **ISO** | International Organization for Standardization |
| **MNO** | Mobile Network Operator |
| **MSCT** | Mobile Initiated (Instant) SCT |
| **MSCT IG** | Mobile Initiated SEPA (Instant) Credit Transfer Payments and Technical Interoperability Guidance |
| **MSG MSCT** | Multi-Stakeholder Group for Mobile Initiated (Instant) SCT |
| **n** | numeric |
| **NFC** | Near-Field Communication |
| **P2P** | Person-to-Person |
| **PID** | Person Identification Data |
| **PISP** | Payment Initiation Service Provider |
| **POI** | Point of Interaction |
| **POS** | Point of Sale |

| | |
|---|---|
| **PSD** | Payment Services Directive |
| **PSP** | Payment Service Provider |
| **PSU** | Payment Service User |
| **QR-code** | Quick Response-code |
| **RTS** | Regulatory Technical Standard |
| **SCT Inst** | SEPA Instant Credit Transfer |
| **SE** | Secure Element |
| **SEPA** | Single Euro Payments Area |
| **SP** | Service Provider |
| **TC** | Technical Committee |
| **TEE** | Trusted Execution Environment |
| **TSP** | Token Service Provider |
| **TTP** | Trusted Third Party |
| **URI** | Uniform Resource Identifier |
| **URL** | Uniform Resource Locator |
| **UVM** | User Verification Method |
| **UWB** | Ultra-Wide band |
| **WBC** | White Box Cryptography |

Table 3: Abbreviations

## 2 Introduction

This document has been developed by the Multi-stakeholder Group on Mobile Initiated SEPA (Instant) Credit Transfers (MSG MSCT) according to the extension of their mandate MSG MSCT 128-21 and to address the invitation made to the EPC by the ERPB included in the ERPB Statement published in November 2021 (see [15]), namely to broaden the scope of the standardisation of MSCTs to other technologies than QR-codes, starting with NFC and continuing with BLE.

The development of the document involved a dedicated work-stream (WS) that was established in January 2022, following an open call for nominations on the EPC website, and the WS Risk and Security for the security related aspects.

For the development of this document the MSG MSCT leveraged the work included in the 2nd release of the Mobile Initiated SEPA (Instant) Credit Transfer Payments and Technical Interoperability Guidance (MSCT IG [10]) and the document on the Standardisation of QR-codes for MSCTs (EPC024-22 - [14].

They started with the identification of MSCT use cases that employ NFC or BLE as proximity technology for the exchange of the necessary transaction information between the payer and the payee to enable the initiation of an MSCT. Note that the MSCT use cases that employ BLE are based on the current work on eIDAS2.0. The MSG MSCT analysed both these proximity technologies in more detail, their usability for payments including the feedback received from some mobile payment service providers that tried to use these proximity technologies in the market, security aspects and main challenges to be addressed.

In view of the lack of maturity of the usage of BLE for payments, the chapter on minimal data elements to be exchanged between the payer and the payee only addresses NFC based MSCTs.

The document further includes a dedicated chapter on the security of the data exchanged between the payer and the payee using these proximity technologies.

Last but not least, the document identifies a number of opportunities towards standardisation, subject to sufficient market interest to pursue the NFC technology for account-based mobile payments.

In order to help developing a successful MSCT ecosystem that provides value for all, it is very important to gather industry opinion and market feedback regarding this document. Therefore an 8-week public consultation is launched before a final version of the document will be prepared. This final version will also be included into the third release of the MSCT IG (EPC269-19, [10]).

# 3 Interoperability of MSCTs

MSCTs are initiated directly (by the payer) or indirectly (by an MSCT service provider at the request of the payer) in compliance with the PSD2 (see [9]), using a mobile device. MSCT solutions are offered by so-called MSCT service providers which are service providers that offer or facilitate a payment service to a payer/payee based on an SCT Instant or an SCT transaction. As an example, an MSCT service provider could be a PSP (e.g. an ASPSP or any party acting as a PISP under PSD2) or a technical service provider supporting a PSP.

MSCTs in Euro are based on the existing SCT Instant scheme or SCT Scheme rulebooks (see [9] and [7] resp.) in the so-called "inter-PSP space" and are therefore using in that space the existing payment infrastructure. They typically use an MSCT application or a browser on the payer's mobile device to initiate or at least authenticate and authorise the SCT (Instant) transaction, besides some features of the mobile device such as the support of CDUVM (e.g., a mobile code or biometrics on the mobile device), the mobile device screen to display transaction information, etc.

For the analysis of the technical interoperability of MSCTs, the following generic 4-corner model was used in the MSCT IG [10]. Hereby it is assumed that both payer and payee have different ASPSPs that are SCT Inst or SCT scheme participants (see Chapter 4 in [10]), while the entities assuming the role of MSCT service provider are depicted as separate entities that are different for the payer and the payee. Obviously, if the role of MSCT service provider would be assumed by an ASPSP the model below would simplify. Alternatively, multiple PSPs (such as a PISP licensed under PSD2 or a CPSP) could be involved between the payer/payee and their respective ASPSP; these models have been studied in Chapter 20 of the MSCT IG [10].

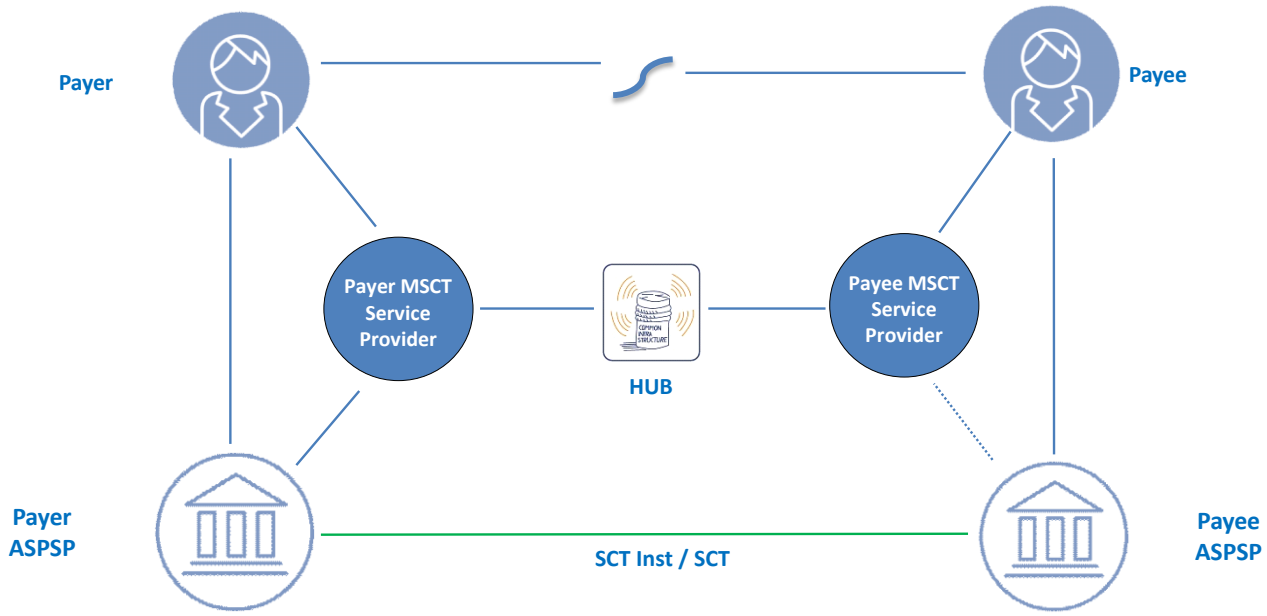



Figure 1: Generic 4-corner interoperability model for MSCTs

As depicted above, the payer's MSCT service provider is linked to the payer's ASPSP and the payee's MSCT service provider may be linked to the payee's ASPSP (this linkage may include both technical and contractual aspects).

The MSCT ecosystem involves some other new stakeholders in the value chain compared to the ones described in the SCT Inst or SCT scheme rulebooks (see [9] and [7] resp.) including a so-called Token Service Provider (TSP) who is a TTP involved if tokens are used in MSCTs as surrogate values for the transaction data (including the merchant/consumer IBAN, merchant/consumer identifier, transaction amount or merchant transaction identifier). The TSP manages the generation and issuance of tokens, and maintains the established mapping of tokens to the related transaction data. For simplification it is assumed in this document that the role of the TSP is assumed or is under the control of the MSCT service provider (and hence the TSP is not depicted in the figure above)[1].

To achieve interoperability for the generic basic 4-corner model, the concept of a HUB was introduced to interconnect the respective MSCT service providers as shown in the figure above. Hereby the term HUB is used to indicate an "infrastructure" that enables interconnectivity between MSCT service providers but it is meant to be agnostic to the way it might be implemented – different implementation models may be possible (centralised or de-centralised (e.g. a direct API)).

The technical interoperability requirements between MSCT service providers have been analysed and defined in detail in Chapters 16 through 20 in the MSCT IG [10]. One of the interoperability aspects is the exchange of (transaction) data between the payer and the payee to enable the initiation of an MSCT. The usage of NFC or BLE as proximity technologies for this data exchange will be treated in the next chapters.

[1] The same is valid in case of usage of a proxy. The role of the provider involved is assumed or is under the control of the MSCT service provider.

# 4 MSCTs use cases based on NFC

## 4.1 Introduction

This chapter is devoted to MSCTs whereby a proximity technology for the data exchange between the payer and the payee is used to enable the initiation of an MSCT, as defined in the MSCT IG [10]. In a similar way as with QR-codes, data may be exchanged using *uni-directional NFC*. This document will focus on MSCTs based on payee-presented data whereby the data refers to payee identification data and transaction data.

The NFC technology may also be used in a bi-directional mode, in a similar way as with mobile contactless card-based payments.

In this chapter a number of MSCT use cases will be described with a diagram depicting the different actors involved and with a decomposition into the different steps of the MSCT transaction which are also shown in a figure. Each MSCT use case is followed by a short evaluation on the interoperability aspects for deployment across SEPA and compliance with the PSD2 [2] and the RTS [3], including a short list of the main challenges.

Note that these MSCT use cases are presented for illustrative purposes; in other words, the list of MSCT use cases described is not meant to be exhaustive but should be seen as examples for specific payment contexts. Likewise, the authentication method used is purely illustrative. More details on payer identification and SCA are provided in the sections 8.2 and 8.3 in the MSCT IG [10].

It is further to be noted that similar as in the MSCT IG , for the MSCT use cases involving a token, the role of the TSP is covered by the MSCT service provider/ASPSP or is at least under their control.

| Payment context | # | MSCT use case description |
|---|---|---|
| **Person-to-Person (P2P) payments** | P2P-1 | Mobile device – Payee-presented data using uni-directional NFC – MSCT app involving a fingerprint |
| **Consumer-to-Business (C2B) payments** | C2B-1 | Mobile device - Payment at a physical POI using uni-directional NFC – merchant-presented data - MSCT app – SCA involving a mobile code |
| | C2B-2 | Mobile device - Offline use case – Payment at a physical POI using bi-directional NFC – single tap - MSCT application - SCA involving facial recognition |
| | C2B-3 | Mobile device - Offline use case – Payment at a physical POI using bi-directional NFC – double tap - MSCT application - SCA involving a mobile code |
| | C2B-4 | Mobile device - Offline use case – Payment at a physical POI using bi-directional NFC and EMV-based SCA involving a fingerprint |

Table 4: Overview illustrative MSCT use cases using NFC

Note that the term “offline” in the table above refers to the payer whereby no mobile network connectivity for their mobile device is required to conduct the transaction.

### 4.2 MSCT use case P2P-1: Mobile device – Payee-presented data using uni-directional NFC – - MSCT app involving a fingerprint

This MSCT use case presents an example for a person-to-person payment based on payee-presented data and relying on the usage of uni-directional NFC technology for the exchange of this data from the payee's mobile device to the payer's mobile device.

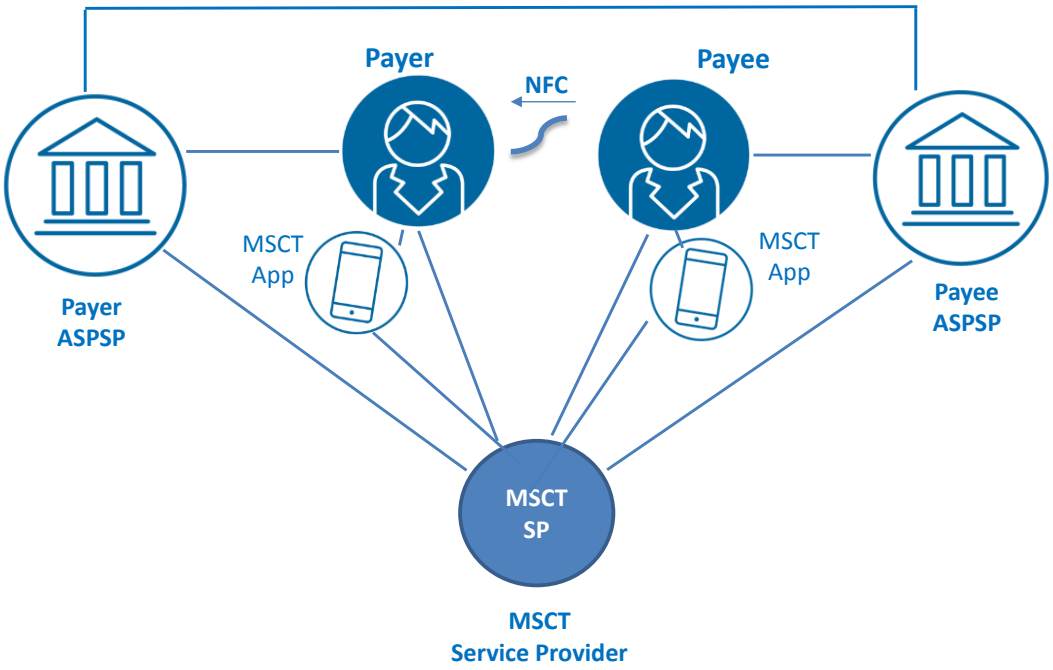



Figure 2: Actors in MSCT Use case P2P-1

Payer and payee may, and frequently will, hold their payment accounts with different ASPSPs. In this example, both ASPSPs are registered with the same MSCT Inst service provider.

In this payment transaction a strong customer authentication (see section 8.3 in the MSCT IG [10]) in accordance with the relevant PSD2 [2] requirements is performed involving a fingerprint (see section 8.2) in the MSCT IG [10]) and the calculation of an authentication code by the MSCT application using a dedicated key. If the MSCT application is provided to the payer by an MSCT service provider instead of the payer's ASPSP, a delegation for payer authentication from the payer's ASPSP to their MSCT service provider is needed. However, this requires an agreement between the payer's ASPSP and the MSCT service provider.

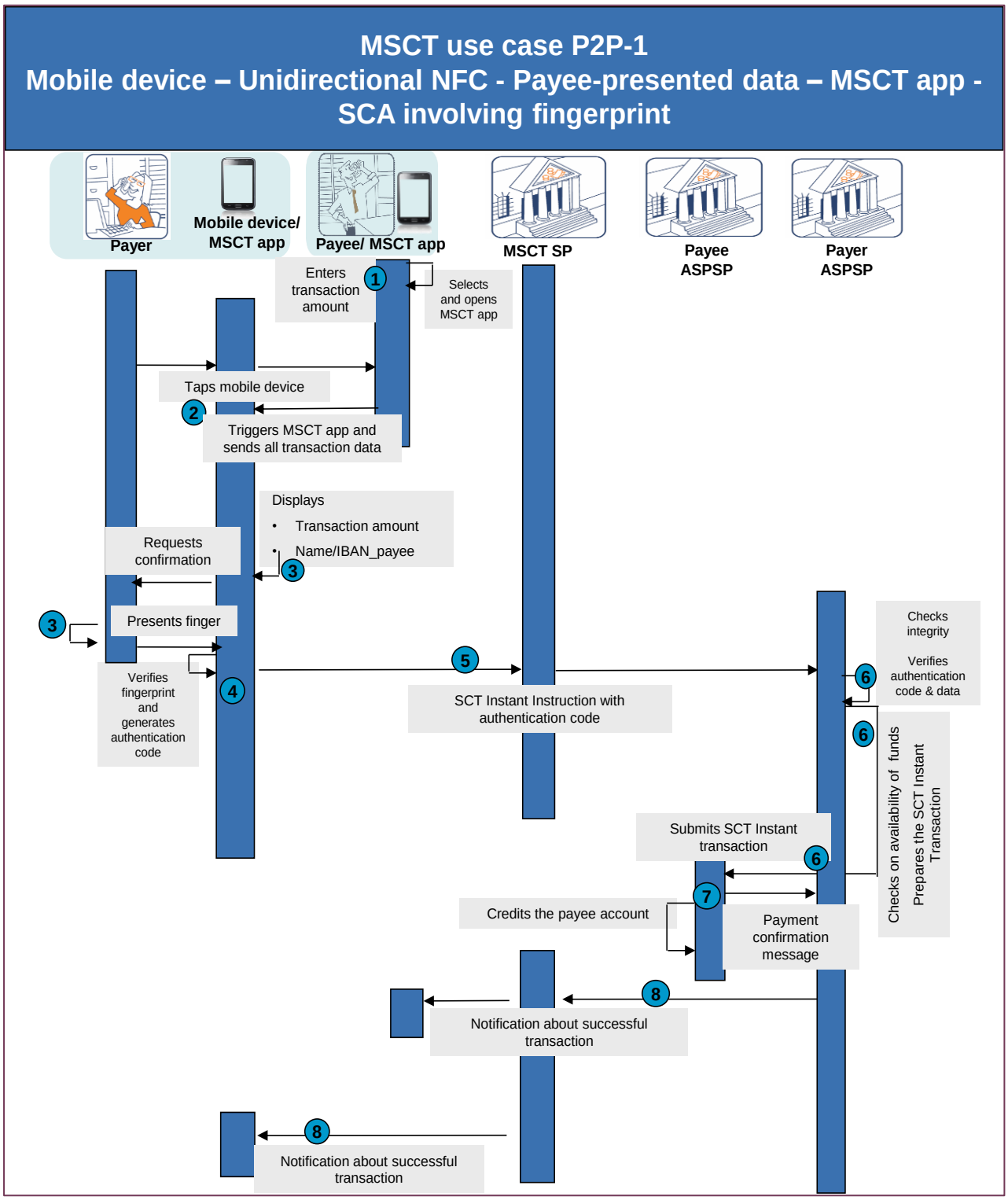



Figure 3: MSCT Use case P2P-1

In the figure above, the following steps are illustrated:

**Step 0**

- Both the payer and payee need to be subscribed to the same MSCT service and need to have downloaded a dedicated MSCT application from the MSCT Inst service provider on their mobile device.
- The MSCT service provider needs to be linked to both ASPSPs.
- During the payment transaction, a mobile internet connection is required.

**Step 1**

- The payee selects and opens the MSCT app on their mobile device which possibly involves the entry of a password.
- Next, the payee selects in the MSCT app the IBAN_payee on which they would like to receive the transfer and enters the transaction amount.

**Step 2**

- The payee taps their mobile device to the payer's mobile device which through NFC triggers the MSCT app on the payer's mobile device and sends all transaction data in clear text to the payer's MSCT app, including the payee's name, IBAN_payee, transaction amount and transaction identifier.

**Step 3**

- The MSCT application on the payer's mobile device pops-up a window with the transaction details including the payee name/IBAN_payee and transaction amount.
- The payer authenticates and confirms the transaction by presenting a fingerprint to the mobile device.

**Step 4**

- Upon successful verification of the fingerprint by the mobile device, an authentication code is calculated by the MSCT app, which is linked to the payee and the transaction amount.

**Step 5**

The SCT Inst instruction, including the payee's name, IBAN_payee, the transaction amount, the transaction identifier and the authentication code are transmitted to the payee's ASPSP via the MSCT service provider.

**Step 6**

- The payer's ASPSP checks the integrity of the SCT Inst instruction and verifies the authentication code.
- The payer's ASPSP checks the availability of funds on the payer's account,
- The payer's ASPSP prepares and submits the SCT Inst transaction to the payee's ASPSP.

**Step 7**

- A confirmation message is returned from the payee's ASPSP to the payer's ASPSP.
- The payee's ASPSP makes the funds available to the payee.

**Step 8**

- The payee is notified by the MSCT service provider (information provided by the payer's ASPSP) that their account has been credited.
- The payer is notified by the MSCT service provider that the payment has been successfully executed (information provided by the payer's ASPSP) and may optionally receive an e-receipt.

| Analysis MSCT Use case P2P-1 | |
|---|---|
| **Interoperability** | • The payer and the payee need to be subscribed to the same MSCT service<br>• The payee's ASPSP and the payer's ASPSP need be linked to the same MSCT service.<br>• For a truly "open" approach and a SEPA-wide interoperability, if the MSCT service provider of the payer is different to the MSCT service provider of the payee, a framework needs to be specified that interconnects the different MSCT service providers. |
| **Challenges** | • Standardisation of payee data transmitted from the payee's mobile device to the payer's mobile device.<br>• Education of PSU on usage of NFC for MSCTs.<br>• The notification messages in step 8 are not included in the SCT Inst scheme. |

Table 5: Analysis MSCT use case P2P-1

*Notes:*

- The interoperability in case different MSCT service providers are involved for the payer and the payee is addressed in Chapters 16 and 17 in the MSCT IG [10]. In this case, a process flow similar as shown in Figure 48 in the MSCT IG applies.

- The minimum data elements in the notification messages are specified in Annex 5 of the MSC IG [10].

### 4.3 MSCT use case C2B-1: Mobile device - Payment at a physical POI using uni-directional NFC – merchant-presented data - MSCT app – SCA involving a mobile code

This MSCT use case presents an example for an in-store payment based on merchant-presented data and relying on the usage of uni-directional NFC technology for the exchange of this data between the consumer mobile device and the merchant POI.

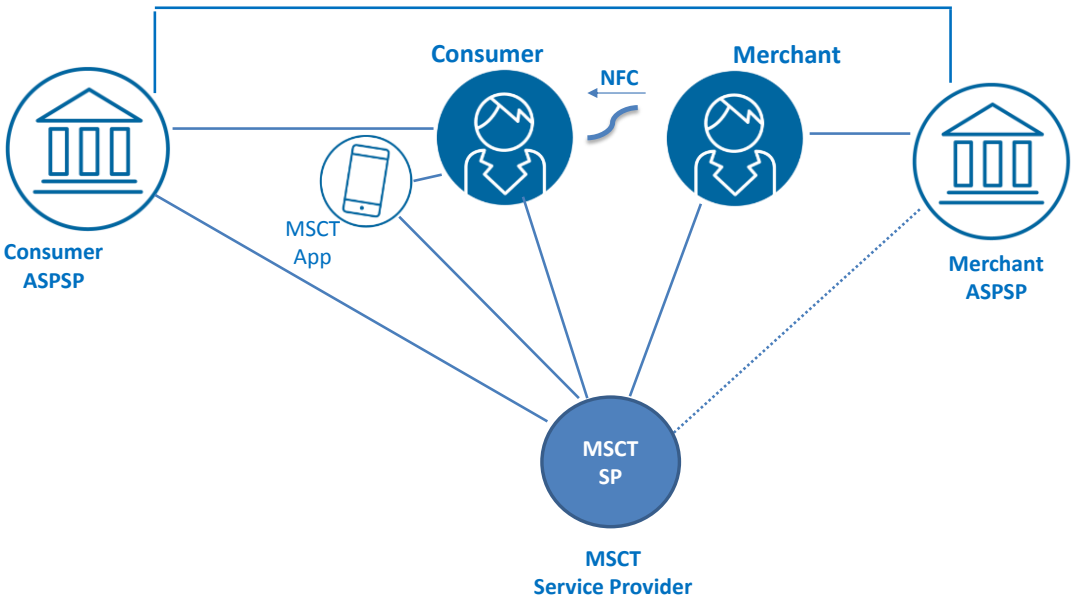



**Figure 4: Actors in MSCT Use case C2B-1**

Consumer and merchant, may, and frequently will, hold their payment accounts with different ASPSPs. In this example, both ASPSPs are registered with the same MSCT Inst service provider.

In this payment transaction a strong customer authentication (see section 8.3 in the MSCT IG [10]) in accordance with the relevant PSD2 [2] requirements is performed involving a mobile code (see section 8.2 in the MSCT IG [10]) and the calculation of an authentication code by the MSCT application using a dedicated key. If the MSCT application is provided to the consumer by an MSCT service provider instead of the consumer's ASPSP, a delegation for payer authentication from the consumer's ASPSP to their MSCT service provider is required. However, this requires an agreement between the consumer's ASPSP and the consumer's MSCT service provider.

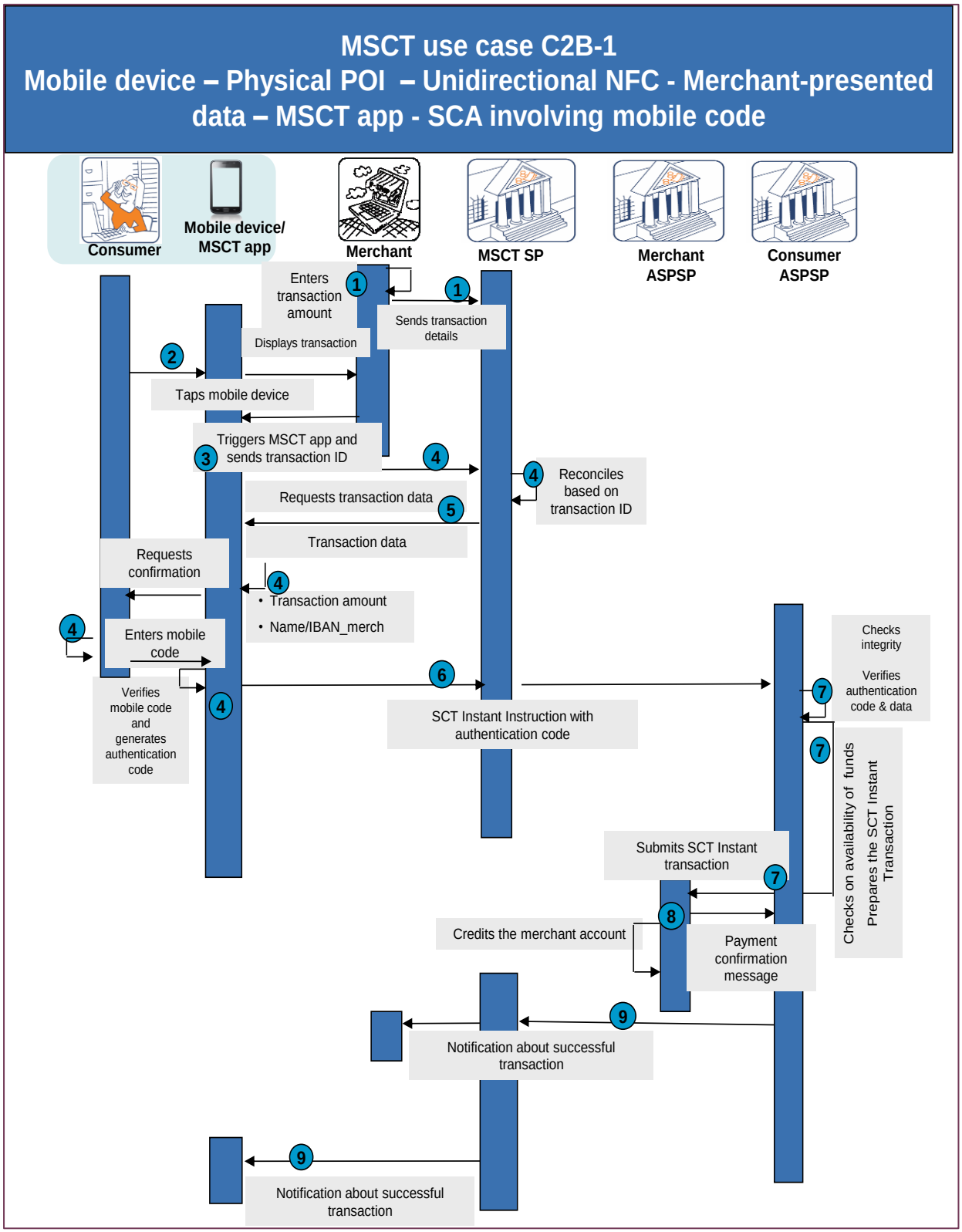



Figure 5: MSCT Use case C2B-1

In the figure above, the following steps are illustrated:

**Step 0**

- The consumer needs to be subscribed to an MSCT Inst service and needs to have downloaded a dedicated MSCT Inst application from the MSCT Inst service provider, linked to a specific payment account of their ASPSP.
- The merchant needs to be subscribed to the same MSCT Inst service with a specific account from their ASPSP and have installed a dedicated NFC protocol (app) on their POI.
- The MSCT service provider needs to be linked to both ASPSPs.
- During the payment transaction, a mobile internet connection is required.

**Step 1**

- The merchant enters the transaction amount on the POI.
- The POI provides the transaction details to the MSCT service provider.
- The transaction amount is displayed on the merchant's POI.

**Step 2**

- The consumer taps their mobile device to the merchant’s POI.

**Step 3**

- The POI through NFC triggers the MSCT application on the consumer’s mobile device and sends the transaction identifier[2] to the MSCT app.

**Step 4**

- The mobile device sends a transaction information request containing the transaction identifier to the MSCT service provider.
- The MSCT service provider reconciles this with the information received from the POI.
- The MSCT Inst application on the consumer’s mobile device pops-up a window with the transaction details including the merchant/trade name/IBAN_merch and transaction amount.
- The consumer authenticates and confirms the transaction by entering a mobile code on the mobile device.

[2] Note that in case different MSCT service providers for the consumer and merchant are involved, also the merchant service provider identifier needs to be transmitted (see Chapters 16 and 17 in the MSCT IG [10]).

**Step 5**

- Upon successful verification of the mobile code by the MSCT Inst application, an authentication code is calculated by the MSCT application, which is linked to the merchant and the transaction amount.

**Step 6**

The SCT Inst instruction, including the merchant / trade name, IBAN_merch, the transaction amount, the merchant transaction identifier and the authentication code are transmitted to the consumer's ASPSP via the MSCT service provider.

**Step 7**

- The consumer's ASPSP checks the integrity of the SCT Inst instruction and verifies the authentication code.
- The consumer's ASPSP checks the availability of funds on the payer's account.
- The consumer's ASPSP prepares and submits the SCT Inst transaction to the merchant's ASPSP.

**Step 8**

- A confirmation message is returned from the merchant's ASPSP to the consumer's ASPSP.
- The merchant's ASPSP makes the funds available to the merchant.

**Step 9**

- The merchant is notified by the MSCT service provider (information provided by the consumer's ASPSP) that their account has been credited.
- The consumer is notified by the MSCT service provider that the payment has been successfully executed (information provided by the consumer's ASPSP) and may optionally receive an e-receipt.

| Analysis MSCT Use case C2B-1 | |
|---|---|
| **Interoperability** | • The consumer and the merchant need to be subscribed to the same MSCT service<br>• The consumer's ASPSP and the merchant's ASPSP need be linked to the same MSCT service.<br>• For a truly "open" approach and a SEPA-wide interoperability, if the MSCT service provider of the payer is different to the MSCT service provider of the merchant, a framework needs to be specified that interconnects the different MSCT service providers. |

| | |
|---|---|
| **Challenges** | • Standardisation of merchant data transmitted from the merchant POI to the mobile device.<br>• Education of PSU on usage of NFC for MSCTs.<br>• The notification messages in step 9 are not included in the SCT Inst scheme. |

**Table 6: Analysis MSCT use case C2B-1**

*Notes:*

- The interoperability in case different MSCT service providers are involved for the consumer and the merchant is addressed in Chapters 16 and 17 in the MSCT IG [10]. In this case, a process flow similar as shown in Figure 48 in the MSCT IG applies.
- The minimum data elements in the notification messages are specified in Annex 5 of the MSC IG [10].

### 4.4 MSCT use case C2B-2: Mobile device - Offline use case – Payment at a physical POI using bi-directional NFC – single tap - MSCT application - SCA involving facial recognition

This MSCT use case presents an example for an in-store payment based on consumer-presented data and relying on an MSCT app on the mobile device of the consumer issued by their ASPSP (= payer's MSCT service provider). This MSCT use case involves a single tap by the consumer using NFC.

Benefitting from the bi-directional NFC communication capability, the consumer's mobile device and the POI exchange the data requested to build the payload, while performing SCA using the MSCT app prior to the tap. The result of this SCA mechanism is a cryptogram generated by the MSCT app which is transmitted by the POI, together with the other transaction data via the merchant's MSCT service provider to the consumer's ASPSP which will then verify this cryptogram. In the described example, the merchant's MSCT service provider acts as a PISP for the exchanges with the consumer's ASPSP.

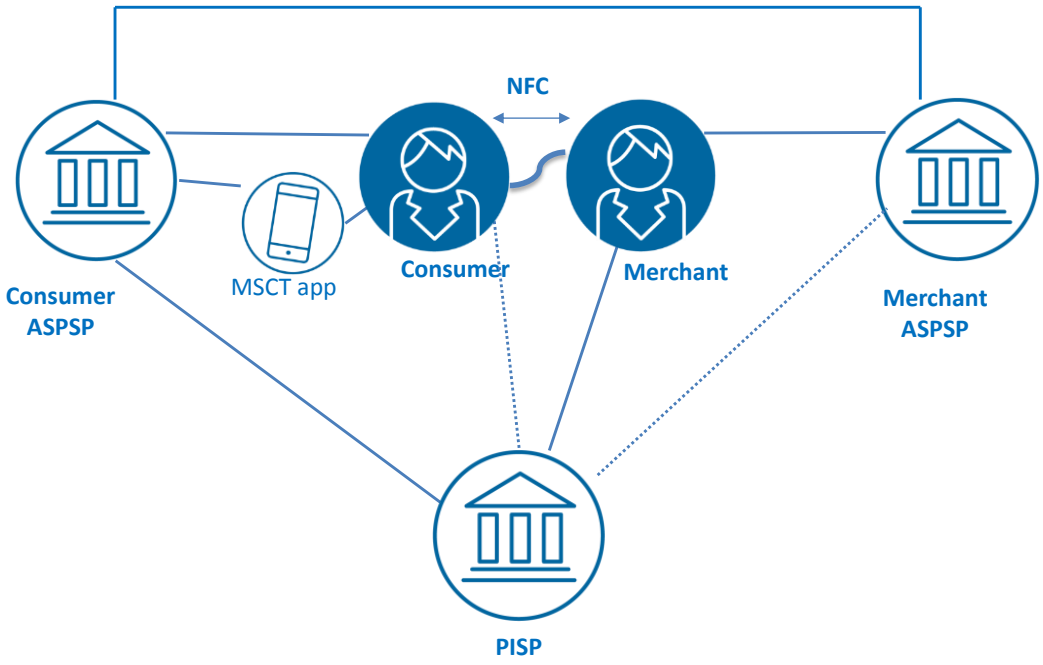



Figure 6: Actors in MSCT Use case C2B-2

Consumer and merchant, may, and frequently will, hold their payment accounts with different ASPSPs. It is assumed that the MSCT app is issued to the payer by their ASPSP while the merchant has a contract with a PISP (= merchant MSCT service provider) that supports the PSD2 API, has downloaded dedicated software on their POI and agreed to make the required PISP information available to the consumer according to the PSD2 Arts. 44 and 45[3].

[3] See also the EBA answer to Q&A 2020_5573.

In this payment transaction a strong customer authentication (see section 8.3 in the MSCT IG [10]) in accordance with the relevant PSD2 [2] requirements is performed involving a facial recognition (see section 8.2 in the MSCT IG [10]) and the calculation of a cryptogram by the MSCT application using a dedicated key, without dynamic linking to the payee and transaction amount[4].

No mobile network connectivity of the payer's mobile device is required in this use case, except for the notification to the consumer of the transaction execution (see Chapter 18 in the MSCT IG [10]).

[4] See EBA Q&A 2020_5247 that specifies that for this payment context no SCA with dynamic linking according to the PSD 2 and the RTS is required.

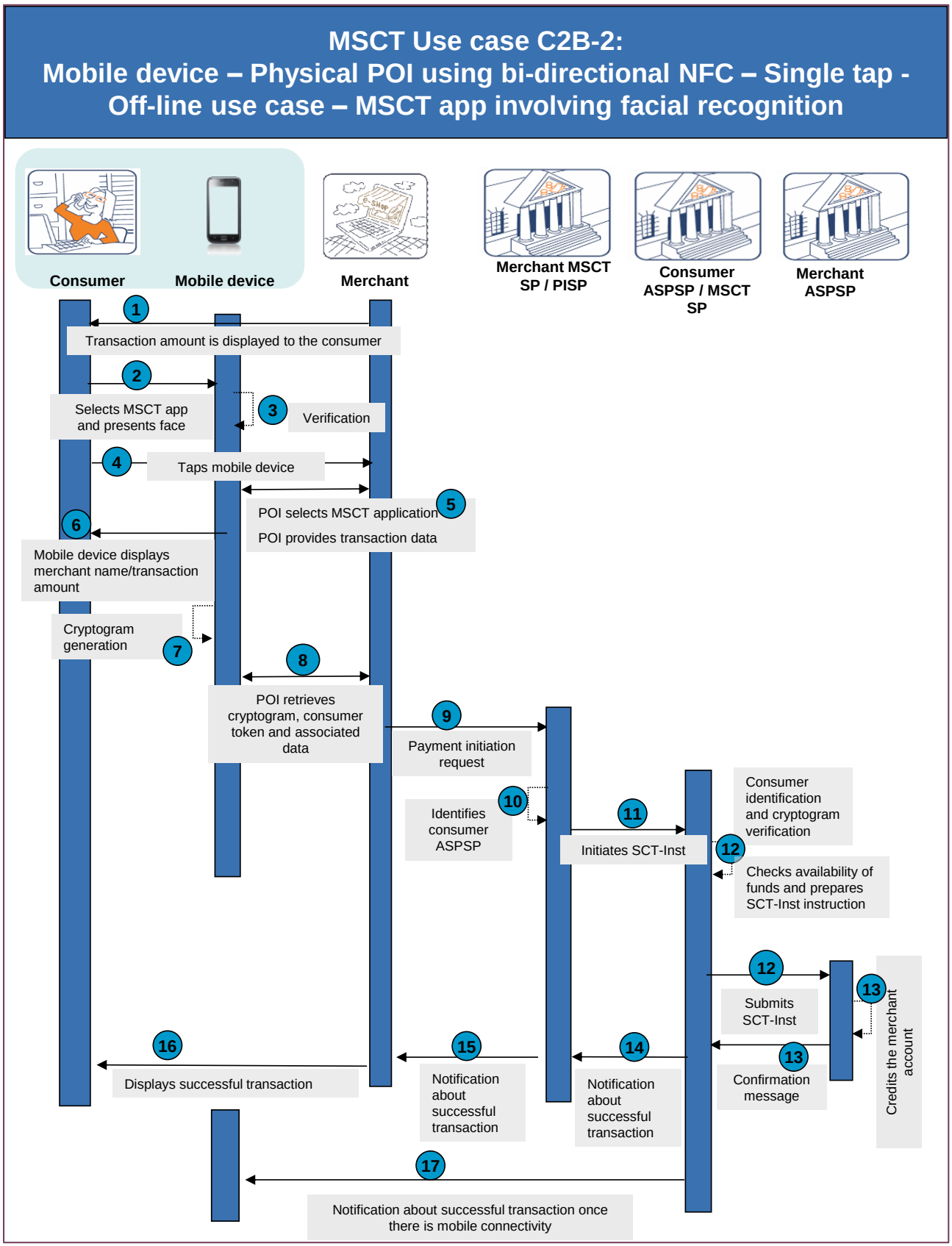



Figure 7: MSCT Use case C2B-2

In the figure above, the following steps are illustrated:

**Step 0**

- As a prerequisite, the consumer would need to be subscribed to the MSCT service and downloaded an MSCT application from their ASPSP on their mobile device.
- A token is used as surrogate value for the consumer identification data, generated by the consumer ASPSP.
- The merchant has contracted with a PISP and has installed their software on the POI.

**Step 1**

The merchant enters the transaction amount which is displayed on the POI.

**Step 2**

The consumer selects and opens the MSCT application on their mobile device and presents their face.

**Step 3**

The face is verified by the mobile device and the verification result is stored in a dedicated consumer verification parameter in the MSCT app.

**Step 4**

The consumer taps their mobile device on the POI. This gesture represents confirmation of the payment and the consumer's consent to the use of the PISP service[5].

**Step 5**

- While the mobile device is in the NFC field, the POI selects the MSCT application.
- While the mobile device is in the NFC field, the POI sends to the MSCT app the transaction amount, the merchant name/trade name, the transaction identifier and other transaction data.

**Step 6**

The merchant name/trade name and transaction amount are displayed to the consumer on the mobile device.

**Step 7**

While the mobile device is in the NFC field, the MSCT app generates a cryptogram using a dedicated key which is unlocked based on the positive facial verification (see dedicated consumer verification parameter in step 3). This cryptogram signs the transaction amount, name/trade name merchant, the transaction identifier, the consumer verification parameter and other data.

[5] In analogy to the EBA answer received on Q&A 2020_5570.

**Step 8**

While the mobile device is in the NFC field, the POI retrieves the cryptogram, the consumer token, the consumer's ASPSP identifier (= consumer's MSCT service provider ID) and other associated data from the MSCT application.

**Step 9**

The POI sends a payment initiation request to the merchant's MSCT service provider (=PISP). The payment initiation request message includes the transaction amount, merchant name/trade name and IBAN_merchant[6], transaction identifier, consumer token, consumer's ASPSP identifier, the cryptogram and other associated data.

**Step 10**

The merchant's MSCT service provider identifies the consumer's ASPSP from the consumer's ASPSP identifier.

**Step 11**

The merchant MSCT service provider, in its role of PISP, initiates a payment with the consumer ASPSP via the PSD2 API, and sends the full transaction data to the consumer's ASPSP, including the transaction amount, the merchant name/trade name and IBAN_merchant[7], transaction identifier, consumer token.

**Step 12**

- The consumer ASPSP, upon receipt of the payload, retrieves the consumer identification data from the token, checks the cryptogram using some of the associated data. They may also perform other optional controls (spending limits, risk management, …).
- The consumer's ASPSP checks the availability of funds on the payer's account.
- The consumer ASPSP prepares and submits the SCT Inst transaction to the merchant ASPSP.

**Step 13**

- A confirmation message is returned from the merchant's ASPSP to the consumer's ASPSP.
- The merchant's ASPSP makes the funds available to the merchant.

**Step 14**

The consumer's ASPSP sends a notification message to the PISP about the execution of the SCT Inst transaction.

[6] Alternatively, the name and IBAN of the merchant may also be added by the merchant MSCT service provider (=PISP).

[7] Alternatively, the name and IBAN of the merchant may also be added by the merchant MSCT service provider (=PISP).

**Step 15**

The PISP (= merchant MSCT service provider) sends a notification message to the merchant about the successful transaction.

**Step 16**

The merchant POI displays to the consumer that the transaction has been successfully executed.

**Step 17**

The consumer is informed by their ASPSP in the MSCT app about the successful transaction as soon as their mobile network connectivity.

| Analysis MSCT Use case C2B-2 | |
|---|---|
| **Interoperability** | • Based on and governed by PSD2.<br>• An MSCT app needs to be specified for the payer's mobile device as well as POI specifications supporting this transaction (a so-called dedicated MSCT kernel). |
| **Challenges** | • The PSD2 API needs to support the functionalities needed (e.g. cryptogram and other associated data, notification messages).<br>• Use of (bi-directional) NFC in certain phones is currently restricted.[8]<br>• Requires a contract between the merchant and the PISP (= merchant MSCT service provider).<br>• Co-existence with card-based payments using NFC on the POI.<br>• Information to the consumer and consumer consent with respect to usage of the PISP (PSD 2 Arts. 44, 45, 64, 66 and 94)[9] and RTS (Art. 30).<br>• Education of PSU on usage of NFC for MSCTs.<br>• The notification messages in steps 14 and 17 are not included in the SCT Inst scheme. |

Table 7: Analysis MSCT Use case C2B-2

*Notes:*

- The interoperability models for MSCTs involving a PISP are analysed in Chapter 20 of the MSCT IG [10].
- The co-existence of MSCT payments based on NFC, next to card-based payments based on NFC have been addressed in [13].

[8] This has been addressed by the recently published Digital Market Act [6] and will therefore be subject to change.

[9] See EBA answer to Q&A 2020_5573.

### 4.5 MSCT use case C2B-3: Mobile device - Offline use case – Payment at a physical POI using bi-directional NFC – double tap - MSCT application - SCA involving a mobile code

This MSCT use case presents an example for an in-store payment based on consumer-presented data and relying on an MSCT app on the mobile device of the consumer issued by their ASPSP (= payer's MSCT service provider). This MSCT use case involves a double tap by the consumer using NFC.

Benefitting from the bi-directional NFC communication capability, the consumer's mobile device and the POI exchange the data requested to build the payload, while performing SCA using the MSCT app between the two taps. The result of this SCA mechanism is a cryptogram generated by the MSCT app which is transmitted by the POI, together with the other transaction data via the merchant's MSCT service provider to the consumer's ASPSP which will then verify this cryptogram. In the described example, the merchant's MSCT service provider acts as a PISP for the exchanges with the consumer's ASPSP.

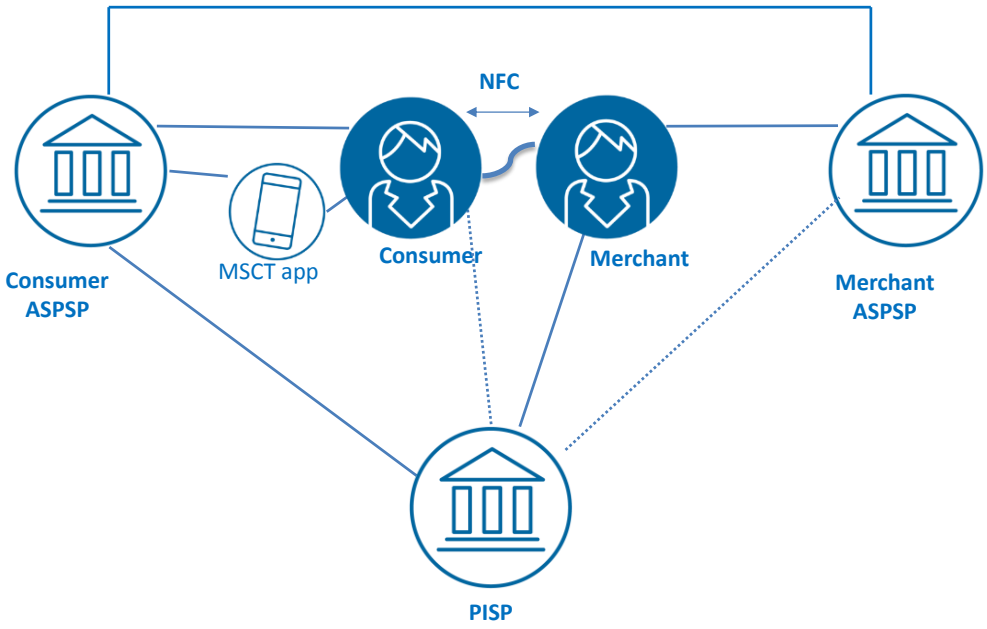



**Figure 8: Actors in MSCT Use case C2B-3**

Consumer and merchant, may, and frequently will, hold their payment accounts with different ASPSPs. It is assumed that the MSCT app is issued to the payer by their ASPSP while the merchant has a contract with a PISP (= merchant MSCT service provider) that supports the PSD2 API, has downloaded dedicated software on their POI and agreed to make the required PISP information available to the consumer according to the PSD2 Arts. 44 and 45[10].

In this payment transaction a strong customer authentication (see section 8.3 in the MSCT IG [10]) in accordance with the relevant PSD2 [2] requirements is performed involving a

[10] See also the EBA answer to Q&A 2020_5573.

mobile code (see section 8.2 in the MSCT IG [10]) and the calculation of a cryptogram by the MSCT application using a dedicated key, with dynamic linking to the payee and transaction amount.

No mobile network connectivity of the payer's mobile device is required in this use case, except for the notification to the consumer of the transaction execution (see Chapter 18 in the MSCT IG [10]).

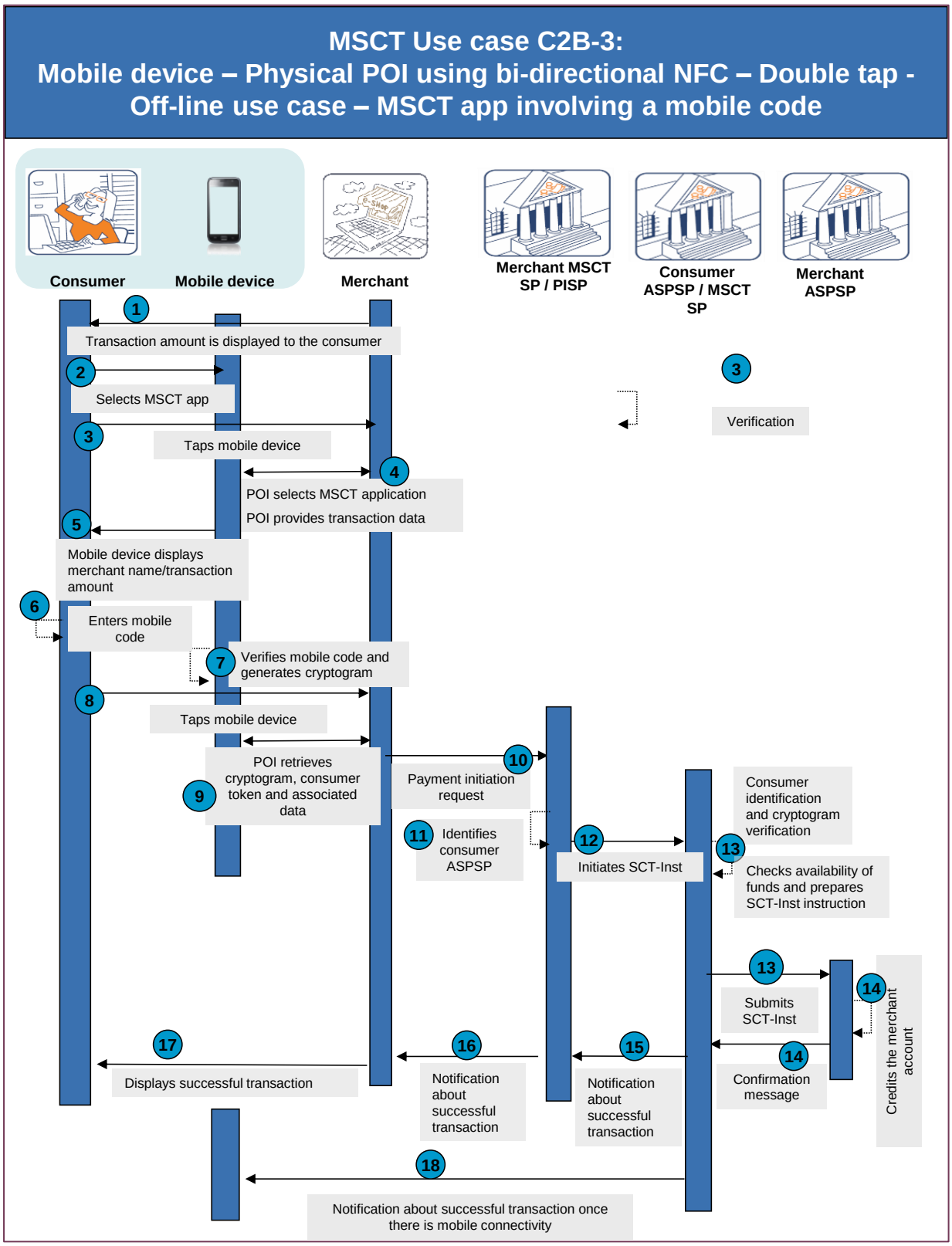



Figure 9: MSCT Use case C2B-3

In the figure above, the following steps are illustrated:

**Step 0**

- As a prerequisite, the consumer would need to be subscribed to the MSCT service and downloaded an MSCT application from their ASPSP on their mobile device.
- A token is used as surrogate value for the consumer identification data, generated by the consumer ASPSP.
- The merchant has contracted with a PISP and has installed their software on the POI.

**Step 1**

The merchant enters the transaction amount which is displayed on the POI.

**Step 2**

The consumer selects and opens the MSCT application on their mobile device.

**Step 3**

The consumer taps for the first time their mobile device on the POI.

**Step 4**

- While the mobile device is in the NFC field, the POI selects the MSCT application.
- While the mobile device is in the NFC field, the POI sends to the MSCT app the transaction amount, the merchant name/trade name, the transaction identifier and other transaction data.

**Step 5**

The merchant name/trade name and transaction amount are displayed to the consumer on the mobile device.

**Step 6**

The consumer authenticates and confirms the transaction by entering a mobile code on the mobile device.

**Step 7**

Upon successful verification of the mobile code by the MSCT Inst application, a cryptogram is calculated by the MSCT application using a dedicated key which is unlocked based on the positive mobile code verification. This cryptogram signs the transaction amount, name/trade name merchant, the transaction identifier, the consumer verification parameter and other data.

**Step 8**

The consumer taps for the second time their mobile device on the POI. This gesture represents confirmation of the payment and the consumer's consent to the use of the PISP service[11].

[11] In analogy to the EBA answer received on Q&A 2020_5570.

**Step 9**

While the mobile device is in the NFC field, the POI retrieves the cryptogram, the consumer token, the consumer's ASPSP identifier (= consumer's MSCT service provider ID) and other associated data from the MSCT application.

**Step 10**

The POI sends a payment initiation request to the merchant's MSCT service provider (=PISP). The payment initiation request message includes the transaction amount, merchant name/trade name and IBAN_merchant[12], transaction identifier, consumer token, consumer's ASPSP identifier, the cryptogram and other associated data.

**Step 11**

The merchant's MSCT service provider identifies the consumer's ASPSP from the consumer's ASPSP identifier.

**Step 12**

The merchant MSCT service provider, in its role of PISP, initiates a payment with the consumer ASPSP via the PSD2 API, and sends the full transaction data to the consumer's ASPSP, including the transaction amount, the merchant name/trade name and IBAN_merchant[13], transaction identifier, consumer token.

**Step 13**

- The consumer ASPSP, upon receipt of the payload, retrieves the consumer identification data from the token, checks the cryptogram using some of the associated data. They may also perform other optional controls (spending limits, risk management, …).
- The consumer's ASPSP checks the availability of funds on the payer's account.
- The consumer ASPSP prepares and submits the SCT Inst transaction to the merchant ASPSP.

**Step 14**

- A confirmation message is returned from the merchant's ASPSP to the consumer's ASPSP.
- The merchant's ASPSP makes the funds available to the merchant.

**Step 15**

The consumer's ASPSP sends a notification message to the PISP about the execution of the SCT Inst transaction.

[12] Alternatively, the name and IBAN of the merchant may also be added by the merchant MSCT service provider (=PISP).

[13] Alternatively, the name and IBAN of the merchant may also be added by the merchant MSCT service provider (=PISP).

**Step 16**

The PISP (= merchant MSCT service provider) sends a notification message to the merchant about the successful transaction.

**Step 17**

The merchant POI displays to the consumer that the transaction has been successfully executed.

**Step 18**

The consumer is informed by their ASPSP in the MSCT app about the successful transaction as soon as their mobile network connectivity is restored.

| Analysis MSCT Use case C2B-3 | |
|---|---|
| **Interoperability** | • Based on and governed by PSD2.<br>• An MSCT app needs to be specified for the payer's mobile device as well as POI specifications supporting this transaction (a so-called dedicated MSCT kernel). |
| **Challenges** | • The PSD2 API needs to support the functionalities needed (e.g. cryptogram and other associated data, notification messages).<br>• Use of NFC in certain phones is currently restricted.[14]<br>• Requires a contract between the merchant and the PISP (= merchant MSCT service provider).<br>• Co-existence with card-based payments using NFC on the POI.<br>• Information to the consumer and consumer consent with respect to usage of the PISP (PSD 2 Arts. 44, 45, 64, 66 and 94)[15] and RTS (Art. 30).<br>• Education of PSU on usage of NFC for MSCTs.<br>• The notification messages in steps 15 and 18 are not included in the SCT Inst scheme.<br>• Consumer experience in view of the double tap. |

Table 8: Analysis MSCT Use case C2B-3

*Notes:*

- The interoperability models for MSCTs involving a PISP are analysed in Chapter 20 of the MSCT IG [10].
- The co-existence of MSCT payments based on NFC, next to card-based payments based on NFC have been addressed in [13]**Error! Reference source not found.**.

[14]This has been addressed by the recently published Digital Market Act [6] and will therefore be subject to change.

[15] See EBA answer to Q&A 2020_5573.

### 4.6 MSCT use case C2B-4: Mobile device - Offline use case – Payment at a physical POI using bi-directional NFC and EMV-based SCA involving a fingerprint

This MSCT use case presents an example for an in-store payment based on consumer-presented data and relying on EMV technology for the authentication of the consumer by their ASPSP. The MSCT use case involves a single tap by the consumer using NFC.

Benefitting from the bi-directional NFC communication capability, the consumer's mobile device and the POI exchange the data requested to build the payload, while performing SCA using a mobile EMV contactless authentication app issued by the consumer's ASPSP. The result of this SCA mechanism is a cryptogram generated by the EMV app which is transmitted by the POI, together with the other transaction data via the merchant's MSCT service provider to the consumer's ASPSP which will then verify this cryptogram. In the described example, the merchant's MSCT service provider acts as a PISP for the exchanges with the consumer's ASPSP.

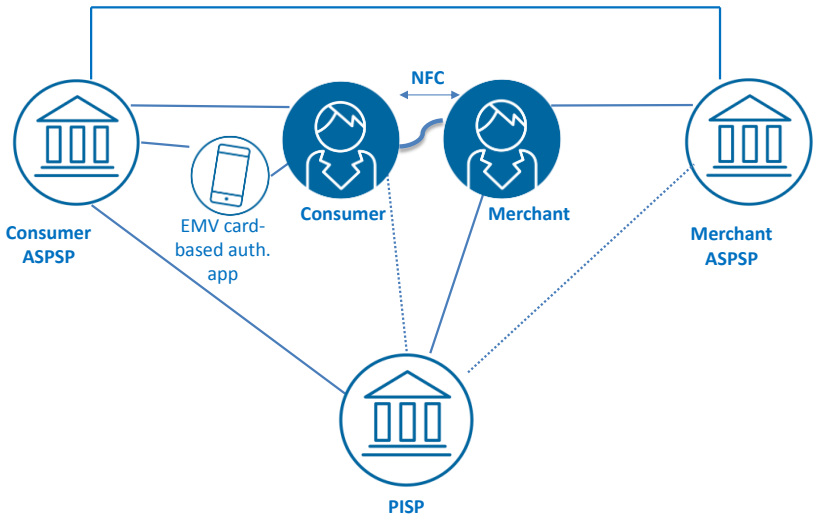



**Figure 10: Actors in MSCT Use case C2B-4**

Consumer and merchant, may, and frequently will, hold their payment accounts with different ASPSPs. The merchant has a contract with a PISP (= merchant MSCT service provider) that supports the PSD2 API, has downloaded dedicated software on their POI and agreed to make the required PISP information available to the consumer according to the PSD2 Arts. 44 and 45[16].

In this payment transaction a strong customer authentication (see section 8.3 in the MSC IG [10]) in accordance with the relevant PSD2 [2] requirements is performed involving a

[16] See also the EBA answer to Q&A 2020_5573.

fingerprint (see section 8.2 in the MSCT IG [10]) and the calculation of a cryptogram by the EMV application using a dedicated key, without dynamic linking of the payee[17].
No mobile network connectivity of the mobile device is required in this use case, except for the possible notification to the consumer of the transaction execution (see Chapter 18 in the MSCT IG [10]).

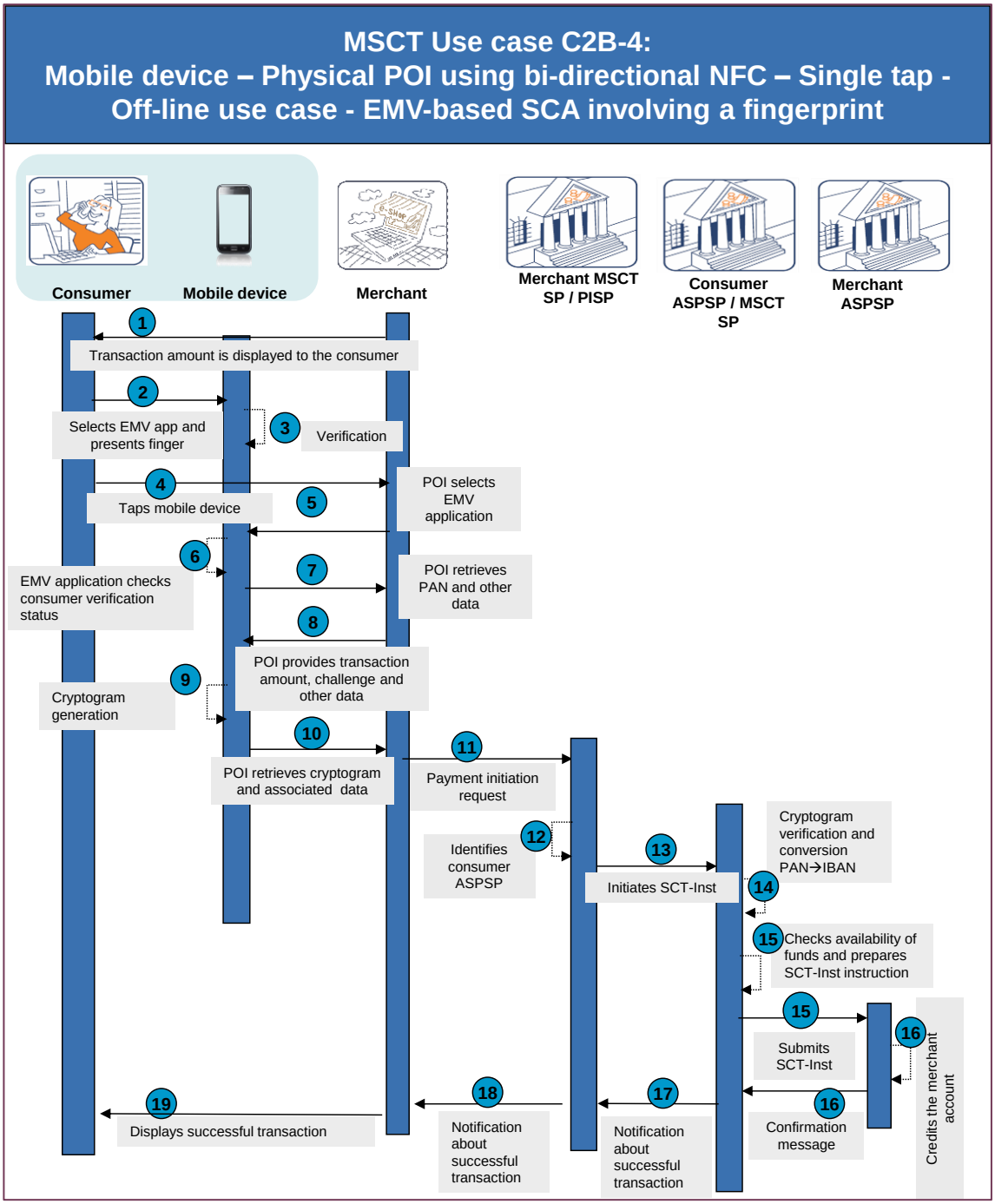



Figure 11: MSCT Use case C2B-4

[17] See EBA Q&A 2020_5247 that specifies that for this payment context no SCA with dynamic linking according to the PSD 2 and the RTS is required.

In the figure above, the following steps are illustrated:

**Step 0**

- As a prerequisite, the consumer has downloaded a mobile EMV contactless authentication app from their ASPSP on their mobile device. They have further registered their IBAN to be converted into a consumer_PAN[18] and this consumer_PAN has been provisioned to the authentication app.
- The merchant has contracted with the MSCT service provider (=PISP) and installed their software on the POI.

**Step 1**

The merchant enters the transaction amount which is displayed on the POI.

**Step 2**

The consumer selects and opens the EMV authentication application on their mobile device and presents a fingerprint.[19]

**Step 3**

The fingerprint is verified by the mobile device and the verification result is stored in the mobile device.

**Step 4**

The consumer taps their mobile device on the POI. This gesture may represent the consumer's consent to the use of the PISP service[20].

**Step 5**

While the mobile device is in the NFC field, the POI selects the EMV authentication application.

**Step 6**

While the mobile device is in the NFC field, the EMV application checks the status of the consumer verification that was stored on the mobile device and stores this result in an EMV parameter (Card Verification Result).

**Step 7**

While the mobile device is in the NFC field, the POI retrieves from the EMV authentication app the PAN and possibly other data.

**Step 8**

While the mobile device is in the NFC field, the POI sends to the EMV application the transaction amount, a challenge and other transaction data such as date, country code, etc.

[18] This consumer_PAN is to be considered as a token for the IBAN of the consumer.

[19] Other consumer verification methods may be applied, see section 8 in the MSCT IG [10].

[20] In analogy to the EBA answer received on Q&A 2020_5570.

**Step 9**

While the mobile device is in the NFC field, the EMV application generates a cryptogram. This cryptogram signs the transaction amount, challenge, the Card Verification Result and other data.

**Step 10**

While the mobile device is in the NFC field, the POI retrieves the cryptogram and other associated data from the EMV application.

**Step 11**

The POI sends a payment initiation request to the merchant's MSCT service provider (=PISP). The payment initiation request message includes the transaction amount, name and IBAN_merchant[21], transaction identifier, consumer_PAN, the cryptogram and other associated data.

**Step 12**

The merchant's MSCT service provider identifies the consumer's ASPSP from the Issuer Identification Number present in the consumer_PAN (typically the first 6 digits).

**Step 13**

The merchant MSCT service provider, in its role of PISP, initiates a payment with the consumer ASPSP via the PSD2 API, and sends the full transaction data to the consumer's ASPSP, including the transaction amount, consumer_PAN, merchant name, merchant_IBAN, transaction identifier, cryptogram.

**Step 14**

- The consumer ASPSP, upon receipt of the payload, checks the cryptogram using some of the associated data. They may also perform other optional controls (spending limits, risk management, …).
- Subsequently to the successful verification of the cryptogram, the consumer ASPSP converts the consumer_PAN back to the IBAN of the consumer.

**Step 15**

- The consumer's ASPSP checks the availability of funds on the payer's account.
- The consumer ASPSP prepares and submits the SCT Inst transaction to the merchant ASPSP.

**Step 16**

- A confirmation message is returned from the merchant's ASPSP to the consumer's ASPSP.
- The merchant's ASPSP makes the funds available to the merchant.

[21] Alternatively, the name and IBAN of the merchant may also be added by the merchant MSCT service provider.

**Step 17**

The consumer's ASPSP sends a notification message to the PISP about the execution of the SCT Inst transaction.

**Step 18**

The PISP (= merchant MSCT service provider) sends a notification message to the merchant about the successful transaction.

**Step 19**

The merchant POI displays to the consumer that the transaction has been successfully executed.

| Analysis MSCT Use case C2B-4 | |
|---|---|
| **Interoperability** | • Based on and governed by PSD2.<br>• An EMV contactless kernel should be selected[22] to serve as a basis for interoperability for the communication with the consumer's mobile device. |
| **Challenges** | • The PSD2 API needs to support the functionalities needed (e.g. PAN, cryptogram and other associated data, notification message).<br>• Use of NFC in certain phones is currently restricted.[23]<br>• Requires a contract between the merchant and the PISP (= merchant MSCT service provider).<br>• Information to the consumer and consumer consent with respect to usage of the PISP (PSD 2 Arts. 44, 45, 64, 66 and 94) [24] and RTS (Art. 30).<br>• Impact of using the EMVCo specifications for an authentication app for MSCTs.<br>• Education of PSU on usage of NFC for MSCTs. |

Table 9: Analysis MSCT Use case C2B-4

*Note:* The interoperability models for MSCTs involving a PISP are analysed in Chapter 20 of the MSCT IG [10].

[22] This would need to be covered by a to be defined "Interoperability Framework for MSCTs"

[23] This has been addressed by the recently published Digital Market Act [6] and will therefore be subject to change.

[24] See EBA answer to Q&A 2020_5573.

# 5 MSCTs use cases based on BLE

## 5.1 Introduction

This chapter is devoted to MSCTs whereby BLE as a proximity technology for the data exchange between the payer and the payee is used to enable the initiation of an MSCT, as defined in the MSCT IG [10].

In this chapter a number of MSCT use cases will be described with a diagram depicting the different actors involved and with a decomposition into the different steps of the MSCT transaction which are also shown in a figure. Each MSCT use case is followed by a short evaluation on the interoperability aspects for deployment across SEPA and compliance with respect to the PSD2 [2] and the RTS [3], including a short list of the main challenges.

For the usage of BLE as proximity technology, a pairing between the payer device and the payee infrastructure (e.g. POI, mobile device, beacon) is necessary before data may be exchanged in a bi-directional way. The examples described in this chapter make use of a payee (merchant)-presented QR-code[25] to provide the necessary information to the payer's mobile device to establish this pairing. Note that this information exchange may also be implemented using uni-directional NFC.

In view of the distances between the payer's mobile device and the payee's mobile device or merchant's POI device enabled by BLE, end-to-end encryption between the two devices would be needed from a security perspective. The examples below make use of symmetric encryption using a secret session key derived from temporary Elliptic Curve Diffie-Hellman (ECDH) key pairs generated by the respective devices for each payment transaction.

Note that several MSCT use cases in this chapter rely on the eIDAS2.0 (see Annex 1) that is still under development.[26]

Note that these MSCT use cases are presented for illustrative purposes, in other words, the list of MSCT use cases described is not meant to be exhaustive but should be seen as examples for specific payment contexts. Likewise, the authentication method used is purely illustrative. More details on payer identification and SCA are provided in the sections 8.2 and 8.3 in the MSCT IG [10]. Note that all these use cases accommodate an offline payment context whereby the payer's mobile device has no mobile network connectivity.

[25]Note that this QR-code is only used to establish a secure connection between the two devices and has a different format than the one standardised in [14].

[26] See also the EU Retail Payments Strategy including the use of EUid, eIDAS signatures and e-receipts, https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52020DC0592&from=EN.

| Payment context | # | MSCT use case description |
|---|---|---|
| **Person-to-Person (P2P) payments** | P2P-2 | Mobile device – Offline use case – Person-to-Person payment with payee-presented QR-code involving a PISP – SCA via BLE using EUDIW involving a fingerprint |
| **Consumer-to-Business (C2B) payments** | C2B-5 | Mobile device - Offline use case - Payment at a physical POI with consumer-presented QR-code involving a PISP – SCA via BLE using an MSCT app involving a fingerprint |
| | C2B-6 | Mobile device – Offline use case - Payment at a physical POI with merchant-presented QR-code involving a PISP – SCA via BLE using an MSCT app involving a mobile code |
| | C2B-7 | Mobile device – Offline use case - Payment at a physical POI with merchant-presented QR-code involving a PISP – SCA via BLE using an EUDIW involving a fingerprint |

Table 10: Overview illustrative MSCT use cases using BLE

## 5.2 MSCT use case P2P-2: Mobile device – Offline use case – Person-to-Person payment with payee-presented QR-code involving a PISP – SCA via BLE using EUDIW involving a fingerprint

This use case presents an example of payer experience whereby their mobile device has no mobile network connection[27] and is used for a payment to a payee. In this use case a combination of two proximity technologies is used: a payee-presented QR-code and BLE.

Both the payer and the payee have preloaded a European Digital Identity Wallet (EUDIW) onto their mobile device and have been provisioned identity and (core identity) attributes certificates[28] from a Person Identification Data (PID) provider that operates under the eIDAS framework[29]. Moreover the respective EUDIWs have also been provisioned with the respective account holder name[30] and IBAN certificates by the respective ASPSPs.

Both wallets support the generation of a Qualified Electronic Signature (QES)[31] and Advanced Electronic Signature (AdES) based on respective asymmetric key pairs as specified by eIDAS [5]. It is further assumed that the QR-code[32] provided by the payee's EUDIW contains the necessary information to enable the establishment of a secure connection (see Chapter 9 MSCT IG [10]) between the payee's EUDIW and the payer's EUDIW via BLE for performing the SCA.

The secure connection between the respective wallets is based on symmetric encryption using a secret session key derived from dedicated session ECDH key pairs that are generated for each transaction by the respective wallets (see

**Figure 14** for an overview of the cryptography).

The payee is registered with a PISP (= payee MSCT service provider) that supports the PSD2 API, has downloaded dedicated software on their mobile device and agreed to make the required PISP information available to the payer according to the PSD2 Arts. 44 and 45[33].

[28] Core attribute certificate may be for example age attestation, an attribute certificate may be for example an identity's address.

[29] See https://ec.europa.eu/futurium/en/content/eidas-regulation-regulation-eu-ndeg9102014

[30] Alternatively, an ASPSP may also make use of a person name attribute certificate issued by a PID provider.

[31] This means that the mobile devices and the respective wallets need to be certified as QSCD (Qualified Signature Creation Device). The fall-back would be to use an AdES (Advance Electronic Signature) supported by a Qualified Certificate.

[32] Note that this QR-code is only used to establish a secure connection between the two devices and has a different format than the one standardised in [14]

[33] See also the EBA answer to Q&A 2020_5573.

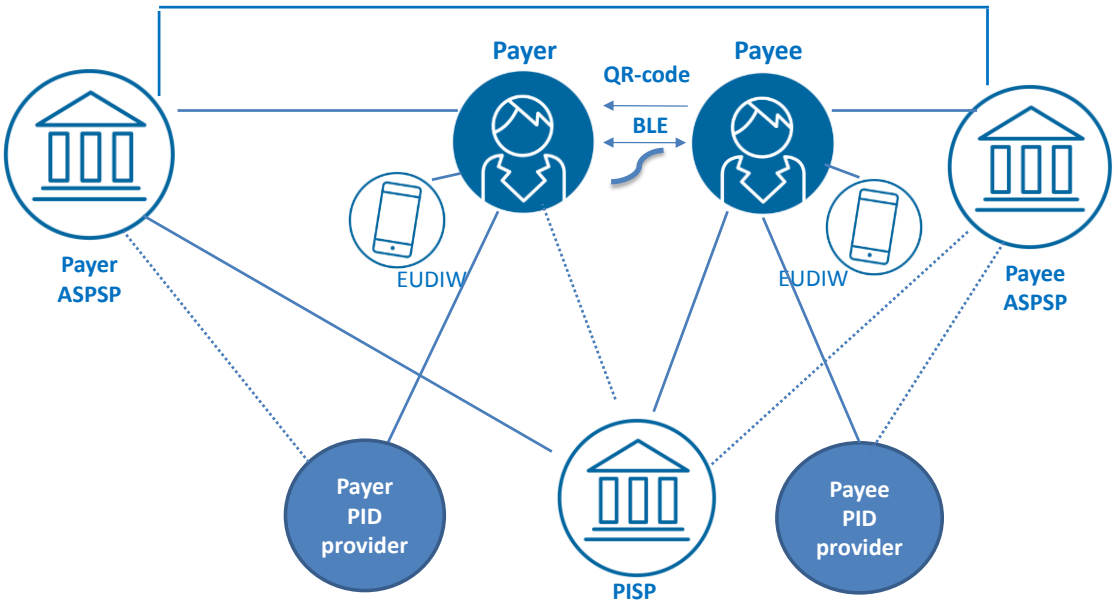


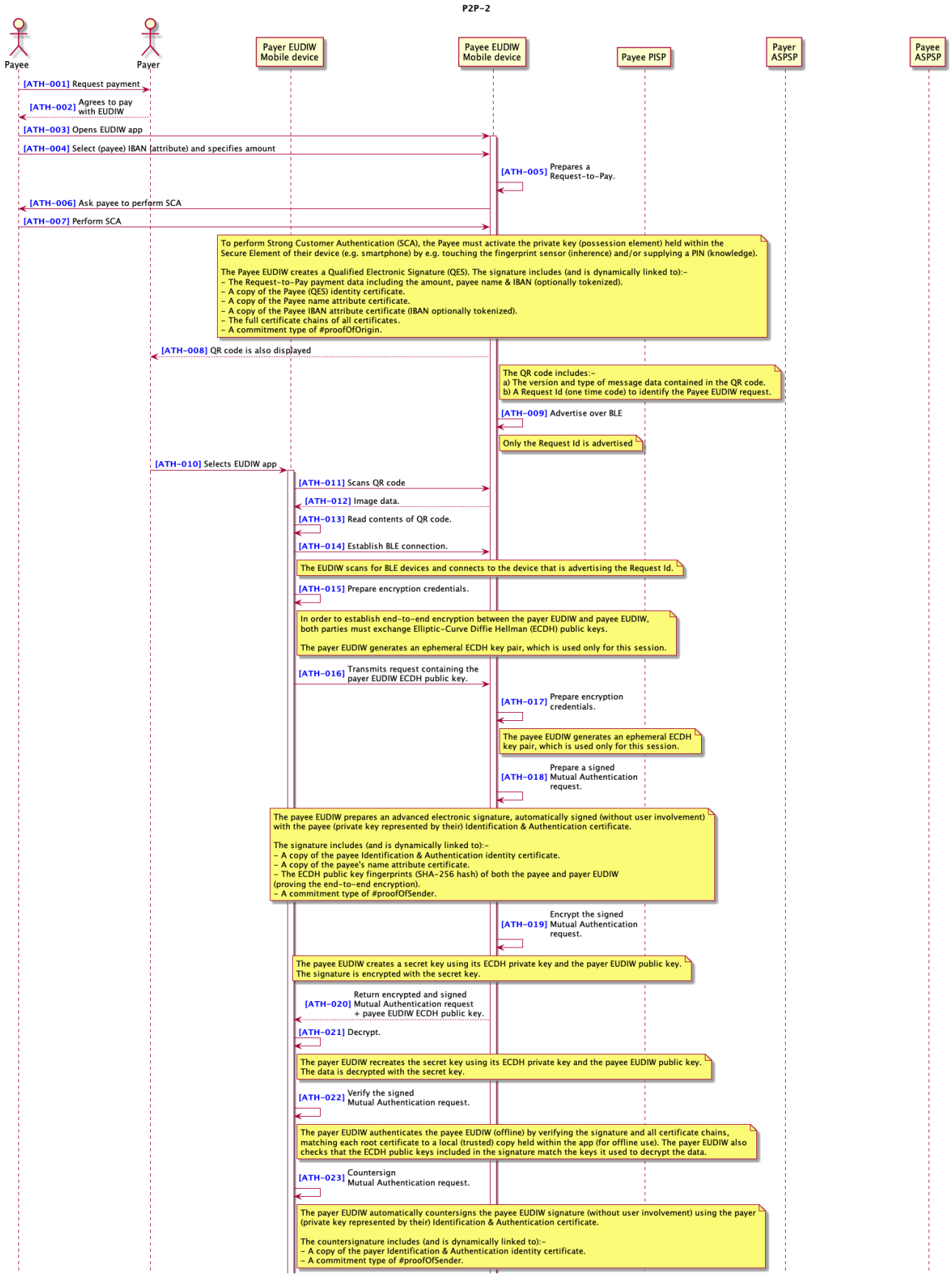

Figure 12: Actors in MSCT Use case P2P-2

Payer and payee, may, and frequently will, hold their payment accounts with different ASPSPs. Both ASPSPs rely on the usage of the EUDIW and the eIDAS2.0 for the PSU authentication.

In this payment transaction a strong customer authentication (see section 8.3 in the MSCT IG [10]) in accordance with the relevant PSD2 [2] requirements is performed involving a fingerprint (see section 8.2 in the MSCT IG [10]) and the calculation of a QES by the payer's EUDIW using a private key.

No mobile network connectivity of the payer's mobile device is required in this use case.

Figure 13: MSCT Use case P2P-2 (to be developed for final version)


P2P-2
Payee
Payer
Payer EUDIW Mobile device
Payee EUDIW Mobile device
Payee PISP
Payer ASPSP
Payee ASPSP
[ATH-001] Request payment
[ATH-002] Agrees to pay with EUDIW
[ATH-003] Opens EUDIW app
[ATH-004] Select (payee) IBAN (attribute) and specifies amount
[ATH-005] Prepares a Request-to-Pay.
[ATH-006] Ask payee to perform SCA
[ATH-007] Perform SCA
To perform Strong Customer Authentication (SCA), the Payee must activate the private key (possession element) held within the Secure Element of their device (e.g. smartphone) by e.g. touching the fingerprint sensor (inherence) and/or supplying a PIN (knowledge).
The Payee EUDIW creates a Qualified Electronic Signature (QES). The signature includes (and is dynamically linked to):-
- The Request-to-Pay payment data including the amount, payee name & IBAN (optionally tokenized).
- A copy of the Payee (QES) identity certificate.
- A copy of the Payee name attribute certificate.
- A copy of the Payee IBAN attribute certificate (IBAN optionally tokenized).
- The full certificate chains of all certificates.
- A commitment type of #proofOfOrigin.
[ATH-008] QR code is also displayed
The QR code includes:-
a) The version and type of message data contained in the QR code.
b) A Request Id (one time code) to identify the Payee EUDIW request.
[ATH-009] Advertise over BLE
Only the Request Id is advertised
[ATH-010] Selects EUDIW app
[ATH-011] Scans QR code
[ATH-012] Image data.
[ATH-013] Read contents of QR code.
[ATH-014] Establish BLE connection.
The EUDIW scans for BLE devices and connects to the device that is advertising the Request Id.
[ATH-015] Prepare encryption credentials.
In order to establish end-to-end encryption between the payer EUDIW and payee EUDIW, both parties must exchange Elliptic-Curve Diffie Hellman (ECDH) public keys.
The payer EUDIW generates an ephemeral ECDH key pair, which is used only for this session.
[ATH-016] Transmits request containing the payer EUDIW ECDH public key.
[ATH-017] Prepare encryption credentials.
The payee EUDIW generates an ephemeral ECDH key pair, which is used only for this session.
[ATH-018] Prepare a signed Mutual Authentication request.
The payee EUDIW prepares an advanced electronic signature, automatically signed (without user involvement) with the payee (private key represented by their) Identification & Authentication certificate.
The signature includes (and is dynamically linked to):-
- A copy of the payee Identification & Authentication identity certificate.
- A copy of the payee's name attribute certificate.
- The ECDH public key fingerprints (SHA-256 hash) of both the payee and payer EUDIW (proving the end-to-end encryption).
- A commitment type of #proofOfSender.
[ATH-019] Encrypt the signed Mutual Authentication request.
The payee EUDIW creates a secret key using its ECDH private key and the payer EUDIW public key. The signature is encrypted with the secret key.
[ATH-020] Return encrypted and signed Mutual Authentication request + payee EUDIW ECDH public key.
[ATH-021] Decrypt.
The payer EUDIW recreates the secret key using its ECDH private key and the payee EUDIW public key. The data is decrypted with the secret key.
[ATH-022] Verify the signed Mutual Authentication request.
The payer EUDIW authenticates the payee EUDIW (offline) by verifying the signature and all certificate chains, matching each root certificate to a local (trusted) copy held within the app (for offline use). The payer EUDIW also checks that the ECDH public keys included in the signature match the keys it used to decrypt the data.
[ATH-023] Countersign Mutual Authentication request.
The payer EUDIW automatically countersigns the payee EUDIW signature (without user involvement) using the payer (private key represented by their) Identification & Authentication certificate.
The countersignature includes (and is dynamically linked to):-
- A copy of the payer Identification & Authentication identity certificate.
- A commitment type of #proofOfSender.

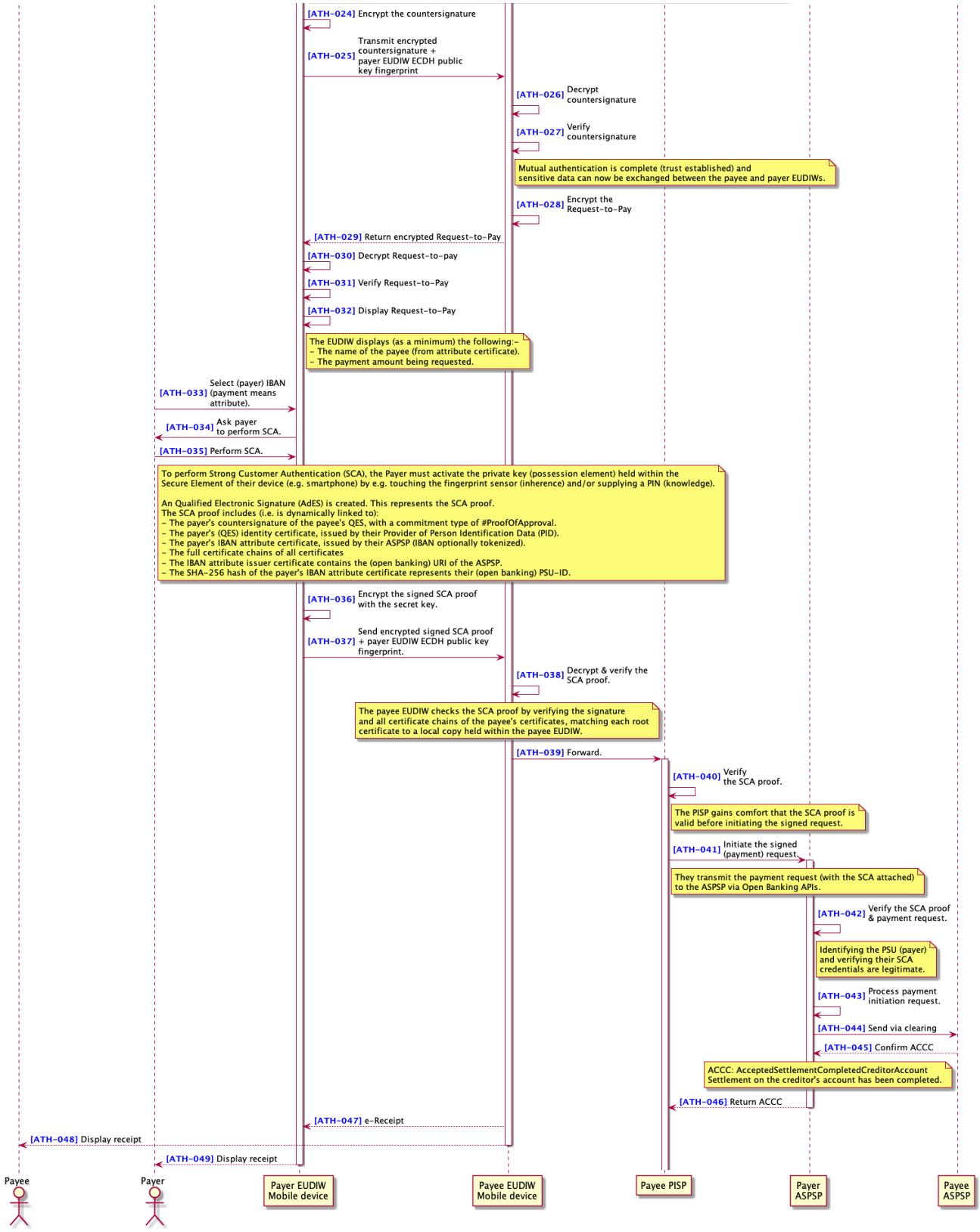



Figure 14: MSCT Use case P2P-2 – overview cryptography

In the figure above, the following steps are illustrated:

**Step 0**

- As a prerequisite, both payer and payee would need to have downloaded an EUDIW and be provisioned with identity and (core identity) attributes certificates from a PID provider and also with account name and IBAN attribute certificates from their respective ASPSPs. It is assumed that the EUDIW supports the generation of AdES and QES, including the cryptographic keys needed, under the eIDAS framework and are compliant with an interoperability standard allowing two EUDIWs to communicate with each other over a BLE proximity connection. It is also assumed that this connection will be encrypted with ECDH session keys. Moreover, the EUDIWs store all eIDAS trusted root certificates[34] and ASPSP trusted root certificates to enable all public key and attribute certificate verifications offline.
- The use case also assumes that the EUDIW will allow the payee to use a PISP (or their ASPSP acting as a PISP) with whom they have been pre-registered, and that the EUDIW is able to securely communicate with this PISP.
- During the payment transaction, there is no mobile communication network required for the consumer's mobile device.

**Step 1**

A payee requests a payer to pay them an amount of money and they agree to use both their EUDIW for this payment.

**Step 2**

- The payee opens their EUDIW on their mobile device, which possibly involves the entry of a password or a biometric verification.
- The payee selects an IBAN (attribute certificate) from their EUDIW and enters the transaction amount that they wish to be paid.

**Step 3**

The payee's EUDIW prepares a Payment Request message. This message contains, as a minimum, the transaction amount, the payee name attribute certificate and a reference to the payee IBAN attribute certificate[35].

**Step 4[36]**

The payee's EUDIW requests the payee to perform SCA.

**Step 5**

- The payee presents a finger to their mobile device[37].

[34] Based on an eIDAS Trusted List of root certificates

[35] For example X.509 attribute certificates.

[36] Note that the SCA of the payee is not required by PSD2.

[37] Other identification methods may be used such as the presentation of a mobile code.

- Upon successful verification of the fingerprint, the payee EUDIW creates a QES using a private key, stored in the EUDIW. The signature includes (and is dynamically linked to):
    - The data of the Payment Request message including the transaction amount, the payee name and the payee IBAN;
    - A copy of the payee (QES) identity certificate;
    - A copy of the payee name attribute certificate;
    - A copy of the payee IBAN attribute certificate;
    - The full certificate chains of all certificates (corresponding to the identity and all attribute certificates);
    - A commitment type of #ProofOfOrigin.

**Step 6**

The payee's EUDIW calculates and displays a QR-code which contains:

- The version and type of message data contained in the QR-code;
- A RequestID to identify the payee EUDIW Payment Request to enable the establishment of a BLE connection (see Step 3).

**Step 7**

The payee EUDIW advertises this RequestID over BLE.

**Step 8**

- The payer opens their EUDIW and scans the QR-code displayed on the payee's mobile device. This gesture may represent the payer's consent to the use of the PISP service[38].
- The payer EUDIW reads the QR-code data and retrieves the RequestID.

**Step 9**

- The payer EUDIW scans for a BLE advertisement containing the RequestID.
- The payer EUDIW then connects to the payee's EUIDW over BLE.

**Step 10**

- The payer EUDIW generates a temporary ECDH key pair, which is used only for this session[39].
- The payer EUDIW transmits the ECDH public key to the payee EUDIW over the BLE connection.

**Step 11**

The payee EUDIW also generates a temporary ECDH key pair, which is used only for this session.

[38] In analogy with the EBA answer received on Q&A 2020_5570.

[39] In order to establish end-to-end encryption between the payer and the payee EUDIWs, both exchange Elliptic Curve Diffie-Hellman (ECDH) public keys.

**Step 12**

- The payee EUDIW prepares a Mutual Authentication request.
- The payee EUDIW automatically signs (without user involvement) the Mutual Authentication request by calculating an AdES, using the payee private key, corresponding to their Identification & Authentication identity certificate.
- The signature includes (and is dynamically linked to):
    - A copy of the payee Identification & Authentication certificate.
    - A copy of the payee's name attribute certificate;
    - The SHA-256 hash of the ECDH public keys of both the payee and the payer EUDIW (providing proof of the end-to-end encrypted session);
    - A commitment type of #ProofOfSender.

**Step 13**

The payee EUDIW encrypts the signed Mutual Authentication request using a secret session key generated from the payee ECDH private key and the payer ECDH public key. The signed Mutual Authentication request is encrypted with the secret session key.

**Step 14**

The payee EUDIW transmits the encrypted and signed Mutual Authentication request together with a copy of the payee ECDH public key to the payer EUDIW.

**Step 15**

On receipt, the payer EUDIW decrypts the signed Mutual Authentication request using the secret session key generated from the payer ECDH private key and the payee ECDH public key.

**Step 16**

The payer EUDIW authenticates the payee EUDIW (offline) by verifying the signature and all certificate chains, matching each root certificate to a local (trusted) copy held within the EUDIW for offline use. The payer EUDIW also checks that the ECDH public keys included in the signature match the keys it used to decrypt the signed Mutual Authentication request.

**Step 17**

- The payer EUDIW automatically countersigns the payee’s Mutual Authentication request signature (without user involvement) by calculating an AdES using the payer private key corresponding to their Identification & Authentication identity certificate.
- The countersignature includes (and is dynamically linked to):
    - A copy of the payer Identification & Authentication identity certificate.
    - A commitment type of #ProofOfSender.

**Step 18**

The payer EUDIW encrypts the countersignature with the secret session key generated in step 10 – ECDH secret key.

**Step 19**

The payer EUDIW transmits the encrypted countersignature together with a SHA-256 hash of the payer ECDH public key to the payee EUDIW.

**Step 20**

The payee EUDIW decrypts the countersignature by using the ECDH secret key.

**Step 21**

The payee EUDIW verifies the countersignature. The payee EUDIW authenticates the payer EUDIW (offline) by verifying the signature and all certificate chains, matching each root certificate to a local (trusted) copy held within the EUDIW for offline use.[40]

**Step 22**

The payee EUDIW encrypts the signed Payment Request message (from Step 5) with the ECDH secret session key.

**Step 23**

The payee EUDIW transmits the encrypted and signed Payment Request message to the payer EUDIW, including the QES.

**Step 24**

The payer EUDIW decrypts the signed Payment Request message using the ECDH secret session key.

**Step 25**

The payer EUDIW verifies the payee QES signature on the Payment Request message, by verifying the certificate chain.

**Step 26**

The payer EUDIW retrieves from the Payment Request message the name of the payee (from the corresponding attribute certificate) and the transaction amount, which are displayed to the payer.

**Step 27**

The payer selects an IBAN (attribute certificate) to pay with. Once the IBAN is selected the payer's EUDIW creates a new Payment Request message by combining the content of the payee's Payment Request message with the payer data including (as a minimum) a reference (e.g. SHA-1 hash) to the payer IBAN attribute certificate[41].

**Step 28**

The EUDIW requests a payer confirmation of the payment by performing SCA.

[40] Mutual authentication is now complete (trust has been established) and sensitive data can now be exchanged between the payee and the payer EUDIWs.
[41] For example X.509 attribute certificate.

**Step 29**

- The payer presents their finger to the mobile phone.
- Upon successful verification of the fingerprint, a QES is created using the private key stored in the EUDIW of the payer on the following data;
    - The payer's countersignature of the payee's QES, with a commitment type of #ProofOfApproval.
    - The payer's (QES) Strong User Authentication Certificate, issued by their Identity Service Provider.
    - The payer's IBAN attribute certificate, issued by their ASPSP, containing the payer's ASPSP Host ID (URI);
    - The full certificate chains of all certificates;
    - The SHA-256 hash of the payer's IBAN attribute certificate which could represent their CustomerID[42].
    - The new Payment Request Message, generated in step 27.

**Step 30**

The payer EUDIW encrypts the signed Payment Request Message with the ECDH secret session key.

**Step 31**

The payer EUDIW transmits the encrypted signed SCA proof together with a copy of the payer ECDH public key SHA-256 hash.

**Step 32**

- The payee EUDIW decrypts the encrypted signed Payment Request message by using the ECDH secret session key;
- The payee EUDIW verifies the payer's QES signature and the full certificate chains of the payer's certificates, matching each root certificate to a local copy held within the payer EUDIW.

**Step 33**

The payee EUDIW forwards the signed Payment request message to their PISP.

**Step 34**

- Next the PISP optionally verifies the payer's and payee's QES on the Payment Request.
- The PISP retrieves the payer's ASPSP's HostID (URI).

**Step 35**

The PISP provides the signed Payment Request message to the payer's ASPSP via their PSD2 API.

**Step 36**

- The payer ASPSP checks the integrity of all the information provided including the verification of both QESs, including all certificate chains.

[42] Alternatively the CustomerID could be supplied within the payer's IBAN attribute certificate

- The payer ASPSP checks the availability of funds on the consumer's account.
- The payer ASPSP prepares and submits the SCT Inst transaction to the payee ASPSP.

**Step 37**

- A confirmation message is returned from the payee ASPSP to the payer ASPSP.
- The payee ASPSP makes the funds available to the merchant.

**Step 38**

The payer ASPSP sends a notification message to the PISP about the execution of the SCT Inst transaction.

**Step 39**

The PISP sends a notification message to the payee's EUDIW about the successful transaction.

**Step 40**

- An e-receipt is sent by the payee's EUDIW to the payer's EUDIW. The e-receipt takes the form of a countersignature on the payer QES and includes the payment status.
- The payer's EUDIW displays the notification message about the successful transaction to the payer.

| Analysis MSCT Use case P2P-2 | |
|---|---|
| **Interoperability** | • Based on and governed by PSD2<br>• EC eIDAS2.0 framework<br>• The payer and the payee need to be subscribed to EUDIW that are provisioned with the identities and attribute certificates by PID providers, registered under the EUDIW interoperability framework.<br>• The specification of an interoperability standard allowing two EUDIW to communicate with each other over BLE. |
| **Challenges** | • The EUDIW needs to support the generation of AdES and QES under the eIDAS framework.<br>• The obligation of ASPSPs to accept SCA based on the EUDIW.<br>• Checking the revocation of the public key certificates used as specified by the eIDAS regulation, potentially creating latency in the transaction.<br>• The EUDIW app needs to store all eIDAS trusted root keys to enable public key certificate verifications offline[43].<br>• Requires a registration of the payee with the PISP. |

[43] Hereby it is assumed that the attribute certificates issued by the ASPSP rely on an eIDAS trusted root certificate.

| | |
|---|---|
| | • Information to the payer with respect to usage of the PISP (PSD 2 Arts. 44 and 45).[44]<br>• Specification of account holder name and IBAN attribute certificates by ASPSPs.<br>• The establishment of the supporting PKI for the issuance of the account holder name and IBAN attribute certificates by the ASPSPs.<br>• Standardisation of data transmitted between the payee's mobile device and the payer's mobile device.<br>• Lack of common specification for usage of BLE between EUDIWs.<br>• Support for the BLE proximity connection standard by the EUDIWs both the payer and the payee.<br>• Liability aspects need to be clarified.<br>• The PSD2 API needs to support the functionalities required (e.g. signed payment request, notification message, etc.).<br>• Education of PSU on usage of two different proximity technologies.<br>• The notification messages in step 38 is not included in the SCT Inst scheme. |

**Table 11: Analysis MSCT use case P2P-2**

*Notes:*

- All MSCT use cases described include the performance of an SCA. Obviously, if SCA is not required when an exemption is applied in accordance with PSD2 and the RTS, the corresponding steps will be omitted and the consumer would just confirm the transaction, e.g. by pressing a button on the consumer device.
- The interoperability models for MSCTs involving a PISP are analysed in Chapter 20 in the MSCT IG [10].
- The integrity of QR-codes is addressed in Chapter 10 in the MSCT IG [10].
- The minimum data elements in the payment request and notification messages are defined in Annex 5 in the MSCT IG [10].

[44] See EBA answer to Q&A 2020_5573.

### 5.3 MSCT use case C2B-5: Mobile device – Offline use case - Payment at a physical POI with consumer-presented QR-code involving a PISP – SCA via BLE using an MSCT app involving a fingerprint

This use case presents an example of consumer experience whereby their mobile device has no mobile network connection[45] and is used for a payment at a physical POI. In this use case two proximity technologies are used: a consumer-presented QR-code and BLE.

The consumer has preloaded a dedicated MSCT app onto their mobile device provided by their ASPSP that supports the generation of an Advanced Electronic Signature (AdES) as specified under the eIDAS framework[46] based on a dedicated asymmetric key pair[47]. It is further assumed that the QR-code provided by the MSCT app[48] contains the necessary information to establish a secure connection (see Chapter 9 MSCT IG [10]) between this app and the merchant POI via BLE for performing the SCA.

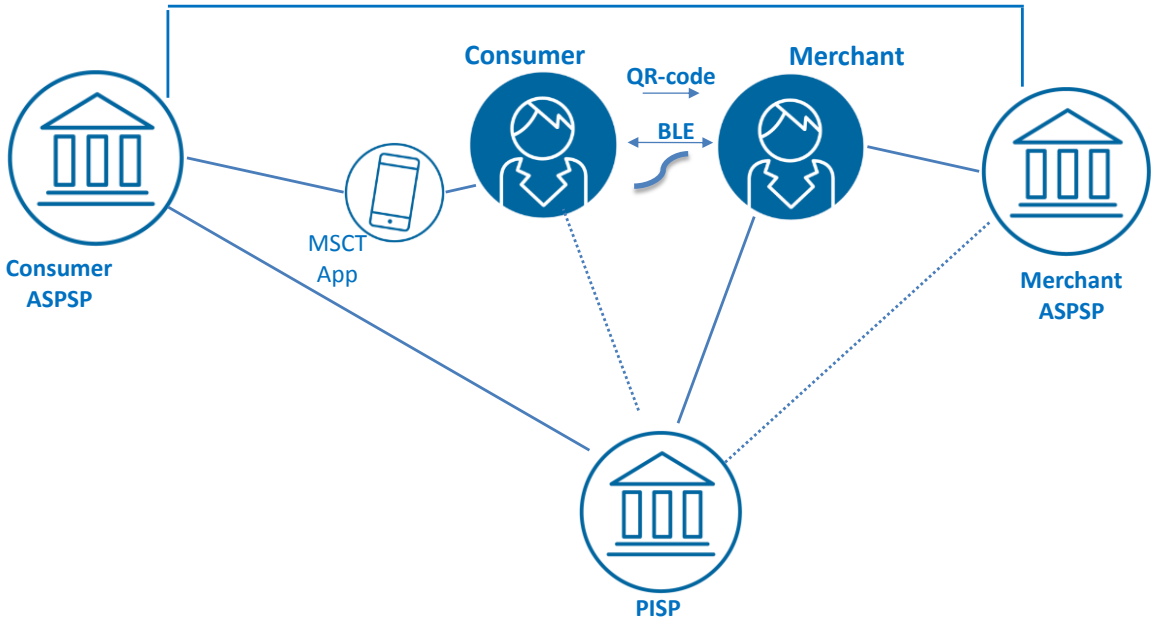



Figure 15: Actors in MSCT Use case C2B-5

Consumer and merchant, may, and frequently will, hold their payment accounts with different ASPSPs.

[45] If the mobile device of the consumer has internet connection, a similar use case could be considered whereby the QR-code could be used to establish an internet connection between the MSCT app and the merchant or the PISP to conduct the transaction.

[46] See https://ec.europa.eu/futurium/en/content/eidas-regulation-regulation-eu-ndeg9102014

[47] See also the EU Retail Payments Strategy including the use of EUid, eIDAS signatures and e-receipts, https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52020DC0592&from=EN.

[48] Unlike in some other MSCT use cases whereby an MSCT app is involved and the SCT Inst is initiated by the MSCT service provider, in this use case it is initiated by a PISP, involved on the merchant side.

The merchant has a contract with a PISP (= merchant MSCT service provider) that supports the PSD2 API, has downloaded dedicated software on their POI and agreed to make the required PISP information available to the consumer according to the PSD2 Arts. 44 and 45[49].

The PISP also has a dedicated asymmetric key pair to generate a QSEAL in accordance with the eIDAS framework.

The exchange of data between the MSCT app on the consumer's mobile device and the PISP is protected through symmetric encryption using a secret session key derived from the temporary ECDH key pairs that are generated for each transaction both by the MSCT app and by the PISP (see **Figure 17** for an overview of the cryptography).

In this payment transaction a strong customer authentication (see section 8.3 in the MSCT IG [10]) in accordance with PSD2 [2] is performed involving a fingerprint that unlocks a cryptographic private key held within the "separate secure execution environment" of the consumer's mobile device to create the AdES (see section 8.2 in the MSCT IG [10]).

[49] See also the EBA answer to Q&A 2020_5573.

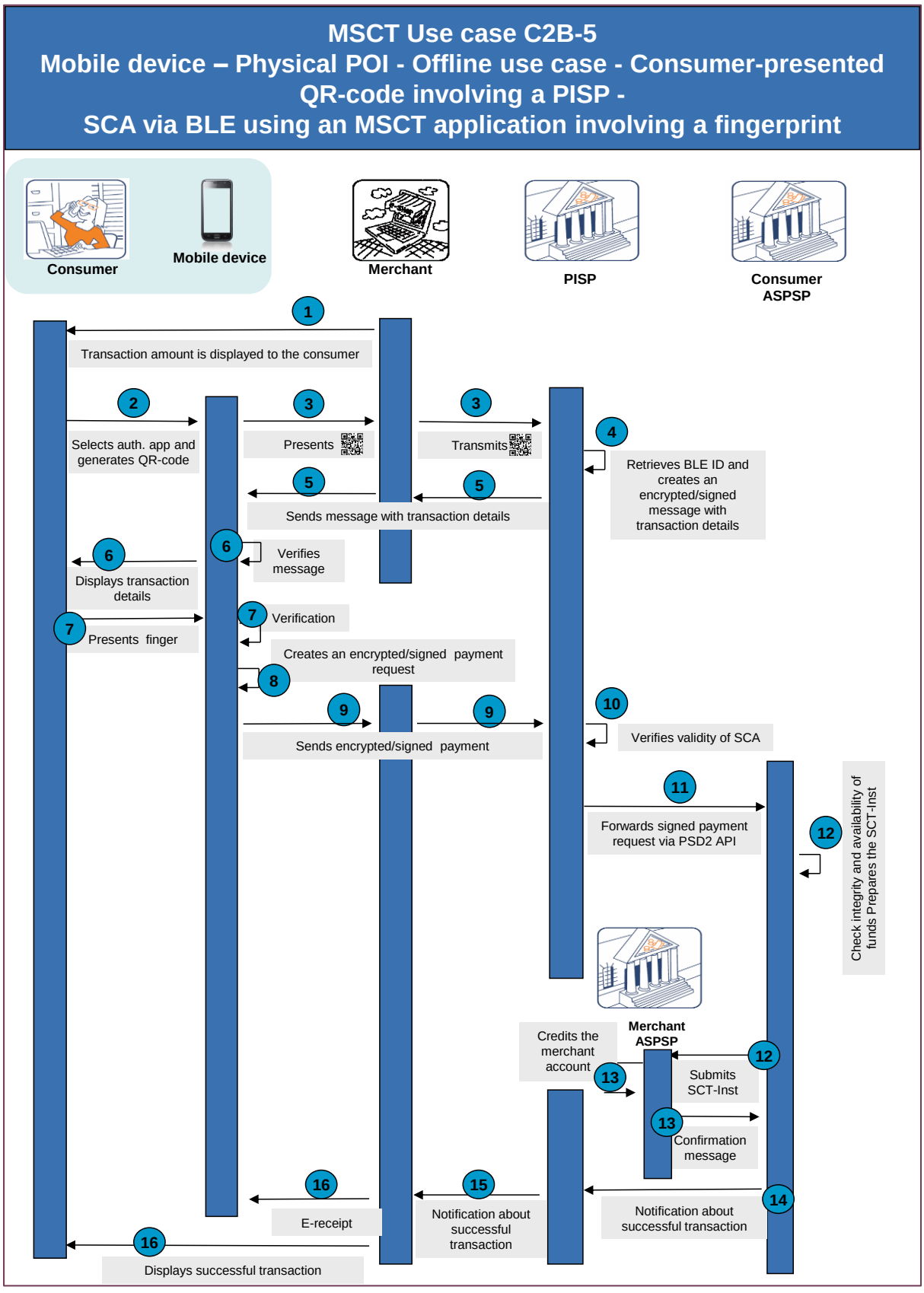



Figure 16: MSCT Use case C2B-5

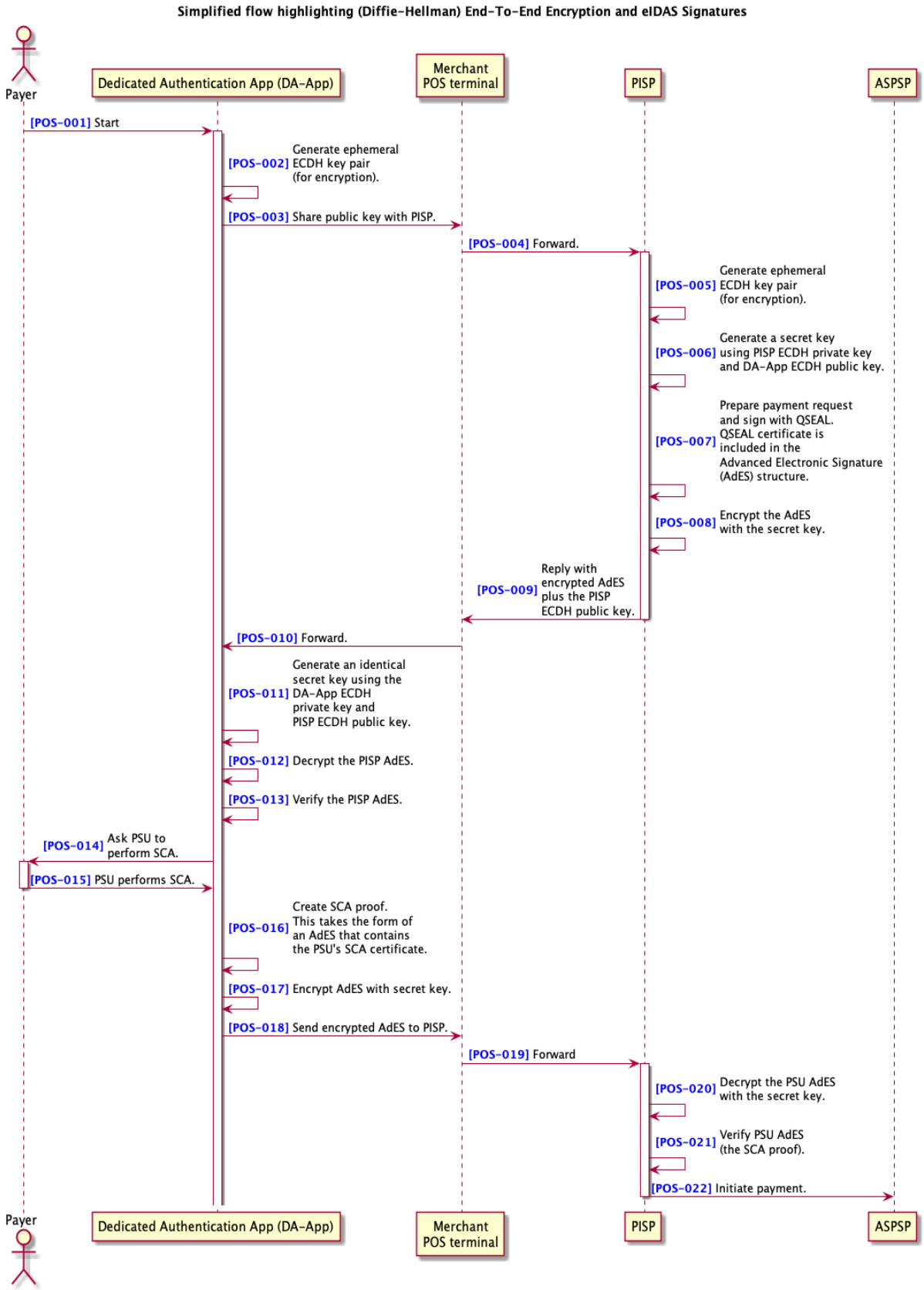



Figure 17: MSCT Use case C2B-5 – Overview cryptography

In the figure above, the following steps are illustrated:

**Step 0**

- As a prerequisite, consumers would need to download an MSCT app from their ASPSP that supports the generation of AdES under the eIDAS framework and the yet to be defined interoperability standard[50] between these apps and PISPs, including the generation of ECDH session keys. Moreover, the app stores all eIDAS root keys to enable public key certificate verifications.
- The merchant is subscribed to the PISP and has installed their software on the POI.
- The PISP also supports the generation of QSEALs under the eIDAS framework and the yet to be defined interoperability standard including the generation of ECDH session keys. It is further enabled to use the consumer ASPSP's PSD2 API.
- During the payment transaction, there is no mobile communication network required for the consumer's mobile device.

**Step 1**

The merchant enters the transaction amount which is displayed on the POI[51].

**Step 2**

- The consumer selects and opens the MSCT app on their mobile device.
- The MSCT app generates a session ECDH key pair and a dynamic QR code-containing the ECDH public key.

**Step 3**

- The consumer presents their QR-code, which is scanned by the merchant's POI. This gesture also represents the consumer's consent to the use of the PISP service[52].
- The POI retrieves the necessary information to establish a BLE connection with the MSCT app on the consumer's mobile device.
- The information contained in the QR-code is provided to the PISP.

**Step 4**

- The PISP retrieves the MSCT app ECDH public key and checks the merchant.
- The PISP generates an ECDH key pair and creates a message containing the transaction details, including - as a minimum - the merchant's name/trade name, IBAN_merchant, transaction identifier and the transaction amount. This message is signed with a PISP QSEAL and encrypted with a secret session key derived from the PISP private key and the MSCT app ECDH public key.

[50] This is considered to be covered by the EUDIW work.

[51] The display of the transaction amount by the POI may happen after step 3, since the consumer identification might have an impact on the final transaction amount.

[52] As clarified by the EBA answer to EBA Q&A 2020_5570.

**Step 5**

The encrypted signed message, including the PISP QSEAL public key certificate and the PISP ECDH public key is transferred to the merchant and from the merchant's POI to the MSCT app on the consumer's mobile device using BLE.

**Step 6**

- The MSCT app on the consumer's mobile device also generates the same secret session key from their ECDH private key and the PISP ECDH public key and decrypts the message.
- Next the MSCT app verifies the PISP QSEAL public key certificate and subsequently the PISP QSEAL (hereby implicitly authenticating the PISP).
- The transaction details (including as a minimum the transaction amount and merchant name/trade name/IBAN) are displayed to the consumer by the MSCT app.
- The MSCT app optionally offers the consumer to select a payment account or presents a default account for approval.

**Step 7**

- The consumer authenticates and confirms the transaction by presenting a finger to their device.
- The mobile device verifies the fingerprint.

**Step 8**

- Upon successful verification of the fingerprint, the MSCT app on the consumer's mobile device further completes the message received with the IBAN_consumer, CustomerID and the ASPSP's HostID (URI).
- The app generates an AdES on the message (dynamically linked to all data elements).
- The app subsequently encrypts the signed data using the secret session key.

**Step 9**

The encrypted/signed message including the AdES public key certificate is transferred from the app via BLE to the POI and further transferred to the PISP.

**Step 10**

- The PISP checks the message received by decrypting the message using the secret session key.
- Next the PISP optionally verifies the AdES public key certificate and subsequently the AdES, and can thereby verify the validity of the SCA.
- The PISP retrieves the consumer's ASPSP's HostID (URI).

**Step 11**

The PISP provides the consumer-signed message as a "signed payment request" to the consumer's ASPSP via their PSD2 API.

**Step 12**

- The consumer ASPSP checks the integrity of all the information provided including the verification of the AdES.
- The consumer ASPSP checks the availability of funds on the consumer's account.
- The consumer ASPSP prepares and submits the SCT Inst transaction to the merchant ASPSP.

**Step 13**

- A confirmation message is returned from the merchant's ASPSP to the consumer's ASPSP.
- The merchant's ASPSP makes the funds available to the merchant.

**Step 14**

The consumer ASPSP sends a notification message to the PISP about the execution of the SCT Inst transaction.

**Step 15**

The PISP sends a notification message to the merchant about the successful transaction.

**Step 16**

The merchant POI displays the successful transaction and provides an e-receipt to the consumer's mobile device via BLE.

| Analysis MSCT Use case C2B-5 | |
|---|---|
| **Interoperability** | • Based on and governed by PSD2<br>• EC eIDAS framework<br>• Yet to be defined BLE connection standard between the MSCT app and the PISP software on the POI. |
| **Challenges** | • The MSCT app from the consumer ASPSP must support the generation of AdES[53] (including the generation of the session key).<br>• The PISP needs to support the generation of QSEALs under the eIDAS framework (including the generation of the session key).<br>• Requires a contract between the merchant and the PISP.<br>• Standardisation of data transmitted between the MSCT app and the POI.<br>• Information to the consumer with respect to usage of the PISP (PSD 2 Arts. 44 and 45).[54] |

[53] Note that the draft regulation for eIDAS2.0 currently specifies the usage of a QES for the digital identity wallet (see Annex 1).

[54] See EBA answer to Q&A 2020_5573.

| | |
|---|---|
| | • Lack of common specification for usage of BLE for payments at the POI and availability of BLE at POI terminals.<br>• Support for the BLE connection standard both by the MSCT app and the POI.<br>• Liability aspects need to be clarified.<br>• The notification messages in steps 14 and 15 are not included in the SCT Inst scheme.<br>• The PSD2 API needs to support the functionalities required (e.g. signed payment request, notification message, etc.).<br>• Education of PSU on usage of two different proximity technologies.<br>• The notification messages in step 14 is not included in the SCT Inst scheme. |

**Table 12: Analysis MSCT Use case C2B-5**

*Notes:*

- All MSCT use cases described include the performance of an SCA. Obviously, if SCA is not required when an exemption is applied in accordance with PSD2 and the RTS, the corresponding steps will be omitted and the consumer would just confirm the transaction, e.g. by pressing a button on the consumer device.
- The interoperability models for MSCTs involving a PISP are analysed in Chapter 20 in the MSCT IG [10].
- The integrity of QR-codes is addressed in Chapter 10 in the MSCT IG [10].
- The minimum data elements in the payment request and notification messages are defined in Annex 5 in the MSCT IG [10].

### 5.4 MSCT use case C2B-6: Mobile device – Offline use case - Payment at a physical POI with merchant-presented QR-code involving a PISP – SCA via BLE using an MSCT app involving a mobile code

This use case presents an example of consumer experience whereby their mobile device has no mobile network connection[55] and is used for a payment at a physical POI. In this use case two proximity technologies are used: a merchant-presented QR-code and BLE.

The consumer has preloaded a dedicated MSCT app onto their mobile device provided by their ASPSP (= consumer's MSCT service provider) that supports the generation of an AdES as specified under the eIDAS framework based on a dedicated asymmetric key pair. It is further assumed that the QR-code provided by the merchant POI[56] contains the necessary information to establish a secure connection (see Chapter 9 MSCT IG [10]) between this POI and the MSCT app via BLE for performing the SCA.

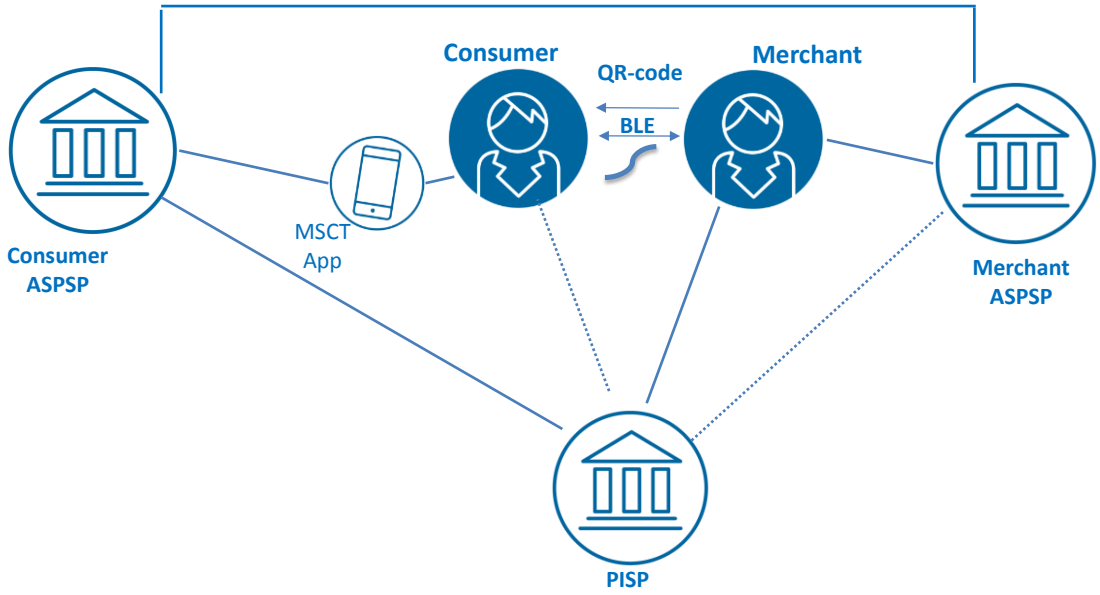



Figure 18: Actors in MSCT Use case C2B-6

Consumer and merchant, may, and frequently will, hold their payment accounts with different ASPSPs.

The merchant has a contract with a PISP (= merchant MSCT service provider) that supports the PSD2 API, has downloaded dedicated software on their POI and agreed to make the required PISP information available to the consumer according to the PSD2 Arts. 44 and 45[57].

[55] If the mobile device of the consumer has internet connection, a similar use case could be considered whereby the QR-code could be used to establish an internet connection between the MSCT app and the merchant or the PISP to conduct the transaction.

[56] Unlike in some other MSCT use cases whereby an MSCT app is involved and the SCT Inst is initiated through the consumer's MSCT service provider, in this use case it is initiated by a PISP, involved on the merchant side.

[57] See also the EBA answer to Q&A 2020_5573.

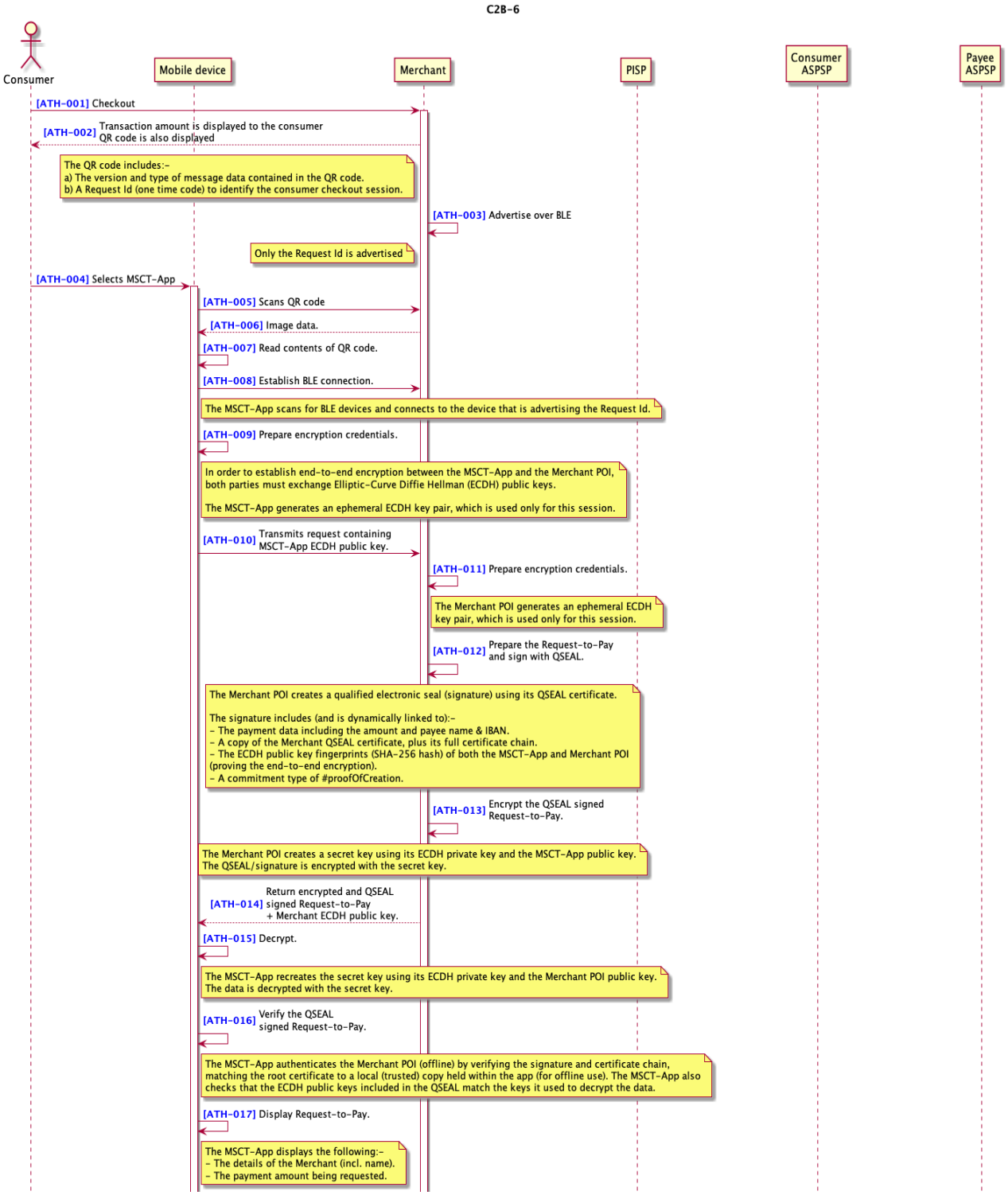
The PISP also has a dedicated asymmetric key pair to generate a QSEAL in accordance with the eIDAS framework, with the corresponding certificates.

The exchange of data between the MSCT app on the consumer's mobile device and the PISP is protected through symmetric encryption using a secret session key derived from dedicated session ECDH key pairs that are generated for each transaction both by the MSCT app and by the PISP from the respective ECDH public keys of the MSCT app and the PISP (see **Figure 20** for an overview of the cryptography).

In this payment transaction a strong customer authentication (see section 8.3 in the MSCT IG [10]) in accordance with PSD2 [2] is performed involving a mobile code that unlocks a cryptographic private key held within the "separate secure execution environment" of the consumer's mobile device to create the AdES (see section 8.2 in the MSCT IG [10]).

No mobile network connectivity of the consumer's mobile device is required in this use case.

**Figure 19: MSCT Use case C2B-6 (to be developed for final version)**


C2B-6
Consumer
Mobile device
Merchant
PISP
Consumer ASPSP
Payee ASPSP
[ATH-001] Checkout
[ATH-002] Transaction amount is displayed to the consumer QR code is also displayed
The QR code includes:-
a) The version and type of message data contained in the QR code.
b) A Request Id (one time code) to identify the consumer checkout session.
[ATH-003] Advertise over BLE
Only the Request Id is advertised
[ATH-004] Selects MSCT-App
[ATH-005] Scans QR code
[ATH-006] Image data.
[ATH-007] Read contents of QR code.
[ATH-008] Establish BLE connection.
The MSCT-App scans for BLE devices and connects to the device that is advertising the Request Id.
[ATH-009] Prepare encryption credentials.
In order to establish end-to-end encryption between the MSCT-App and the Merchant POI, both parties must exchange Elliptic-Curve Diffie Hellman (ECDH) public keys.
The MSCT-App generates an ephemeral ECDH key pair, which is used only for this session.
[ATH-010] Transmits request containing MSCT-App ECDH public key.
[ATH-011] Prepare encryption credentials.
The Merchant POI generates an ephemeral ECDH key pair, which is used only for this session.
[ATH-012] Prepare the Request-to-Pay and sign with QSEAL.
The Merchant POI creates a qualified electronic seal (signature) using its QSEAL certificate.
The signature includes (and is dynamically linked to):-
- The payment data including the amount and payee name & IBAN.
- A copy of the Merchant QSEAL certificate, plus its full certificate chain.
- The ECDH public key fingerprints (SHA-256 hash) of both the MSCT-App and Merchant POI (proving the end-to-end encryption).
- A commitment type of #proofOfCreation.
[ATH-013] Encrypt the QSEAL signed Request-to-Pay.
The Merchant POI creates a secret key using its ECDH private key and the MSCT-App public key.
The QSEAL/signature is encrypted with the secret key.
[ATH-014] Return encrypted and QSEAL signed Request-to-Pay + Merchant ECDH public key.
[ATH-015] Decrypt.
The MSCT-App recreates the secret key using its ECDH private key and the Merchant POI public key.
The data is decrypted with the secret key.
[ATH-016] Verify the QSEAL signed Request-to-Pay.
The MSCT-App authenticates the Merchant POI (offline) by verifying the signature and certificate chain, matching the root certificate to a local (trusted) copy held within the app (for offline use). The MSCT-App also checks that the ECDH public keys included in the QSEAL match the keys it used to decrypt the data.
[ATH-017] Display Request-to-Pay.
The MSCT-App displays the following:-
- The details of the Merchant (incl. name).
- The payment amount being requested.

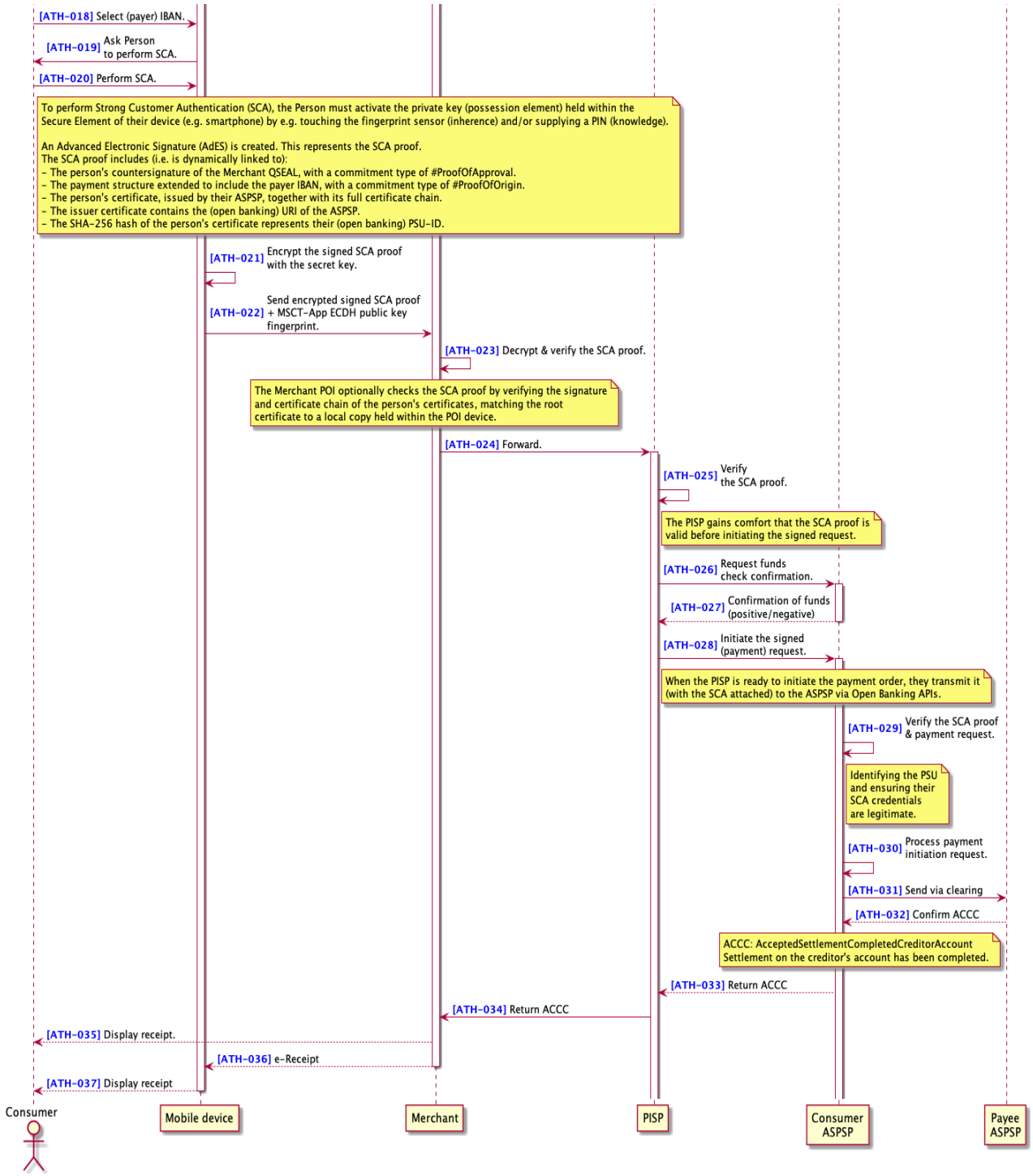



Figure 20: MSCT Use case C2B-6 – Overview cryptography

In the figure above, the following steps are illustrated:

**Step 0**

- As a prerequisite, consumers would need to download an MSCT app from their ASPSP that supports the generation of AdES under the eIDAS framework and the yet to be defined interoperability standard between these apps and PISPs, including the generation of ECDH session keys. They have also been provisioned with an Identity certificate for SCA (corresponding to their account holder name) and with an IBAN attribute certificate by their ASPSP. In addition, the MSCT app should store all eIDAS root keys to enable public key certificate verifications offline.
- The merchant is registered with the PISP who is enabled to use the consumer ASPSP's PSD2 API and has installed their software on the POI.
- The PISP software supports the generation of QSEALs under the eIDAS framework, the storage of the corresponding QSEAL certificates and the yet to be defined interoperability standard including the generation of ECDH session keys.
- The merchant has been provisioned with a merchant name / trade name (of the account holder) attribute certificate and an IBAN attribute certificate by their ASPSP.
- During the payment transaction, there is no mobile communication network required for the consumer's mobile device.

**Step 1**

- The merchant enters the transaction amount which is displayed on the POI.
- The POI generates a QR-code which contains
  - The version and type of message data contained in the QR-code.
  - A RequestId to identify the consumer checkout session

  and is displayed on the POI.

**Step 2**

The POI advertises the RequestId over BLE.

**Step 3**

- The consumer selects and opens the MSCT app on their mobile device.
- The consumer scans the QR-code with their MSCT app. This gesture may represent the consumer's consent to the use of the PISP service[58].
- The MSCT app retrieves the RequestID from the QR-code.

**Step 4**

- The MSCT app scans for a BLE advertisement containing the RequestId.
- The MSCT app then connects to the POI over BLE.

**Step 5**

- The MSCT-app generates a temporary ECDH key pair, which is used only for this session.
- The MSCT app transmits the ECDH public key to the POI over the BLE connection.

[58] In analogy to the EBA answer received on Q&A 2020_5570.

**Step 6**

The merchant POI generates a temporary ECDH key pair, which is used only for this session.

**Step 7**

The merchant POI prepares a payment request and signs it with their QSEAL. The signature includes (and is dynamically linked to):

- The payment data including the transaction amount and merchant name/trade name and IBAN_merchant;
- A copy of the merchant name/trade name attribute certificate;
- A copy of the merchant QSEAL certificate;
- A copy of the IBAN_merchant attribute certificate;
- The full certificate chains of all certificates;
- The SHA-256 hash of the ECDH public keys of both the MSCT app and merchant POI;
- A commitment type of #ProofOfCreation.

**Step 8**

- The POI generates a secret session key using its ECDH private key and the MSCT-app ECDH public key.
- The POI encrypts the QSEAL signed payment request with the secret session key.

**Step 9**

The POI sends the encrypted QSEAL signed payment request together with a copy of the merchant's ECDH public key to the consumer's MSCT-app via BLE.

**Step 10**

- The MSCT-app generates the secret session key using its ECDH private key and the POI ECDH public key and decrypts the payment request with this secret session key.
- The MSCT-app authenticates the POI by verifying the QSEAL on the payment request and the certificate chain, matching the root certificate to a local (trusted) copy held within the app (for offline use). The MSCT-app also checks that the ECDH public keys included in the QSEAL match the keys it used to decrypt the encrypted signed payment request.

**Step 11**

- The transaction details (including as a minimum the merchant's name/trade name/IBAN and the transaction amount) are displayed to the consumer by the MSCT app.
- The MSCT app optionally offers the consumer to select a payment account or presents a default account for approval.

**Step 12**

- The consumer authenticates and confirms the transaction by entering a mobile code on their mobile device.
- Upon successful verification of the mobile code, an AdES is generated using the private key stored on the consumer's mobile device on the following data:
    - The consumer's countersignature of the merchant QSEAL, with a commitment type of #ProofOfApproval.

  - The consumer's Identity certificate for SCA, issued by their PID provider,
  - The consumer's IBAN attribute certificate, issued by their ASPSP, containing the consumer's ASPSP Host ID (URI);
  - The SHA-256 hash of the consumer's Identity certificate for SCA which could represent their CustomerID[59].
- The MSCT app encrypts the signed payment request message with the secret session key.

**Step 13**

The encrypted signed payment request message is transmitted together with a copy of the MSCT app ECDH public key SHA-256 hash from the MSCT app to the POI via BLE.

**Step14**

- The POI decrypts the encrypted signed payment request message using the secret session key.
- The PISP software optionally checks the SCA proof by verifying the consumer's AdES signature and the full certificate chain of the consumer's certificates, matching each root certificate to a local copy held within the POI device.

**Step 15**

The POI transmits the signed payment request message to the merchant's PISP.

**Step 16**

- The PISP retrieves the consumer's ASPSP's HostID (URI) and derives the CustomerID.

**Step 17**

The PISP provides the (consumer) signed payment request message to the consumer's ASPSP via their PSD2 API.

**Step 18**

- The consumer ASPSP checks the integrity of all the information provided including the verification of the AdES.
- The consumer ASPSP checks the availability of funds on the consumer's account.
- The consumer ASPSP prepares and submits the SCT Inst transaction to the merchant ASPSP.

[59] Alternatively the CustomerID could be supplied within the payer's IBAN attribute certificate

**Step 19**

- A confirmation message is returned from the merchant ASPSP to the consumer ASPSP.
- The merchant ASPSP makes the funds available to the merchant.

**Step 20**

The consumer ASPSP sends a notification message to the PISP about the execution of the SCT Inst transaction.

**Step 21**

The PISP sends a notification message to the merchant about the successful transaction.

**Step 22**

- An e-receipt is sent by the merchant POI to the consumer's MSCT app.
- The MSCT app displays the notification message about the successful transaction to the consumer.

| Analysis MSCT Use case C2B-6 | |
|---|---|
| **Interoperability** | • Based on and governed by PSD2<br>• EC eIDAS framework<br>• The specification of an interoperability standard enabling the MSCT app and the POI to communicate with each other over BLE. |
| **Challenges** | • The MSCT app from the consumer ASPSP must support the generation of AdES under the eIDAS framework (including the generation of the session key).<br>• The PISP needs to support the generation of QSEALs under the eIDAS framework.<br>• Specification of merchant name /trade name and IBAN attribute certificates.<br>• The establishment of the supporting PKI for the issuance of the merchant name /trade name and IBAN attribute certificates by the ASPSPs.<br>• The PISP needs to support the generation of QSEALs under the eIDAS framework.<br>• The MSCT app and the POIs need to locally store all trusted root certificates to enable public key certificate verifications offline.<br>• Requires a contract between the merchant and the PISP.<br>• Standardisation of data transmitted between the MSCT app and the POI. |

| | |
|---|---|
| | • Information to the consumer with respect to usage of the PISP (PSD 2 Arts. 44 and 45).[60]<br>• Lack of common specification for usage of BLE for payments at the POI and availability of BLE at POI terminals.<br>• Support for the BLE connection standard both by the MSCT app and the POI.<br>• Liability aspects need to be clarified.<br>• The notification messages in step 20 is not included in the SCT Inst scheme.<br>• The PSD2 API needs to support the functionalities required (e.g. signed payment request, notification message, etc.).<br>• Education of PSU on usage of two different proximity technologies. |

**Table 13: Analysis MSCT Use case C2B-6**

*Notes:*

- The interoperability models for MSCTs involving a PISP are analysed in Chapter 20 in the MSCT IG [10].
- The integrity of QR-codes is addressed in Chapter 10 in the MSCT IG [10].
- The minimum data elements in the payment request and notification messages are defined in Annex 5 in the MSCT IG [10].

[60] See EBA answer to Q&A 2020_5573.

## 5.5 MSCT use case C2B-7: Mobile device – Offline use case - Payment at a physical POI with merchant-presented QR-code involving a PISP – SCA via BLE using an EUDIW involving a fingerprint

This use case presents an example of consumer experience whereby their mobile device has no mobile network connection[61] and is used for a payment at a physical POI. In this use case two proximity technologies are used: a merchant-presented QR-code and BLE.

The consumer has preloaded an EUDIW onto their mobile device and has been provisioned with an Identity Certificate for SCA and attribute certificates[62] from a PID provider which operates under the eIDAS framework. Moreover, the EUDIW has also been provisioned with the IBAN attribute certificate by the consumer's ASPSP. The EUDIW supports the generation of a QES as specified under the eIDAS framework based on a dedicated asymmetric key pair. It is further assumed that the QR-code provided by the merchant POI contains the necessary information to establish a secure connection (see Chapter 9 MSCT IG [10]) between the POI and the EUDIW via BLE for performing the SCA.

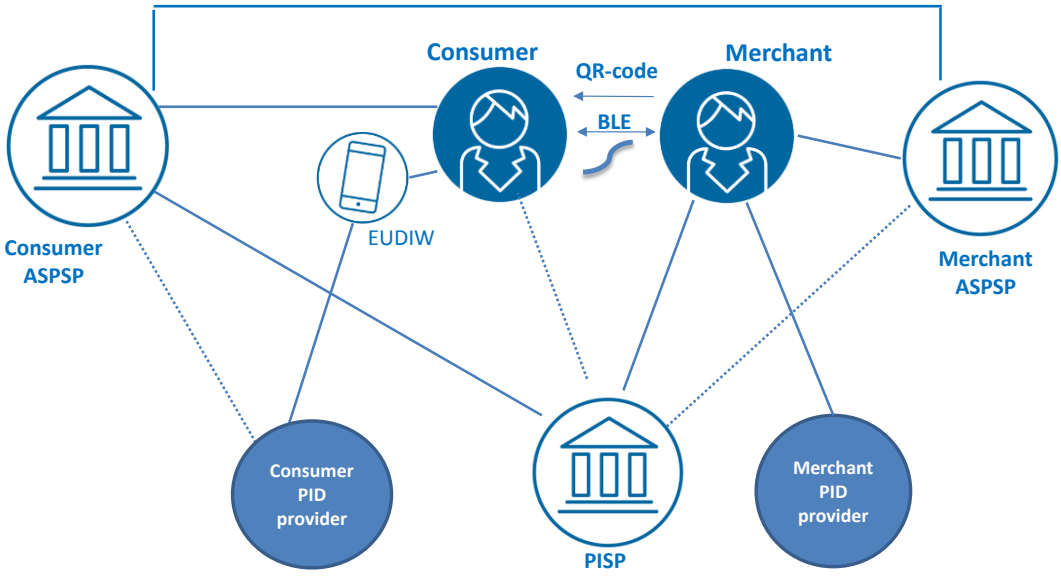



Figure 21: Actors in MSCT Use case C2B-7

Consumer and merchant, may, and frequently will, hold their payment accounts with different ASPSPs. The consumer's ASPSPs relies on the usage of the EUDIW and the eIDAS2.0 for the PSU authentication.

[62] It is hereby assumed that the consumer name attribute certificate issued by the PID provider corresponds to their account holder name.

The merchant has a contract with a PISP (= merchant MSCT service provider) that supports the PSD2 API, has downloaded dedicated software on their POI and agreed to make the required PISP information available to the consumer according to the PSD2 Arts. 44 and 45[63].

The PISP also has a dedicated asymmetric key pair to generate a QSEAL in accordance with the eIDAS framework with the corresponding certificates.

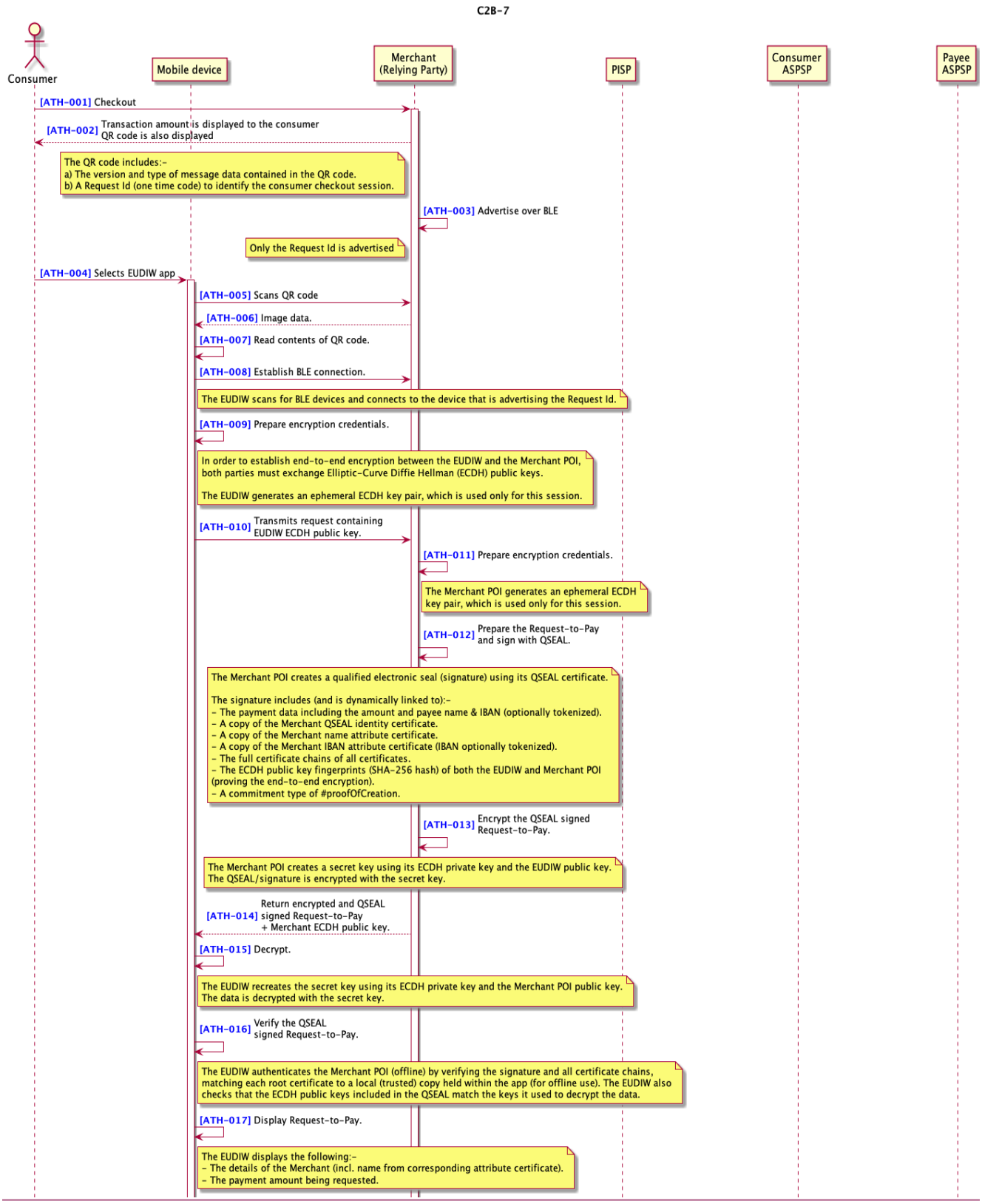
The exchange of data between the EUDIW on the consumer's mobile device and the PISP is protected through symmetric encryption using a secret session key derived from dedicated session ECDH key pairs that are generated for each transaction both by the EUDIW and the PISP from the respective ECDH public keys of the EUDIW and the PISP (see **Figure 23** for an overview of the cryptography).

In this payment transaction a strong customer authentication (see section 8.3 in the MSCT IG [10]) in accordance with PSD2 [2] is performed involving a fingerprint that unlocks a cryptographic private key held within the "separate secure execution environment" of the consumer's mobile device to create the QES (see section 8.2 in the MSCT IG [10]).

No mobile network connectivity of the consumer's mobile device is required in this use case.

**Figure 22: MSCT Use case C2B-6 (to be developed for final version)**

[63] See also the EBA answer to Q&A 2020_5573.


C2B-7
Consumer
Mobile device
Merchant (Relying Party)
PISP
Consumer ASPSP
Payee ASPSP
[ATH-001] Checkout
[ATH-002] Transaction amount is displayed to the consumer
QR code is also displayed
The QR code includes:-
a) The version and type of message data contained in the QR code.
b) A Request Id (one time code) to identify the consumer checkout session.
[ATH-003] Advertise over BLE
Only the Request Id is advertised
[ATH-004] Selects EUDIW app
[ATH-005] Scans QR code
[ATH-006] Image data.
[ATH-007] Read contents of QR code.
[ATH-008] Establish BLE connection.
The EUDIW scans for BLE devices and connects to the device that is advertising the Request Id.
[ATH-009] Prepare encryption credentials.
In order to establish end-to-end encryption between the EUDIW and the Merchant POI, both parties must exchange Elliptic-Curve Diffie Hellman (ECDH) public keys.
The EUDIW generates an ephemeral ECDH key pair, which is used only for this session.
[ATH-010] Transmits request containing EUDIW ECDH public key.
[ATH-011] Prepare encryption credentials.
The Merchant POI generates an ephemeral ECDH key pair, which is used only for this session.
[ATH-012] Prepare the Request-to-Pay and sign with QSEAL.
The Merchant POI creates a qualified electronic seal (signature) using its QSEAL certificate.
The signature includes (and is dynamically linked to):-
- The payment data including the amount and payee name & IBAN (optionally tokenized).
- A copy of the Merchant QSEAL identity certificate.
- A copy of the Merchant name attribute certificate.
- A copy of the Merchant IBAN attribute certificate (IBAN optionally tokenized).
- The full certificate chains of all certificates.
- The ECDH public key fingerprints (SHA-256 hash) of both the EUDIW and Merchant POI (proving the end-to-end encryption).
- A commitment type of #proofOfCreation.
[ATH-013] Encrypt the QSEAL signed Request-to-Pay.
The Merchant POI creates a secret key using its ECDH private key and the EUDIW public key.
The QSEAL/signature is encrypted with the secret key.
[ATH-014] Return encrypted and QSEAL signed Request-to-Pay + Merchant ECDH public key.
[ATH-015] Decrypt.
The EUDIW recreates the secret key using its ECDH private key and the Merchant POI public key.
The data is decrypted with the secret key.
[ATH-016] Verify the QSEAL signed Request-to-Pay.
The EUDIW authenticates the Merchant POI (offline) by verifying the signature and all certificate chains, matching each root certificate to a local (trusted) copy held within the app (for offline use). The EUDIW also checks that the ECDH public keys included in the QSEAL match the keys it used to decrypt the data.
[ATH-017] Display Request-to-Pay.
The EUDIW displays the following:-
- The details of the Merchant (incl. name from corresponding attribute certificate).
- The payment amount being requested.

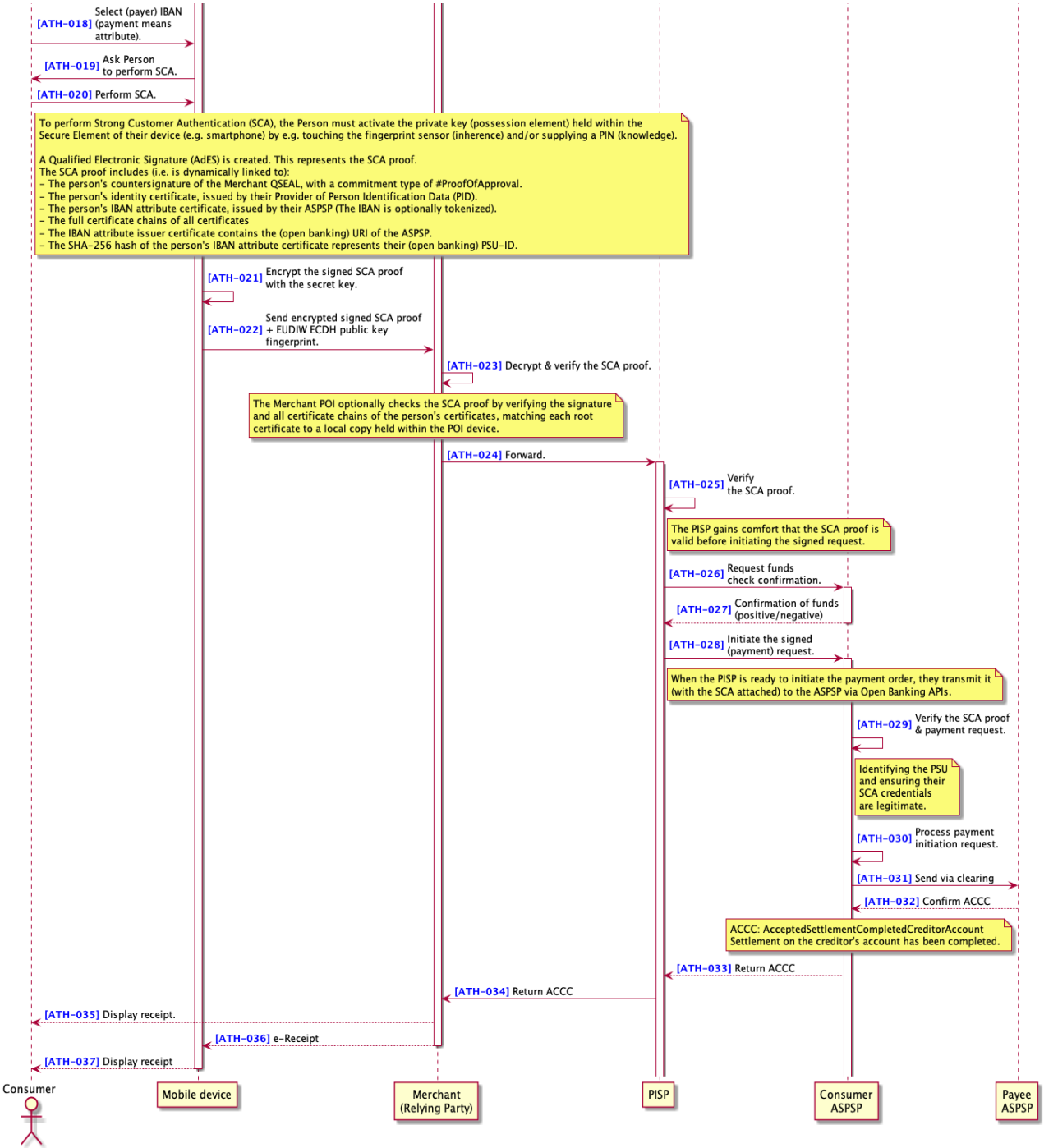



Figure 23: MSCT Use case C2B-7 – Overview cryptography

In the figure above, the following steps are illustrated:

**Step 0**

- As a prerequisite, consumers would need to have downloaded an EUDIW and be provisioned with Strong User Authentication Identity[64] and attribute certificates from a PID provider and also with an IBAN attribute certificates from their ASPSP. It is assumed that the EUDIW supports the generation of QES, including the cryptographic keys needed, under the eIDAS framework and are compliant with an interoperability standard allowing the EUDIW to communicate with the POI over a BLE proximity connection. It is also assumed this connection will be encrypted with ECDH session keys. In addition, the EUDIW should store all eIDAS root keys[65] to enable public key certificate verifications offline.
- The merchant is registered with the PISP who is enabled to use the consumer's ASPSP PSD2 API and has installed their software on the POI.
- The PISP also supports the generation of QSEALs under the eIDAS framework, the storage of the corresponding QSEAL certificates and the yet to be defined interoperability standard including the generation of ECDH session keys.
- The merchant has been provisioned with a merchant name/trade name (of the account holder) attribute certificate and an IBAN attribute certificate by their ASPSP.
- During the payment transaction, there is no mobile communication network required for the consumer's mobile device.

**Step 1**

- The merchant enters the transaction amount which is displayed on the POI.
- The POI generates a QR-code which contains
  - The version and type of message data contained in the QR-code.
  - A RequestId to identify the consumer checkout session

  and is displayed on the POI.

**Step 2**

The POI advertises the RequestId over BLE.

**Step 3**

- The consumer selects and opens the EUDIW on their mobile device.
- The consumer scans the QR-code with their EUDIW. This gesture may represent the consumer's consent to the use of the PISP service[66].
- The EUDIW retrieves the RequestID from the QR-code.

**Step 4**

- The consumer EUDIW scans for a BLE advertisement containing the RequestId.

---

[64] It is hereby assumed that the consumer Strong User Authentication attribute certificate corresponds to their account holder name.

[65] Based on an eIDAS Trusted List of root certificates.

[66] In analogy to the EBA answer received on Q&A 2020_5570.

- The consumer EUDIW then connects to the POI over BLE.

**Step 5**

- The EUDIW generates a temporary ECDH key pair, which is used only for this session.
- The EUDIW transmits the ECDH public key to the POI over the BLE connection.

**Step 6**

The merchant POI generates a temporary ECDH key pair, which is used only for this session.

**Step 7**

The merchant POI prepares a payment request and signs it with their QSEAL. The signature includes (and is dynamically linked to):-

- The payment data including the transaction amount and merchant name/trade name and IBAN_merchant;
- A copy of the merchant QSEAL certificate;
- A copy of the merchant name/trade name attribute certificate;
- A copy of the merchant IBAN attribute certificate;
- The full certificate chains of all certificates;
- The SHA-256 hash of both the ECDH public keys of both the consumer's EUDIW and merchant POI;
- A commitment type of #ProofOfCreation.

**Step 8**

- The merchant POI generates a secret session key using its ECDH private key and the EUDIW public key.
- The POI encrypts the QSEAL signed payment request with the secret session key.

**Step 9**

The POI sends the encrypted QSEAL signed payment request together with a copy of the merchant's ECDH public key to the consumer's EUDIW via BLE

**Step 10**

- The EUDIW generates the secret session key using its ECDH private key and the POI ECDH public key.
- The EUDIW decrypts the encrypted signed payment request with this secret session key.
- The EUDIW authenticates the POI by verifying the QSEAL signature on the payment request and the certificate chain, matching the root certificate to a local (trusted) copy held within the wallet (for offline use). The EUDIW also checks that the ECDH public keys included in the QSEAL match the keys it used to decrypt the encrypted signed payment request.

**Step 11**

- The transaction details (including the merchant's name/trade name/IBAN and the transaction amount) are displayed to the consumer by the EUDIW.
- The EUDIW optionally offers the consumer to select a payment account or presents a default account for approval.

**Step 12**

- The consumer authenticates and confirms the transaction by presenting their finger to their mobile device.
- Upon successful verification of the fingerprint, the EUDIW generates an QES using a private key, stored in the EUDIW on the consumer's mobile device on the following data:
    - The consumer's countersignature of the merchant QSEAL, with a commitment type of #ProofOfApproval;
    - The consumer's Identity Certificate for SCA (Strong User Authentication certificate), issued by their PID provider;
    - The consumer's IBAN attribute certificate, issued by their ASPSP, containing the the consumer's ASPSP HostID (URI);
    - The full certificate chains of all certificates;
    - The IBAN attribute issuer certificate;
    - The SHA-256 hash of the consumer's Identity Certificate for SCA which could represent their CustomerID.
- The MSCT app encrypts the signed payment request message with the secret session key.

**Step 13**

The encrypted signed payment request message is transmitted together with a copy of the EUDIW ECDH public key SHA-256 hash from the EUDIW to the POI via BLE.

**Step 14**

- The POI decrypts the encrypted signed payment request using the secret session key.
- The PISP software optionally checks the SCA proof by verifying the consumer's QES signature and the full certificate chain of the consumer's certificates, matching each root certificate to a local copy held within the POI device.

**Step 15**

The POI transmits the signed payment request message to the merchant's PISP.

**Step 16**

The PISP retrieves the consumer's ASPSP HostID (URI) and derives the CustomerID.

**Step 17**

The PISP provides the consumer-signed payment request to the consumer's ASPSP via their PSD2 API.

**Step 18**

- The consumer ASPSP checks the integrity of all the information provided including the verification of the QES.
- The consumer ASPSP checks the availability of funds on the consumer's account.
- The consumer ASPSP prepares and submits the SCT Inst transaction to the merchant ASPSP.

**Step 19**

- A confirmation message is returned from the merchant ASPSP to the consumer ASPSP.
- The merchant ASPSP makes the funds available to the merchant.

**Step 20**

The consumer ASPSP sends a notification message to the PISP about the execution of the SCT Inst transaction.

**Step 21**

The PISP sends a notification message to the merchant about the successful transaction.

**Step 22**

- An e-receipt is sent by the merchant POI to the consumer's EUDIW.
- The EUDIW displays the notification message about the successful transaction to the consumer.

| Analysis MSCT Use case C2B-7 | |
|---|---|
| **Interoperability** | • Based on and governed by PSD2<br>• EC eIDAS framework<br>• The specification of an interoperability standard enabling the EUDIW and the POI to communicate with each other over BLE. |
| **Challenges** | • The EUDIW needs to support the generation of QES under the eIDAS framework (including the generation of the session key).<br>• The PISP needs to support the generation of QSEALs under the eIDAS framework.<br>• Specification of merchant name/trade name and IBAN attribute certificates.<br>• The establishment of the supporting PKI for the issuance of the merchant name/trade name and IBAN certificates by the ASPSPs.<br>• The EUDIW and the POIs need to locally store all trusted root certificates to enable public key certificate verifications offline.<br>• Requires a contract between the merchant and the PISP.<br>• Standardisation of data transmitted between the EUDIW and the POI.<br>• Information to the consumer with respect to usage of the PISP (PSD 2 Arts. 44 and 45).[67]<br>• Lack of common specification for usage of BLE for payments at the POI and availability of BLE at POI terminals. |

[67] See EBA answer to Q&A 2020_5573.

| | |
|---|---|
| | • Support for the BLE connection standard both by the EUDIW and the POI.<br>• Liability aspects need to be clarified.<br>• The notification message in step 20 is not included in the SCT Inst scheme.<br>• The PSD2 API needs to support the functionalities required (e.g. signed payment request, notification message, etc.).<br>• Education of PSU on usage of two different proximity technologies. |

Table 14: Analysis MSCT Use case C2B-7

*Notes:*

- The interoperability models for MSCTs involving a PISP are analysed in Chapter 20 in the MSCT IG [10].
- The integrity of QR-codes is addressed in Chapter 10 in the MSCT IG [10].
- The minimum data elements in the payment request and notification messages are defined in Annex 5 in the MSCT IG [10].

# 6 Usage of NFC or BLE as proximity technologies for MSCTs

## 6.1 Introduction

Different proximity technologies have entered the market over the past years that may be used to conduct mobile payments. In this document the technologies NFC and BLE are considered. Note that the usage of QR-codes has been handled in the MSCT IG [10] and in EPC024-22 [14]. It is also to be noted that other new technologies such as ultra-wide band (UWB) and ultrasonic are emerging but the payment market adoption is still in its early days. Therefore, they are currently not analysed by the MSG MSCT.

## 6.2 NFC

### 6.2.1 Protocol

NFC (Near Field Communication) is a contactless protocol for mobile devices specified by the NFC Forum for multi-market usage and by EMVCo for mobile card payment applications. NFC Forum specifications (see [24]) are based on ISO/IEC 18092 [18] but have been extended for harmonisation with EMVCo and interoperability with ISO/IEC 14443 [22] infrastructures.

NFC is a radio frequency technology operating within the RF band of 13.56 MHz at rates ranging from 106 to 424 kbit/s. It operates at very short ranges of up to 4 cm (“proximity”) so that the user has to perform a voluntary gesture to initiate a communication between two devices by approaching them.

Each full NFC-enabled device can work in three modes:

- NFC card emulation: enabling the devices to act like smart cards (either using a Secure Element, or Host Card Emulation).
- NFC reader/writer: enabling the device to read information stored on NFC tags embedded in labels or smart posters. NFC tags are passive data stores which can be read, and under some circumstances written to, by an NFC device.
- NFC peer-to-peer: enabling two NFC-enabled devices to communicate with each other to exchange information in an ad-hoc fashion.

The NFC Data Exchange Format (NDEF) is a standardised data format maintained by the NFC Forum[68] that can be used to exchange information in reader/writer or peer-to-peer mode.

In the context of MSCT, if a mobile device OS only allows operation in NFC reader mode, the NFC technology could be utilised *uni-directionally* to read data from an NFC capable device (e.g., mobile phone) or to communicate data from the payee’s POI to the payer’s mobile device. If allowed by the mobile device OS, the NFC technology could be utilised for a *bi-directional exchange* of payer/payee identification and transaction data.

[68] https://nfc-forum.org/product/nfc-data-exchange-format-ndef-technical-specification/

### 6.2.2 Security of bi-directional NFC

The security of bi-directional NFC payments is based on the following:

- NFC is a proximity protocol, which can only be activated up to a distance of 4 cm between the payer and the payee devices, such as an NFC enabled mobile phone or a contactless POS terminal. Therefore the probability that the two payment transaction devices as well as a secondary "sniffing" acceptance device are present at the same place is quite unlikely, which reduces the risk of fraud. In addition, the NFC transaction is quite fast and normally does not take more than half a second, which also mitigates the risks.
- The security of the NFC transaction is normally ensured at the application level (e.g. EMV-based card payments, DESFire, MiFare, Calypso), not at the NFC protocol level. This actually means that protection of the sensitive data transmitted between the payer and the payee devices is protected and encrypted using keys and certificates specified by the respective payment services, as an example, implemented in the form of applications in the devices and such applications are tested and certified to withstand cryptographic attacks and ensure the required level of protection (see also Chapter 12 in the MSCT IG [10]).
- Following the specifications and requirements of payment services using NFC in their operations, the payer and the payee devices usually perform one-way (payer device by the payee device) authentication, but ideally would need to perform mutual authentication. This includes the verification of public key certificates and cryptograms (digital signatures), and exchange of data in an encrypted or tokenised form when necessary. Such data exchange is based on asymmetric or symmetric cryptography with the necessary key and public key certificate exchanges between the parties and the provisioning of keys and certificates into the payer and the payee devices before payments may occur at the POI.
- Normally, to protect keys and certificates stored in payer and payee devices, different techniques are used. One example would be using hardware secure storages also called secure elements, software secure storages such as those based on white box cryptography or a variety of hybrid technologies, based on combinations of software and hardware such as secure enclaves or trusted execution environments (see also Chapters 11 and 12 in the MSCT IG [10]).

### 6.2.3 Usage of bi-directional NFC for payments

Using the NFC bi-directional protocol might be challenging for the following reasons:

- Access to the NFC stack, which is needed to perform a transaction, is made proprietary on certain mobile phone platforms whereby terms and conditions for accessing the NFC antenna apply (see also Chapter 22 in the MSCT IG [10]). As a consequence, the NFC antenna cannot be activated by a third-party application such as MSCT, installed on such mobile device. In addition, on these mobile platforms, the use of the NFC antenna is bound to the use of the SE on these mobile devices, while the access to the SE is also proprietary. This basically eliminates the ability of

software-based applications to activate the NFC stack and antenna even without using the SE[69].

- The NFC interaction time between the two devices is very short and can only be performed if the payer and the payee devices are close to each other, so there is no technical possibility to establish a proper link between the two devices and send a substantial amount of data in both directions, which limits the value and capability of the NFC protocol in general – basically, only a few short messages can be exchanged in a fraction of a second while the payer device is activated by the electromagnetic field of the payee device.
- Mobile NFC devices deployed today are basically used as regular contactless physical cards as they do not have the capability to send more data, integrate value-added services such as loyalty, cannot automatically select a payment card in the wallet, which would be preferred by the consumer or merchant (e.g. there is no way to automatically select a merchant-ASPSP co-branded card, even if it is in the mobile wallet) and that is because of the limitation of the NFC and also payment protocols underneath.

### 6.2.4 Security of uni-directional NFC

The security of uni-directional NFC payments is different compared to bi-directional NFC:

- In uni-directional NFC there is no way to send data in both directions between the payer and the payee devices, hence it is not possible to establish a secure channel between the two devices.
- SInce it is a one-way communication, it is impossible to exchange certificates and verify digital signatures, hence it is impossible to perform mutual authentication at the POI.
- Only a limited amount of data can be transmitted, so it would probably be a reference, token or transaction ID with an URL of a server, so the actual transaction would happen at the back-ends between two servers (provided by MSCT service providers) rather than at the POI – it is possible to establish a proper secure channel between two servers and apply the appropriate secure protections.
- Alternatively, when payee-presented data is read by the payer device from the payee device, the payer device can connect directly to the MSCT service provider server and then the MSCT app in the mobile device can authenticate to the MSCT service provider server and vice-versa to establish a trusted channel after certificates have been verified to proceed with the payment transaction. In addition, in this scenario sensitive data can also be encrypted at the transport and application level through an exchange of session keys.
- An alternative option is that after the data is transmitted by the payee device (e.g. POI) to the payer's mobile device, a trusted secure channel is established between

[69] This has been addressed by the recently published Digital Market Act [6] and will therefore be subject to change.

the two devices over BLE, if supported by both devices (in analogy to the usage of a QR-code in the MSCT use cases described in Chapter 5).

### 6.2.5 Usage of uni-directional NFC for payments

Using uni-directional NFC reader /writer mode technology such as NFC NDEF tags, which can be generated by the payer and/or the payee devices in combination with servers, they are connected to, seems to be a good alternative, although they may also present some challenges, as follows:

- All contactless POS terminals, deployed in the market today, are configured to recognise the NFC card emulation mode only, which means that when an NFC enabled mobile device is presented, the POS terminal would enable the card emulation mode, instead of trying to generate an NFC tag to be read by the payer's mobile phone or to read an NFC tag presented by a payer's mobile phone. That said, POS terminals would need to be updated to be able to switch from the card emulation mode to the tag read/write mode, for example depending on which application is used: the cashier or payer would need to press the "MSCT payment service" button on the ECR/POI screen, so that the NFC card emulation mode is switched off and the NFC tag read/write mode is turned on.
- Most mobile phones in the market today, enabled with NFC, natively support the NFC card emulation mode, basically act as a card, even if the phone is turned off, so the NFC card emulation mode would be selected by default at the POI if the NFC phone is presented. In order to avoid collision, the NFC card emulation mode has to be turned off at the POS and also, an application such as MSCT has to be launched on the payer's mobile phone, because not all NFC enabled mobile devices can support NFC tag read/write capability without an application in the foreground (activated), which worsen the consumer experience. However these limitations in the current imposed capabilities should disappear over time to offer an equal level playing field amongst all mobile payments by allowing the MSCT app to be selected as the default payment app.

### 6.2.6 Additional challenges with NFC

Additional challenges that have been experienced in early pilots and limited trials for account-based payments using NFC, more in particular at the POI include:

- Some mobile payment providers have reported that it is difficult to convince POS terminal vendors to make changes and add the usage of the NFC protocol for account-based payments to the POI, next to the "standard" NFC protocols used for the (international) card schemes. This NFC protocol for account-based payments would require a dedicated specification and development. Moreover, some manufacturers are reluctant to use NFC for account-based payments in view of potential issues related to POS terminal type approval/certification for (mobile) card payments using NFC.
- Difficulties when the POS terminal turns on the NFC signal, other payment instruments than an MSCT might be triggered on the payer's mobile device (see also above).

- On some mobile phone platforms, the prioritisation of payment apps seem to be a problem (see above).

## 6.3 Bluetooth and Bluetooth Low Energy

### 6.3.1 Protocol

#### Bluetooth

Bluetooth is an industry standard according to IEEE 802.15.1 for bidirectional data transmission between devices over relatively short distances using radio technology. They may be operated worldwide without approval but robustness against interference (e.g., by WLANs or cordless telephones) needs to be implemented[70]. The actual achievable range depends not only on the transmission power but also on several further parameters such as for example, the sensitivity of a receiver and the designs of the transmitting and receiving antennas used by radio communication modules, or obstacles between transmitter and receiver. There are different range classes: Class 1 (max. 100 m), Class 2 (max. 10 m), Class 3 (max. 1 m).

#### Pairing

The establishment of a connection always takes place under the protocol architecture according to the specifically supported Bluetooth release version. A connection can originate from any Bluetooth enabled device. As soon as Bluetooth devices are put into operation, the individual Bluetooth controllers identify themselves within two seconds. Since this connection time for payment application at the POI is much too long, currently only the variant "Bluetooth Low Energy (BLE)" is applied in payment contexts.

#### Bluetooth Low Energy

Bluetooth Low Energy (BLE), is a radio technology with which devices in an environment up to about 10 meters can be networked. Compared to "classic" Bluetooth, BLE offers significantly shorter connection times. Based on the protocol Bluetooth version Low Energy V4.0 (and later) a “connectionless” (non-statically paired) operation can be established in only 3 ms and data transmission can be completed after 6 ms.

BLE transmissions can be made secure against unauthorised intrusion if they are operated as a connection with multi-level dynamic key allocation. Static key assignment limits security. When the key is transmitted, exactly this part of the communication is particularly at risk, since only the successful exchange of the key protects a BLE connection.

Unlike NFC, with radio ranges of typically < 10 cm, BLE has ranges of many meters, depending on its range class. This causes practical problems for use at the POIs, as several mobile devices can be in the reception range of the POI. As a consequence, an MSCT payment must be explicitly confirmed by the consumer on the mobile device once the connection has been successfully established.

In analogy to NFC technology (see above), the usage of the BLE technology for making proximity payments requires that the Bluetooth functionality on the consumer’s mobile device is switched on, which should be handled by the MSCT application. BLE is available on

[70]To achieve robustness against interference, frequency hopping is used, in which the frequency band is divided into 79 channels at 1 MHz intervals, which are changed up to 1600 times per second.

most mobile phones but the technology as such is challenging to secure proximity. Different phones have different characteristics so if no additional technology is used to secure the proximity for instance the wrong person might get the payment request. There are different technologies provided by different vendors to secure (which might involve patents) which creates a dependency on third party providers. It usually also requires some extra software to be integrated into the payment application.

### 6.3.2 Security of BLE

The security for payment operations based on BLE protocol should be based on the following principles:

- Since in case of BLE the distance between the payer and the payee device might be substantial, they need to perform a hand-shake with mutual authentication between the two devices before sensitive data can be exchanged.
- Subsequent to this mutual authentication, a secure channel should be established between the payer and the payee device, enabling to send sensitive data encrypted both at transport and application level.
- In analogy to NFC-based card payments, the security of BLE-based MSCTs, is to be based on asymmetric cryptography for the (mutual) authentication between the payer and the payee device, which requires the set-up and management of a PKI, including the validity of public key certificates. Likewise, the calculation of a cryptogram may be based on symmetric cryptography (between the payer device and the payee's PSP) or asymmetric cryptography.
- Sensitive data (e.g. cryptographic keys) have to be stored securely in a secure environment such as a SE, TEE or WBC.

### 6.3.3 Usage of BLE for payments

The following considerations should be taken into account when using BLE for payments:

- The access to the BLE stack and antenna is not restricted on most of the mobile devices, which makes them usable for payments.
- A proper secure communication link between the payer and the payee devices needs to be established based on cryptographic algorithms, requiring the generation and appropriate exchange of keys and public key certificates.
- BLE radio can be turned on and off on the mobile device, although since BLE is broadly used (e.g. for connecting headsets, cars or external speakers), it is normally forcedly turned on by the operating systems of mobile devices, which makes it almost always enabled.
- BLE requires the mobile devices to be powered, although the power consumption of the BLE radio is really insignificant and does not drain the battery.
- Most of the existing POS terminals in the market are not equipped with BLE technology, although there are technical ways to enable them, for example, by plugging in SIM cards with a BLE chip. New POS terminals are mainly equipped with BLE and also, other devices such as mobile phones and tablets can be used as an alternative to POS terminals – such devices all have BLE installed.
- Linking the payer and the payee devices in order to establish a secure channel and perform a payment transaction can be done by means of using a QR-code, an NFC

tag or any other method. This step would be needed to let the payer device uniquely identify which payee device is used to perform a payment transaction. If NFC is used, this linking is done by means of tapping the mobile phone to the contactless reader. However, the combination of the usage of NFC/QR-code and BLE might be confusing for the payer and have a negative impact on the payer experience.

- BLE is a bi-directional protocol, allowing to exchange a substantial amount of data – once the link is established it can last for a long time, which allows sending multiple digital signatures back and forth as well as other sensitive and value-added data without limitations that exist in bi-directional NFC (i.e. the payer device has to be in the field of the payee device and only for a short period of time).

### 6.3.4 Additional challenges with BLE

Additional challenges that have been experienced in early pilots and limited trials for account-based payments using BLE, some more in particular at the POI, include:

- The consumer needs to enable BLE - on some mobile phones this requires the activation of location services.
- Problems have been encoutered with signals of multiple cash registers in retail shops when using BLE, leading to cumbersome configurations.
- BLE technology on some mobile phones is more sensitive than others which implies that BLE parameter settings need to be defined for every mobile phone type.
- A separate device might need to be integrated on the merchant side - merchants are not used to integrate such devices.
- In some countries merchants moved away from usage of BLE.
- Problems of having MSCT app woken up by BLE – the payer first needs to open MSCT app, than choose QR-code or BLE.
- Beacon BLE technology seems to produce some errors.

Obviously the availability of BLE on mobile devices and POI is further evolving which as a result might have an influence on some of those challenges.

# 7 Minimum data sets for data exchange between the payer and the payee for MSCTs based on NFC

## 7.1 Introduction

This chapter is devoted to MSCTs whereby NFC is used as proximity technology for the data exchange between the payer and the payee to enable the initiation of an MSCT. In case NFC is used in a uni-directional way, the minimum data set for payee-presented data is discussed in section 7.2. For bi-directional NFC, the reader is referred to section 7.3.

## 7.2 Minimum data sets for uni-directional NFC

### 7.2.1 Introduction

This section considers the exchange of data (payee identification data and transaction data) using NFC by the payee (e.g. merchant POI or payee's mobile device) to an MSCT app on the payer's mobile device. For the purpose of this document, the following three cases with respect to the type of payee-presented data are considered:

- The payee-presented data includes a "(payee) token": in this case, a de-tokenisation process needs to take place such that all the data (payee identification and transaction data) can be derived from the token and provided to the payer via their MSCT service provider. This generally requires the support of the payee's MSCT service provider (in analogy to the Information Request/Response messages sent over the HUB for MSCTs based on QR-codes – see in section 17.5 in the MSCT IG [10]) prior to the initiation of the MSCT transaction.
- The payee-presented data contains a "proxy" for the payee identification data. In this case the data that is not in clear, but corresponds to the proxy, needs to be provided by the payee's MSCT service provider upon request from the payer's MSCT service provider in analogy to the Information Request/Response messages sent over the HUB for MSCTs based on QR-codes – see in section 17.5 in the MSCT IG [10]) prior to the initiation of the MSCT transaction.
- The payee-presented data includes all data in "clear" (e.g. the payee's name, trade name, IBAN of the payee's account, transaction amount, etc.). This enables the immediate initiation of the MSCT transaction.

Next to this data exchanges also an *identifier of the payee MSCT service provider* is needed for routing purposes by the HUB for the exchange of messages between the respective MSCT service providers.

Note also that in the last two cases described above, appropriate security measures need to be taken to ensure the integrity of the data and the confidentiality as appropriate (see Chapter 8).

### 7.2.2 Minimum data set for payee-presented data

The minimum data set to be exchanged between the payee and the payer, will rely on the MSCT transaction feature, as described above:

1. If the payee-presented data provided to the payer contains a (payee) token, the minimum data will consist of both routing info and the token as payload. The

minimum data will be forwarded in a Transaction Information Request message through the HUB from the payer's MSCT service provider to the payee's MSCT service provider for de-tokenisation into the transaction data (see Annex 1).

2. If the payee-presented data provided to the payer contains only part of the transaction data in clear (e.g., contains a proxy), the transaction data will need to be further completed by the payee's MSCT service provider. The minimum data set will consist of both routing info and the available transaction data (e.g. the proxy). The minimum data will be forwarded in a Transaction Information Request message through the HUB from the payer's MSCT service provider to the payee's MSCT service provider for completion of the transaction data.
3. If the payee-presented data provided to the payer contains all transaction data "in clear", the minimum data set will consist of both routing info and all necessary payload data.

Therefore the minimum data sets for the payee-presented data, covering the three cases described above are as follows:

| Data transmitted by payee device to payer's mobile device with uni-directional NFC |
|---|
| **Payee-presented data includes a token:**<br><br>[Version]+[Type]+ [Payee MSCT Service Provider ID] + [(payee) token] |
| **Payee-presented data contains a proxy for the payee:**<br><br>[Version]+[Type]+ [Payee MSCT Service Provider ID] + [proxy] + [a clear-text name/value string] |
| **Payee-presented data includes all transaction data "in clear":**<br><br>[Version]+[Type]+ [Payee MSCT Service Provider ID] + [a clear-text name/value string] |

Table 15: Minimum data set exchanged by the payee's device to the payer's device for MSCTs based on uni-directional NFC with payee-presented data

**Version**

A version number shall support further updates to the set of data elements.

/1/ refers to the first version.

**Type**

The type indicates what kind of payment context is expected.

The following coding shall be applied:

- /m/ mobile payment at the POI
- /e/ e-commerce (and m-commerce) payment
- /i/ invoice payment
- /p/ person-to-person payment
- /w/ opening a URL in a webview (e.g. virtual POI).

**MSCT service provider ID**

An identifier needs to be assigned to every MSCT service provider for routing purposes. This will require an eligibility checking and registration of the MSCT service provider under a "to be defined" MSCT Interoperability Framework.

This MSCT Interoperability Framework should also responsible for the issuance of the MSCT service provider ID.

The coding of the MSCT service provider ID shall be 3 characters alphanumeric (an).

## 7.3 Minimum data sets for bi-directional NFC for C2B and B2B payments

### 7.3.1 Introduction

This section considers the exchange of data (payer and payee identification data and transaction data) using NFC between the payee (e.g. merchant POI or MSCT app on payee's mobile device) and an MSCT app on the payer's mobile device.

This section will make a distinction whether a single or double tap is performed by the payer's mobile device to the payee's device (e.g., POI or mobile device).

### 7.3.2 Minimum data sets for data exchanged between payer and payee

| **Data transmitted by payee device to payer's mobile device if SCA without dynamic linking is performed** | | **Coding** |
|---|---|---|
| Name payee (account holder) (O) | | 1 to 70 an |
| Trade name merchant (O for C2B and B2B) | | 1 to 35 an |
| Name of payee reference party (O) | | 1 to 70 an |
| Trade name of payee reference party (O) | | 1 to 35 an |
| IBAN/Proxy/Alias of payee account (M) | | 1 to 34 an |
| MCC (M for C2B and B2B) | Merchant Category Code | 4 n |

| | | |
|---|---|---|
| Type of payment instrument (M) | SCT or SCT Inst | 3 to 4 an |
| Purpose of (instant) credit transfer (includes e.g. merchant transaction identifier) (O) | Data for reconciliation purposes at payee (e.g., merchant) – is included from initiation through entire transaction payment chain | 1 to 4 an |
| Currency (M) | | 1 to 3 an |
| Transaction amount (M) | | 1 to 12 n |
| Challenge generated by payee device (M) | | 10 n |

**Table 16: Minimum data set exchanged by the payee's device to the payer's device for MSCTs based on bi-directional NFC with single tap with SCA without dynamic linking**

| **Data transmitted by payee device to payer's mobile device if SCA with linking is performed** | | **Coding** |
|---|---|---|
| Name payee (account holder) (M) | | 1 to 70 an |
| Trade name merchant (M for C2B and B2B) | | 1 to 35 an |
| Name of payee reference party (O) | | 1 to 70 an |
| Trade name of payee reference party (O) | | 1 to 35 an |
| IBAN/Proxy/Alias of payee account (M) | | 1 to 34 an |
| MCC (M for C2B and B2B) | Merchant Category Code | 4 n |
| Type of payment instrument (M) | SCT or SCT Inst | 3 to 4 an |
| Purpose of (instant) credit transfer (includes e.g. merchant transaction identifier) (O) | Data for reconciliation purposes at payee (e.g., merchant) – is included from initiation through entire transaction payment chain | 1 to 4 an |
| Currency (M) | | 1 to 3 an |

| | | |
|---|---|---|
| Transaction amount (M) | | 1 to 12 n |
| Challenge generated by payee device (M) | | 10 n |

**Table 17: Minimum data set exchanged by the payee's device to the payer's device for MSCTs based on bi-directional NFC with single tap with SCA with dynamic linking**

Note that all the data in Table 16 and Table 17 is to be used for the calculation of the cryptogram by the MSCT app on the payer's mobile phone.

| Data transmitted by payer's mobile device to payee device | | Coding |
|---|---|---|
| Payer MSCT service provider ID (M) | | 1 to 70 an |
| Payer token[71] (M) | | 1 to 70 an |
| Cryptogram (M) | | 8 bytes |
| User verification result (M) | | 1n |
| Remittance information structured or Remittance information unstructured (O) | Information supplied by the payer in the SCT Inst/ SCT Instruction and transmitted to the payee in order to facilitate the payment reconciliation | 1 to 35 an |

**Table 18: Minimum data set exchanged by the payer's mobile device to the payee device for MSCTs based on bi-directional NFC with single tap**

Note that all these data, except the remittance information is to be used for the calculation of the cryptogram by the MSCT app on the payer's mobile phone.

[71] See also Chapter 8.

# 8 Security aspects of data exchanged using NFC or BLE

The data exchanged between the payer's mobile device and the payee's device (e.g., POI, mobile phoned) may contain both sensitive and non-sensitive payment data that can be used by different entities involved in the processing of the MSCT transaction.

In principle, this data may be static, e.g., payee account data and related payment details for a fixed transaction amount (typical use case may be an NFC tag on a product) or may be dynamic to initiate/identify a single specific MSCT transaction (e.g., at a POI).

Tampering the data exchanged may lead to fraudulent transactions or data leakage. Therefore the sensitive payment data exchanged should be adequately protected while also the integrity of the data elements exchanged should be ensured to avoid any service disruptions. Obviously the integrity of this data, as appropriate, shall be checked before any transaction information is displayed to the payer on their mobile device.

Below a more detailed analysis is made for each of the two technologies used for MSCTs. Note also that additional security measures have been specified in the MSCT IG [10].

**Payee-presented data**

Proxies or data that are present "in clear" in the exchanged data need an integrity protection to avoid manipulations with the intention to initiate fraudulent transactions (e.g., to a fake payee or with a wrong transaction amount).

Based on Art. 4(32) of PSD2 [2], the IBAN is not considered to be sensitive payment data and can therefore be included in clear-text in payee-presented data for the initiation of a transaction e.g. at the payee's POI. However, since its disclosure may be used to carry out fraud, it will be for PSPs to assess the risks arising from transmitting the IBAN in clear-text between the payee's infrastructure (e.g, POI, mobile device) and the payer's mobile device. Subsequently, PSPs should decide whether it is necessary to implement corresponding security measures to mitigate these risks[72].

It should further be noted that in certain countries (e.g., France, Sweden, …), there are recommendations to protect the IBAN outside the inter-PSP space. This means that in some countries it is recommended that the IBAN is not included "in clear" into the payee-presented data.

In view of the considerations made above, if possible, the usage of a dynamic token to represent the payee identification and transaction data, in particular for C2B or B2B payments, is recommended.

In addition, to protect the data exchanged, the MSCT application on the payer's/payee's mobile device or MSCT software on the POI must enforce a properly encrypted and authenticated connection to the payer's/payee's MSCT service provider (as already specified in the MSCT IG - Chapter 9, [10]).

[72] See also the EBA answer to question EBA Q&A 2020_5477.

### Payer-presented data

If Customer IDs, IBANs and proxies would be present "in clear" in payer-presented data, they would need integrity protection to avoid mistakes with the initiation of transactions (e.g. using the wrong payer).

Moreover, the CustomerID might be a payer credential (e.g. for access to the online banking system). The capture of the CustomerID and IBAN could lead to impersonation attacks and initiation of fraudulent transactions (see for example [11], [20]) and reputational damage while also contaminating other payment instruments such as SDD. In view of the EBA answer to EBA Q&A 5476 that states *"the Customer ID cannot be included in a clear-text in a payer-presented QR-code for the initiation of credit transfers at the point of interaction without any security measures (e.g. encryption, tokenisation, transport layer security) ensuring its confidentiality during the QR-code life-cycle*", and the further clarification given on the generation of the QR-codes in EBA Q&A 2021_6298, the MSG MSCT concluded that CustomerID in "clear" is not allowed in the payer-presented data, independently of the proximity technology used.[73]

Based on Art. 4(32) of PSD2 [2], the IBAN is not considered to be sensitive payment data and can therefore be included in clear-text in payer-presented data for the initiation of a transaction at the payee's POI. However, since its disclosure may be used to carry out fraud, it will be for PSPs to assess the risks arising from transmitting the IBAN in clear-text free text between the payee's POI and the payer's mobile device. Subsequently, PSPs should decide whether it is necessary to implement corresponding security measures to mitigate these risks[74].

It should further be noted that in certain countries (e.g., France, Sweden, …), there are recommendations to protect the IBAN outside the inter-PSP space. This means that in some countries it is recommended that the IBAN is not included "in clear" into the payer-presented data.

If the payer-presented data is static (e.g., a static token) the same risk as described above applies, namely it could lead to impersonation attacks and initiation of fraudulent transactions (see for example [11], [20]) and reputational damage.

In view of the considerations made above, the usage of a dynamic token (i.e. that can only be used once) to represent the payer identification data, in particular for C2B payments is recommended.

In addition, to protect the data exchanged, the MSCT application on the payer's/payee's mobile device or MSCT software on the POI must enforce a properly encrypted and

[73] ETPPA tabled a dissenting opinion on the impact of the EBA answer. In their view the EBA answer does not allow the removal of these options, because a) any non-PSP – including payers themselves – should still be allowed to provide the CustomerID in clear-text, b) PIS@POS could not work without, because PSD2 APIs require the CustomerID in clear-text as well, and c) tokenisation can never be mandated, because the introduction of a tokeniser brings an unnecessary gatekeeper into the process, which adds cost, complexity and competition issues.

[74] See also the EBA answer to question EBA Q&A 2020_5477.

authenticated connection to the payer's/payee's MSCT service provider (as already specified in the MSCT IG - Chapter 9, [10]).

Note that for both modes, appropriate security measures should be applied by the entity/application creating the data to be exchanged (see also Chapters 11 and 12 in the MSCT IG [10]).

# 9 Towards standardisation of MSCTs based on NFC

In this Chapter, only some topics towards the standardisation of MSCTs based on NFC will be identified, subject to sufficient market interest to pursue this technology for account-based mobile payments. This is due to the current many unknowns for MSCTs based on BLE, the work on eIDAS2.0, including the European Digital Wallet, that is still under development and the current lack of usage of this technogy for account-based payments by mobile payment service providers in the market today.

One of the major issues that has hindered the market take-up of NFC for mobile account-based payments was the difficulties encountered with the usage of NFC on some mobile platforms as described in section 6.2. It is expected that with the implementation of the newly published Digital Market Act [6], some of these obstacles will disappear over time.

The next issue is the constraints regarding SCA which are put on MSCTs by the PSD 2 [2] and the RTS [3] and are directly impacting the payer and the payee experience. In addition, those may also have a negative impact on the transaction time, more in particular for retail payments. As a result, these regulations seem to create an unlevel playing field for MSCTs compared to the usage of other mobile payment instruments.[75]

Next, also a further analysis of all aspects related to the co-existence of NFC-based MSCTs with contactless card payments (based on NFC) is needed. This should be covered by the European Cards Stakeholders Group (ECSG) in view of their scope extension. Those aspects include payment instrument selection, PCI certification of POIs in view of the addition of a new NFC-based payment instrument, etc.

Although this document identifies in Chapter 7 the minimum data elements to be exchanged between the payer and the payee to enable the initiation of an MSCT based on uni- or bi-directional NFC, further standardisation work is needed to specify in detail the NFC messages exchanged between the payer and the payee as well as the specification of the cryptograms used between the payer's app and the payer's MSCT service provider / ASPSP.

Finally, the issuance of MSCT service provider IDs for routing purposes to achieve interoperability of MSCTs is needed. In analogy to what has been stated in the document on S*tandardisation of QR-codes for MSCTs* (see EPC024-22, [14] ), the eligibility check of MSCT service providers should become part of the overall governance of an "Interoperability Framework for MSCTs". The latter would also involve the establishment of a directory for the registration of the MSCT service providers and the issuance of MSCT service provider IDs.

It is finally to be noted that this document discussed some of the technical standardisation requirements for MSCTs based on NFC or BLE, in addition to the technical interoperability

[75] See for instance the EBA answers to EBA Q&A 2020_5247 and 2020_5367.

aspects already specified in the MSCT IG [10]. Obviously further work would be needed on the governance aspects under a potential future “Interoperability Framework for MSCTs”.

# 10 Conclusions

This document describes MSCT use cases based on NFC and BLE for the data exchange between the payer's mobile device and the payee device (POI, mobile device, etc.) to enable the initiation of a SEPA (instant) credit transfer. It is to be noted that most of the MSCT use cases employing BLE that are described in this document are based on the work on eIDAS2.0 which is currently still under development.

The document further analyses some technical aspects of these proximity technologies, their usage for payments, the related security aspects and the challenges detected.

It further specifies the minimum data sets to be exchanged between the payer's mobile device and the payee's device when NFC is used either in uni-directional or bi-directional mode.

The document also contains a dedicated chapter on security aspects related to the data exchanged using NFC or BLE to initiate MSCTs.

The document has a dedicated chapter that highlights some topics to be addressed towards a further standardisation of MSCTs based on NFC if there is sufficient market interest to take up this type of MSCTs. The MSG MSCT refrained for the time being from further analysing MSCTs based on BLE in view of the current many unknowns for MSCTs based on BLE, the work on eIDAS2.0, including the European Digital Wallet that is still under development and the current lack of usage of this technology for account-based payments by mobile payment service providers in the market today.

One of the major issues that has hindered the market take-up of NFC for mobile account-based payments was the difficulties encountered with the usage of NFC on some mobile platforms as described in section 6.2. It is expected that with the implementation of the newly published Digital Market Act [6], some of these obstacles will disappear over time.

Also a further analysis of all aspects related to the co-existence of NFC-based MSCTs with contactless card payments (based on NFC) is needed. This should be covered by the European Cards Stakeholders Group (ECSG) in view of their scope extension.

Note that it is proposed that the future governance aspects related to interoperability of MSCTs based on NFC should become part of the overall Governance of an "Interoperability Framework for MSCTs". The latter also involves the eligibility check of MSCT service providers, the establishment of a directory for the registration of the MSCT service providers and the issuance of MSCT service provider identifiers.

In order to help developing a successful MSCT ecosystem that provides value for all, it is very important to gather industry opinion and market feedback regarding this document. Therefore a 10-week public consultation will be held before the final version of the document will be developed.

# Annex 1: A short introduction to eIDAS2.0 including EUDIW

## A1.1 Introduction

This annex provides clarifications on the terminology used in Chapter 5 related to the eIDAS2.0 and the European Digital Identity Wallet (EUDIW), based on *eSignature FAQ*

**Electronic Signatures**

The eIDAS Regulation defines three levels of electronic signature: 'simple' electronic signature, Advanced Electronic Signature (AdES) and Qualified Electronic Signature (QES), authenticated by a natural person.

**'Simple' electronic signature**

An electronic signature is defined as "data in electronic form which is attached to or logically associated with other data in electronic form and which is used by the signatory to sign". Thus, something as simple as writing your name under an e-mail might constitute an electronic signature.

**Advanced electronic signature (AdES)**

An advanced electronic signature is an electronic signature which is additionally:

- uniquely linked to and capable of identifying the signatory;
- created in a way that allows the signatory to retain control;
- linked to the message in a way that any subsequent change of the data is detectable.

The most commonly used technology able to provide these requirements relies on the use of asymmetric cryptography (sometimes also referred to as public-key cryptography), involving a private-public key pair and a public-key infrastructure (PKI), which includes the use of public key certificates.

**Qualified electronic signature (QES)**

A qualified electronic signature is an advanced electronic signature which is additionally:

- created by a qualified signature creation device (QSCD);
- and is based on a qualified certificate for electronic signatures.

## A1.2 Electronic SEALs

Like the electronic signature, the eIDAS Regulation defines three levels of electronic seal: 'simple' electronic seal, advanced electronic seal and qualified electronic seal. Electronic seals are usually automatically generated by a trusted electronic device and are linked to a legal entity.

**'Simple' electronic seal**

An electronic seal is defined as "data in electronic form, which is attached to or logically associated with other data in electronic form to ensure the latter’s origin and integrity".

**Advanced electronic seal (AdES)**

An advanced electronic seal is an electronic seal which is additionally:

- uniquely linked to the creator of the seal;
- capable of identifying the creator of the seal;
- created using electronic seal creation data that the creator of the seal can, with a high level of confidence under its control, use for electronic seal creation; and
- linked to the data to which it relates in such a way that any subsequent change in the data is detectable.

**Qualified electronic seal (QES)**

Similar to a qualified electronic signature, a qualified electronic seal is an advanced electronic seal which is additionally:

- created by a qualified seal creation device (QSCD);
- and is based on a qualified certificate for electronic seals.

### A1.3 Certificates

**Identity or Public key Certificate**

In cryptography, a public key certificate is sometimes also referred to as a digital certificate or identity certificate.

A public key certificate contains a copy of the public key that corresponds with a cryptographic private key. It also contains information about the characteristics and ownership of the public key. The certificate contents are electronically signed by the issuer of the certificate, the so-called Certification Authority, and the signature is added to the certificate. If necessary, the issuer can subsequently revoke the certificate. This will invalidate any associated attribute certificates (see below).

**Attribute Certificate**

A digital certificate that binds a set of descriptive data items directly to an Identity Certificate. An Attribute Certificate does not contain a public key.

For each Identity Certificate held by a subject (e.g. a person), there may be multiple associated attribute certificates, each of which contains discrete information about the subject. This allows for selective disclosure.

Similar to an Identity Certificate, an Attribute Certificate is electronically signed by the issuer of the certificate and the signature is added to the certificate. If necessary, the issuer can subsequently revoke the certificate.

Examples of attribute certificates are:

- A person’s name

- A person's email address
- A person's IBAN.

### A1.4 Types of Identity Certificate

**Identity Certificate for Strong User Authentication/Strong Customer Authentication**

This certificate is associated to a private key that is under the direct control of the subject who is a natural person. Two Factor Authentication (2FA) techniques are employed. For example, the private key (possession element) may be unlocked with a PIN (knowledge element) or a biometric fingerprint (inherence element).

These signatures covey non-repudiation. The overall goal of the non-repudiation is to be able to prove that a particular signed message (e.g., a payment) is undeniably associated with a particular entity.

Only these non-repudiation signatures can convey content commitment types of ProofOfApproval (i.e., personal approval) and ProofOfOrigin (i.e., the combination of creation plus personal approval).

**Identity Certificate for Identification and Authentication**

This certificate is associated to a private key that is NOT under the direct control of the entity. This certificate can be used for mutual authentication of two devices.

These signatures can convey ProofOfCreation, ProofOfSender and ProofOfReceipt. They cannot convey personal approval.

### A1.5 European Digital Identity Wallet (EUDIW)

The European Digital Identity Wallet which is currently being developed by the European Commission will be available to EU citizens, residents, and businesses who want to identify themselves or provide confirmation of certain personal information. The aim is that it can be used for both online and offline public and private services across the EU. It is expected that the EUDIW will based on eIDAS2.0.

Every EU citizen and resident in the Union should be able to use a personal digital wallet.

### A1.6 Additional information

More information on eIDAS2.0 may be found through the following link:

https://ec.europa.eu/transparency/expert-groups-register/core/api/front/document/73759/download.

More information on cryptography may be found in EPC342-08 [8].

More information on the EUDIW may be found through the following link:

https://ec.europa.eu/info/strategy/priorities-2019-2024/europe-fit-digital-age/european-digital-identity_en

## Annex 2: List of participants to MSG MSCT Plenary

The following organisations have contributed to the development of this document through their participation in the Plenary of the multi-stakeholder group Mobile initiated SEPA (instant) Credit Transfers (MSG MSCT).

| |
|---|
| AIB on behalf of Banking & Payments Federation Ireland (BPFI) - representing European Payments Council (EPC) |
| BEUC - European Consumer Organisation |
| BlueCode |
| BP |
| Bridge - representing European Third Party Providers Association (ETPPA) |
| Carrefour - representing EuroCommerce |
| Circle K |
| Colruyt - representing EuroCommerce |
| Crédit Agricole - representing European Payments Council (EPC) |
| Crédit Mutuel - representing European Payments Council (EPC) |
| DnB Bank – representing European Payments Council (EPC) |
| EACT - European Association of Corporate Treasurers |
| Estonian Banking Association- representing European Payments Council (EPC) |
| EMPSA - European Mobile Payment Systems Association |
| Fiserv |
| Getswish |
| Huawei |
| Idemia - representing Smart Payment Association |
| IKEA - representing EuroCommerce |
| Intesa Sanpaolo on behalf of Italian Banking Association (ABI) – representing EPC |
| La Banque Postale - representing European Payments Council (EPC) |
| Mastercard |
| Millennium bcp – representing European Payments Council (EPC) |
| Monei |
| National Clearing House KIR |
| Nexi Payments |
| nexo |

| |
|---|
| OpenWay |
| Orange - representing GSMA |
| Payconiq |
| PPRO - representing European Third Party Providers Association (ETPPA) |
| Rabobank - representing European Payments Council (EPC) |
| TAS Group |
| Thales – representing Smart Payment Association |
| Tink – representing European Third Party Providers Association (ETPPA) |
| Vipps |
| Visa |
| W3C |
| Eurosystem – as observer |
| European Central Bank (ECB) – as observer |
| European Commission – as observer |

Table 19: The MSG MSCT Plenary

## Annex 3: List of participants MSG MSCT Work-Stream on interoperability of MSCTs based on NFC or BLE

The following organisations have contributed to the development of this document through their participation in Work-Stream interoperability of MSCTs based on NFC or BLE of the multi-stakeholder group Mobile initiated SEPA (instant) Credit Transfers (MSG MSCT).

| |
|---|
| Apple – representing European Payment Institutions Federation (EPIF) |
| AvatarPay |
| Capsys Informatics Ltd |
| Colruyt – representing EuroCommerce |
| Crédit Agricole - representing European Payments Council (EPC) |
| Crédit Mutuel - representing European Payments Council (EPC) |
| DnB Bank – representing European Payments Council (EPC) |
| EMVCo |
| EPI – European Payments Initiative |
| Fortress Mobile Ltd |
| Idemia - representing Smart Payment Association |
| IKEA - representing EuroCommerce |
| ING - representing European Payments Council (EPC) |
| Mastercard |
| Monei |
| nexo |
| OpenWay |
| PPRO - representing European Third Party Providers Association (ETPPA) |
| Redsys |
| Quali-Sign |
| SIBS on behalf of Millennium bcp – representing European Payments Council (EPC) |
| Thales – representing Smart Payment Association |
| Tink – representing European Third Party Providers Association (ETPPA) |

Table 20: The MSG MSCT Work-Stream Interoperability of MSCTs based on NFC or BLE

The multi-stakeholder group wishes to inform that this document is provided "as is" without warranty of any kind, whether expressed or implied, including, but not limited to, the warranties of merchantability and fitness for a particular purpose. Any warranty of non-infringement is expressly disclaimed. Any use of this document shall be made entirely at the

user's own risk, and neither the multi-stakeholder group nor any of its members shall have any liability whatsoever to any implementer for any damages of any nature whatsoever, directly or indirectly, arising from the use of this document, nor shall the multi-stakeholder group or any of its members have any responsibility for identifying any IPR.