D1.1

Report on use case and system architecture requirements

Project number:	610436
Project acronym:	MATTHEW
Project title:	MATTHEW: Multi-entity-security using active Transmission Technology for improved Handling of Exportable security credentials Without privacy restrictions
Start date of the project:	1st November, 2013
Duration:	36 months
Programme:	FP7-ICT-2013-10

Deliverable type:	Report
Deliverable reference number:	ICT-610436 / D1.1 / 1.0
Work package contributing to the deliverable:	WP 1
Due date:	Mar 2014 – M05
Actual submission date:	1st April, 2014

Responsible organisation:	GTO
Editor:	GTO – Christian DIETRICH
Dissemination level:	Public
Revision:	1.0

Abstract:	This document defines the requirements of the MATTHEW system architecture to demonstrate three different use cases: Mobile payment, Access Control and Advanced Ticketing. These applications will allow the evaluation of the technologies developed and evaluated within the MATTHEW project: active contactless, antenna designs, multiple secure elements, PUFs and privacy requirement.
Keywords:	MATTHEW, applications, use cases, scenarios, requirements

Editor

Dietrich Christian (GTO)

Contributors (ordered according to beneficiary numbers)

Bock Holger, Buchsbaum Martin (IFAT)

Capomaggio Gregory (GTO)

Erich Wenger (IAIK)

Martin Deutschmann (TEC)

Pavel Kristof (IMA)

Paillier Pascal (CRX)

Copyrights

Disclaimer

The research leading to these results has received funding from the European Union’s Seventh Framework Programme (FP7-ICT-2013) under grant agreement n° 610436.

Executive Summary

This first deliverable of the MATTHEW project forms the foundations of the project and gives the directions for the work addressed by the partners in the following Work Packages.

The partners identified three application domains applicable to the MATTHEW framework, and detailed the requirements to be able to implement them in demonstrators at the end of the project. To cover the short and long term scope of the project the applications are split in two categories.

The first category contains the two use case: Mobile Payment and Access Control, these are existing contactless applications that can be transferred on the MATTHEW platform to enhance the user experience. They will be used to evaluate and validate the hardware technologies developed in the MATTHEW project. The low impact of their transfer on the existing infrastructure will allow the adoption of the active contactless and antennas design technologies in the project, and stimulate the industrial exploitation.

The second category consists in the use case Advance Ticketing, a research oriented application, which will let the partners investigate the introduction of the Multi Entity, the PUF (Physically Unclonable Function) technology, ABC credential and group signatures technologies for the secure and anonymous issuance, transfer of transport tickets in the MATTHEW system architecture. The first results of these investigations, lead in WP2, are reflected in the use case requirements detailed in this document.

After the definition of the MATTHEW System Architecture components the partners will work on the MATTHEW Platform Specification, continue the work in the WP2 – Multi Entity Security, start the work in WP3 – Hardware Component development, and initiate the first discussion for the WP4 – Application development.

Contents

List of Figures

Chapter 1 Introduction

This document summarises the results of the two initial tasks from the MATTHEW project. The purpose of these tasks was to set the bases of the project, clarifying the target applications if possible from the use case scenario up to the protocols, actors and MATTHEW system architecture requirements. This started in task 1.1 by focusing on the use case requirements, and has been finalised for three domains with the Mobile Payment, the Access Control and the Advanced Ticketing applications.

For the two first application domains, the environment exists, thus the MATTHEW project objective is to enhance the user experience thanks to the active contactless integration in small hardware platforms. The partners clarified the way this integration can be performed and completed by describing a detailed diagram flow between the involved actors. These scenarios will support the validation of the contactless technology improved within the project. Further these small hardware platforms will be integrated in the mobile phone Multi-Entity environment and allow an evaluation and characterisation of the technologies. As these two applications focus on the improvement of the contactless experience, they have no impact on the existing infrastructures. This approach will allow fast adoption and rapid introduction of the technologies developed and evaluated in the MATTHEW project.

The third application, Advanced Ticketing, will implement Multi-Entity scenarios and for that integrate technologies still in the research domain, further investigated in WP2, such as PUF, anonymity, ABC Credentials or group signatures. The involved actors and use case have been described, and the associated requirements identified.

In task 1.2 these results have been used to define the requirements for the elements of the MATTHEW System Architecture. The hardware and software constraints for the Contactless card/ token, the Mobile host, the terminals and the backend required for setting up the final demonstrators for the two applications Mobile Payment and Access Control are detailed.

Chapter 2 Use Case Requirements

The MATTHEW use cases – chosen to illustrate the flexibility and adaptability of the MATTHEW approach – will demonstrate the applicability of the MATTHEW architecture and concepts to different areas of use.

The MATTHEW project will define a platform architecture [ref. to D1.2] for upcoming and future generations of mobile devices with multiple secure entities involved. Further we will create ways to transfer some of those entities or security relevant credentials stored on such entities taking privacy protection into consideration.

Instances of this platform will be extracted with the respective feature sets fulfilling use case oriented requirements.

There are two types of use cases represented within MATTHEW. One type being a set of use cases close to industrial exploitation (advanced mobile payment and access control), the other type being more research oriented and still further away (estimated 5-8 years) from market exploitation (privacy enhanced transferable public transport credentials).

Nevertheless the MATTHEW platform aims to incorporate in an integrative way all use case oriented requirements collected from both types of use cases.

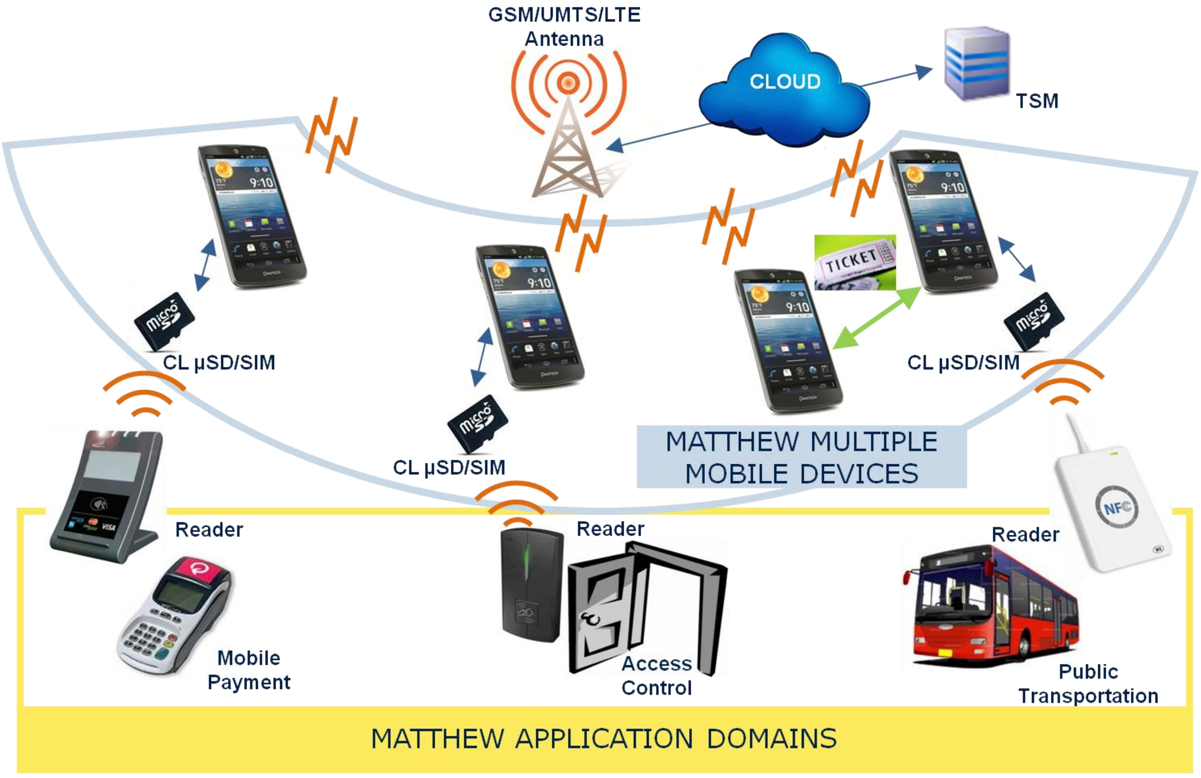


Figure 1: MATTHEW Application framework overview

In addition MATTHEW will take care on prerequisites for highly reliable implementation of the NFC (Near Field Communication)-based communication by driving active transmission technology to a new level of radio transmission quality.

2.1 Requirements from Use Case 1 – Advanced mobile banking application

2.1.1 Introduction

A contactless payment system uses a wireless device such as a card or token instead of a contact chip card or magnetic stripe card. A payment is contactless when you do not swipe or insert your credit or debit card at the checkout. Instead, you just hold your card up to ten centimeters away from the payment reader at the register and your payment information is sent wirelessly and processed.

Contactless devices offer a faster and more convenient alternative to cash for low-value purchases at fast food restaurants and convenience stores. They are also ideal for remote or unattended payment situations, such as vending machines, road tolls or parking meters.

Recent years have seen the emergence of NFC technology in different kinds of portable equipments such as smartphones or tablets, providing new functionalities to these objects, especially contactless payment capabilities. In such topology, the payment application must be installed into a dedicated secure IC (Integrated Circuit) which is not issued by the bank. This secure IC could be a NFC SIM (Subscriber Identity Module) card delivered by a mobile operator or an embedded secure element controlled by the handset manufacturer. In this case, banks have to sign agreements and sometimes pay for fees to these third parties to install and manage their payment application. Furthermore, they usually have to deal with many actors to cover a wide range of customers. Actually this weakness of the business model is a big obstacle to the development of the NFC mobile payment.

In this context, the contactless micro SD (Secure Digital) card proposed in the MATTHEW project appears as a real suitable solution for banks, or other payment service providers, that can freely issue and manage their own standalone payment device to their customers without having to deal any agreements with third parties.

2.1.2 Mobile payment actors

The Figure 2 represents a synoptic of the different equipments involved in a Mobile Contactless Payment (MCP) solution based on a contactless micro SD card.

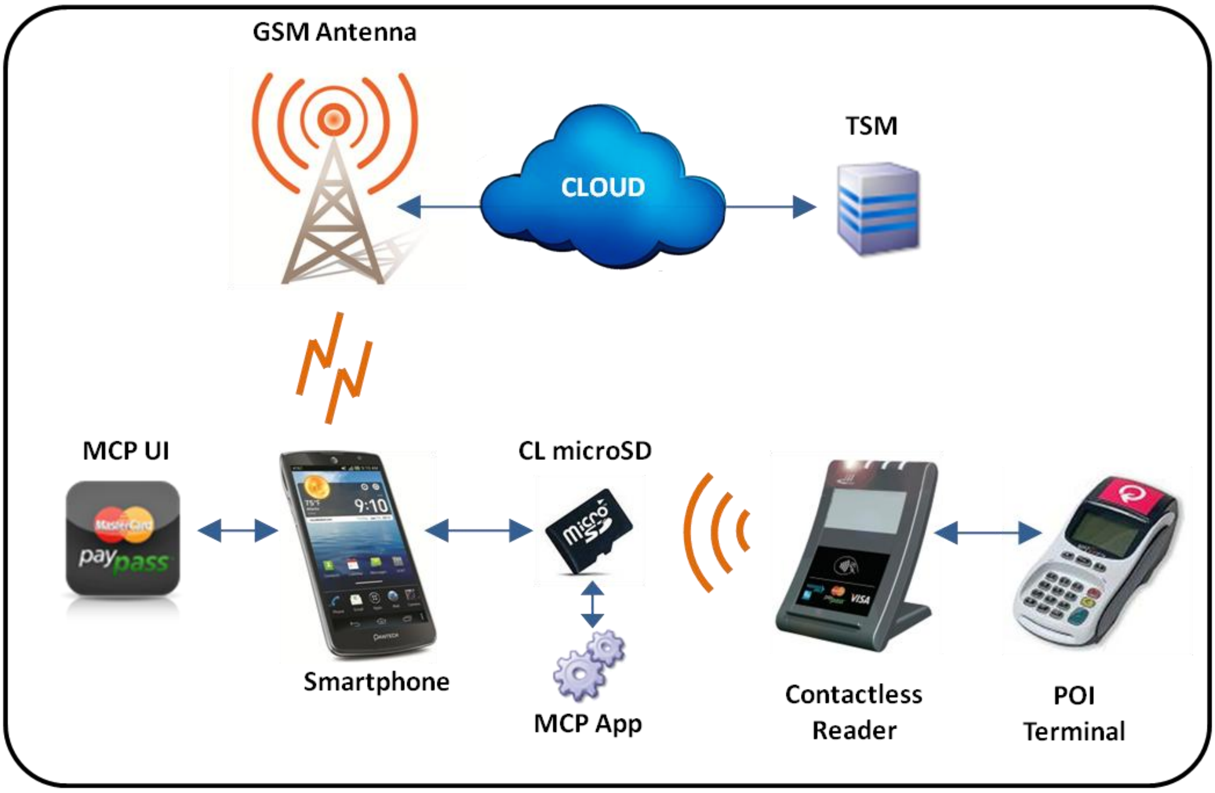


Figure 2: Mobile contactless payment infrastructure overview

In such business models, a bank or a payment service provider (Paypal, Google Wallet, etc...) can propose a contactless micro SD card to their customers which can be inserted in their mobile phones in case they are compatible. This new form factor will be usually issued in addition to ordinary payment cards.

The customer will have to download the dedicated mobile contactless payment user interface application (MCP UI) available in an online store (Play store, Apple Store, etc...). Optionally this application installer could be pre-loaded by the issuer in the Flash memory of the micro SD card. The customer is then able to initiate a contactless payment through this user interface, to revise the history of performed transactions or to access online services provided by the issuer.

Thanks to the “over the air” connection provided through the mobile phone, the bank or the payment service provider can manage remotely the secure MCP application embedded into the micro SD card. The Trusted Service Manager (TSM) works behind the scenes to make the entire management of your payment account onto your cell phone efficient and secure. A TSM knows both banking and mobile phone security and systems, bridging multiple banks and operators while ensuring that consumer credit card information is completely secure.

At the point of sale (POS) the retailer would calculate the amount owed by the customer and provide options for the customer to make payment. In a mobile payment infrastructure, a new generation of POI (Point of Interaction) terminals will be designed such that contactless devices (e.g. NFC phones or stickers) can be used for payments.

2.1.3 Mobile payment scenarios

The introduction of the micro SD card in the payment infrastructure impacts the part of the transactions related to the end user interaction with the MCP UI and the POI terminal connected to the system back office. As a consequence only the “off line” scenarios are described in this section. In this section the two main scenarios encountered in the mobile contactless payment context will be described.

The Mobile Contactless Payment application embedded in the micro SD card is able to select one or the other scenario according to the transactions history or the transaction data. For example, the MCP application will require a Cardholder Verification Method (double tap scenario) if the number of transactions exceeded the counter limit or if the amount is above the threshold specified by the card issuer.

2.1.3.1 Off-line single tap transaction scenario

Single tap is the basic payment flow where no Cardholder Verification Method (CVM) is requested to finalize the transaction. This payment method is typically intended for low value payments or when entering PIN is not convenient such as tolls.

Step 0: Pre Requisite

- The customer has previously inserted a contactless micro SD card into the memory extension slot of his/her mobile phone.
- The customer opens his/her mobile MCP UI before starting the transaction. Optionally, the customer can enter a code/password to authenticate to the MCP UI. By default, the contactless interface of the micro SD card is disabled to avoid involuntary or fraudulent payment transactions without the knowledge of the end user.
- The merchant enters the transaction amount on the POI terminal.

Step 1: Payment Request

- The transaction amount is displayed on the merchant’s POI terminal.
- The POI terminal requests for a card payment.
- The POI terminal enables the contactless terminal to allow a contactless transaction.

Step 2: Enable payment

- To perform the contactless payment, the customer clicks on the “Pay” button of the MCP UI. This action enables the contactless interface on the micro SD card.
- At this stage the MCP UI is waiting for an “end of transaction” (EOT) event returned by the memory card. As SD interface doesn’t provide any interrupt capabilities, the MCP UI has to regularly question the memory card (polling mode).

Step 3a: Payment aborted

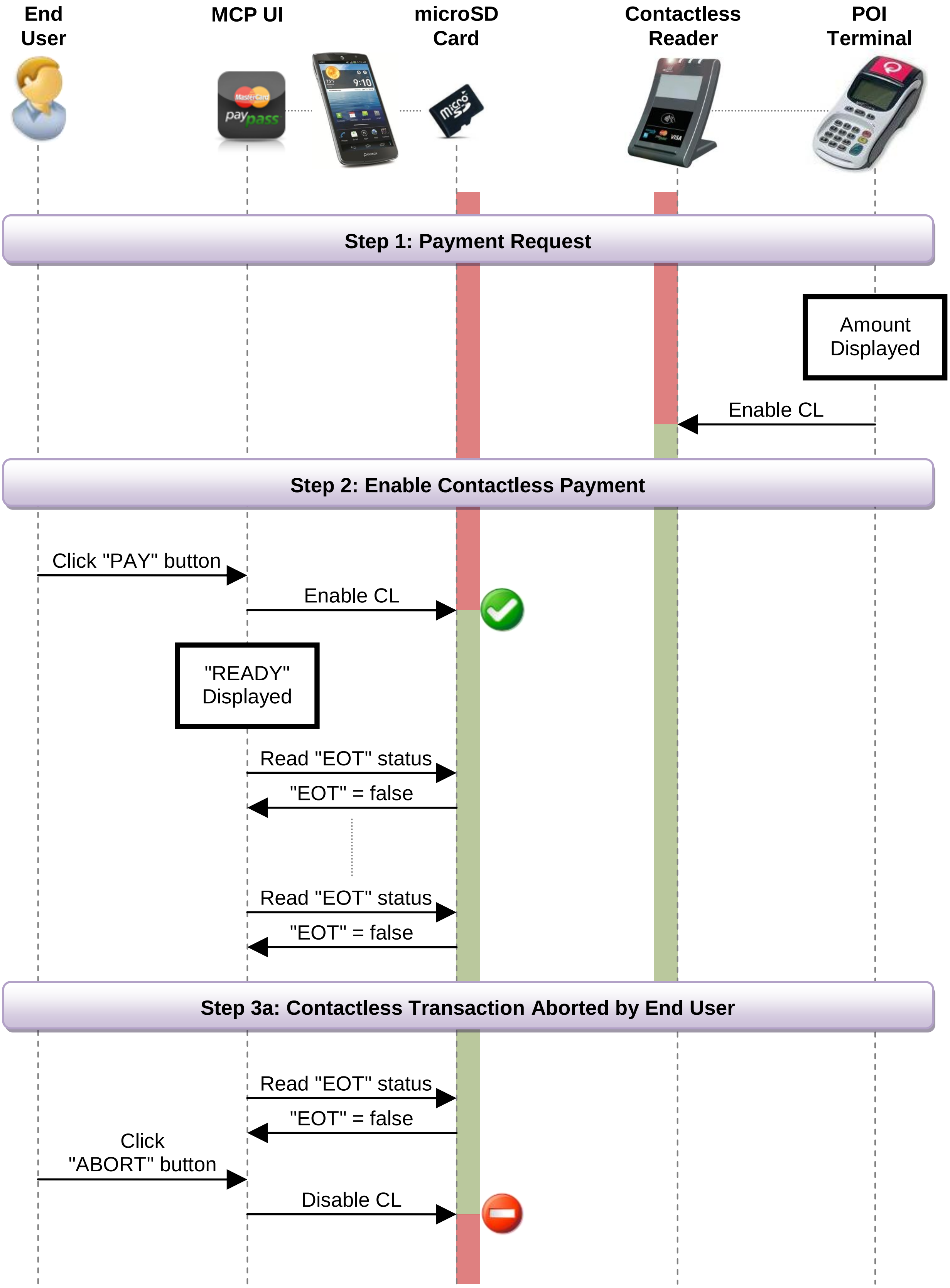
- The customer can cancel at any time the payment through the MCP UI by clicking on the "Abort" button.
- This action disables the contactless interface on the micro SD card.

Step 3b: Payment timeout

- If no "end of transaction" event occurs during a specified time, the payment operation is aborted, the mobile application disable the contactless interface on the micro SD card, and a timeout message is displayed on the screen.

Step 3c: Contactless transaction performed

- The customer taps his/her mobile phone on the contactless reader area. (The customer holds his/her mobile phone close to the contactless reader area until an audible tone and/or a visible signal takes place).
- The POI selects the contactless technology.
- The POI checks the available applications available into the contactless micro SD card and selects the appropriate MCP application through the PPSE.
- The contactless terminal automatically retrieves the MCP application configuration.
- The contactless reader transmits all transaction information to the merchant's POI terminal.
- An audible tone and/or visible signal then indicate that the mobile phone – contactless reader interaction is completed. After this, the mobile phone can be removed from the contactless reader area. Note, however, that the transaction processing at the POI might still continue.
- An off-line MCP application authentication/authorisation is performed by the POI.
- After processing the off-line authorisation, the merchant's POI terminal displays an approval or decline.
- Information about the current transaction (e.g. transaction amount) is saved in the MCP application log file.
- At the same time an "end of transaction" notification is returned by the micro SD card to the MCP UI. The MCP UI disables the contactless interface on the micro SD card then read the MCP application log file to display on the mobile phone information about the current transaction.


End User
MCP UI
microSD Card
Contactless Reader
POI Terminal
Step 1: Payment Request
Amount Displayed
Enable CL
Step 2: Enable Contactless Payment
Click "PAY" button
Enable CL
"READY" Displayed
Read "EOT" status
"EOT" = false
Read "EOT" status
"EOT" = false
Step 3a: Contactless Transaction Aborted by End User
Read "EOT" status
"EOT" = false
Click "ABORT" button
Disable CL

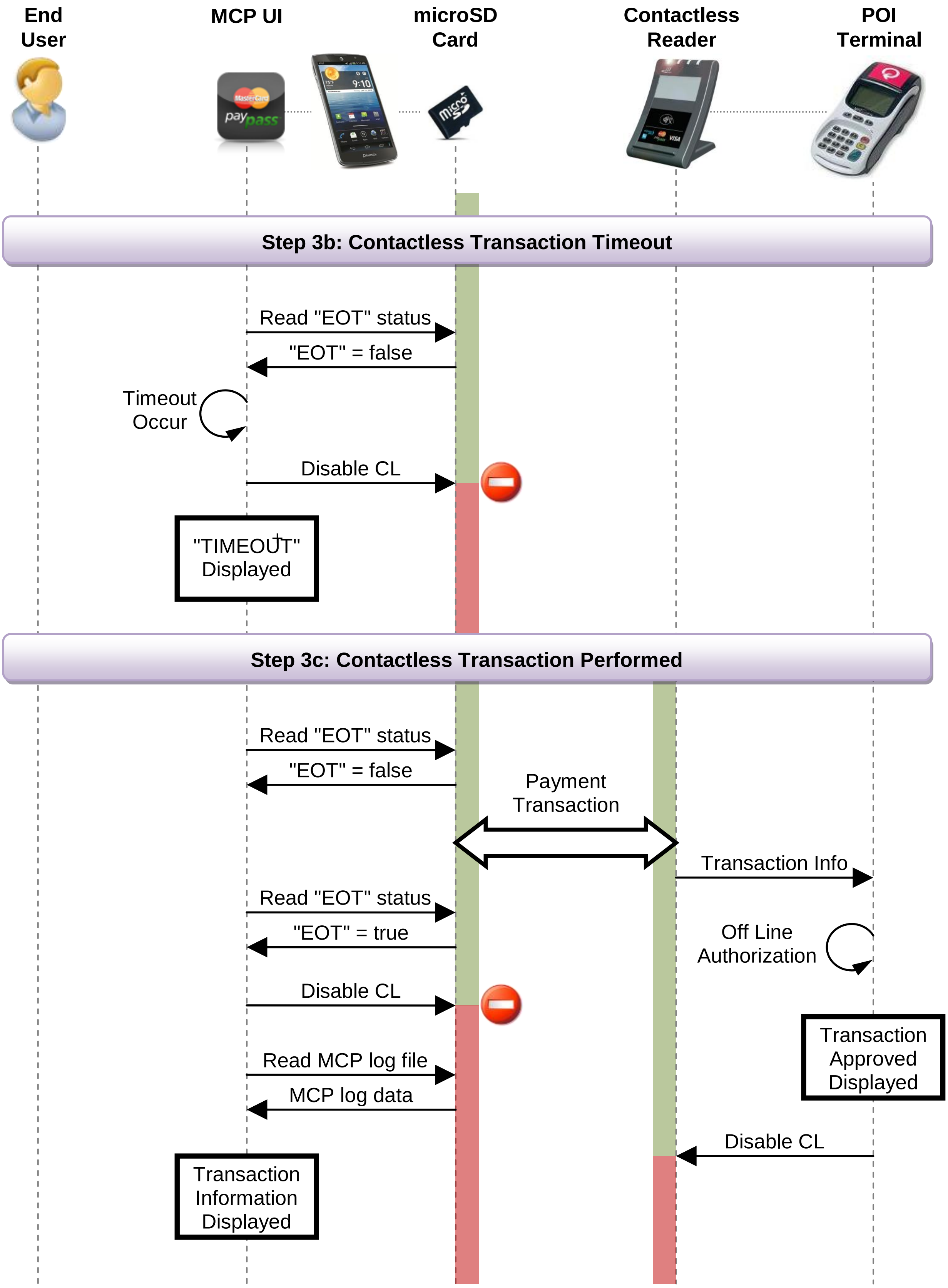


Figure 3: Off-line single tap transaction scenario flow

2.1.3.2 Offline double tap transaction scenario

Double tap payment flow is used when a Cardholder Verification Method (CVM) is requested to finalize the transaction. In this scenario the customer has to enter a PIN code on his handset at the end of the first tap. This payment method is typically intended for high value payments or if the customer has exceeded a certain amount of transactions.

Step 0: Pre Requisite

- The customer has previously inserted a contactless micro SD card into the memory extension slot of his/her mobile phone.
- The customer opens his/her mobile MCP UI before starting the transaction. Optionally, the customer can enter a code/password to authenticate to the MCP UI. By default, the contactless interface of the micro SD card is disabled to avoid involuntary or fraudulent payment transactions without the knowledge of the end user.
- The merchant enters the transaction amount on the POI terminal.

Step 1: Payment Request

- The transaction amount is displayed on the merchant's POI terminal.
- The POI terminal requests for a card payment.
- The POI terminal enables the contactless terminal to allow a contactless transaction.

Step 2: Enable payment

- To perform the contactless payment, the customer clicks on the "Pay" button of the MCP UI. This action enables the contactless interface on the micro SD card.
- At this stage the MCP UI is waiting for an "end of transaction" (EOT) event returned by the memory card. As SD interface doesn't provide any interrupt capabilities, the MCP UI has to regularly question the memory card (polling mode).

Step 3a: Payment aborted

- The customer can cancel at any time the payment through the MCP UI by clicking on the "Abort" button.
- This action disables the contactless interface on the micro SD card.

Step 3b: Payment timeout

- If no "end of transaction" event occurs during a specified time, the payment operation is aborted, the mobile application disable the contactless interface on the micro SD card, and a timeout message is displayed on the screen.

Step 3c: 1st Contactless transaction performed

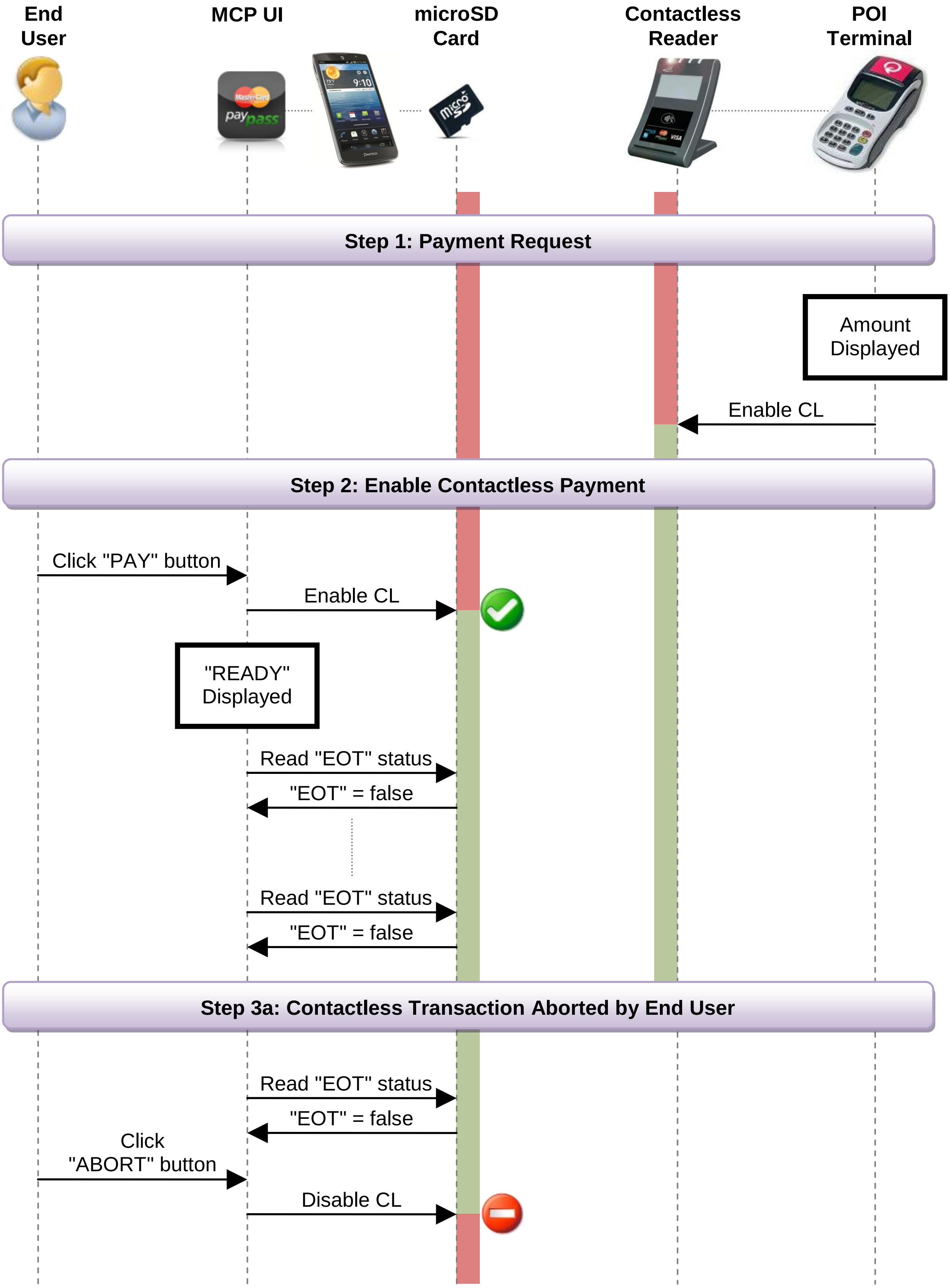
- The customer taps (1st tap) his/her mobile phone on the contactless reader area. (The customer holds his/her mobile phone close to the contactless reader area until an audible tone and/or a visible signal takes place).
- The POI selects the contactless technology.
- The POI checks the available applications available into the contactless micro SD card and selects the appropriate MCP application through the PPSE.
- The contactless terminal automatically retrieves the MCP application configuration including the CVM list (off-line CVM in this scenario).
- The contactless reader transmits all transaction information to the merchant’s POI terminal.
- A specific audible tone and/or visible signal indicate that “half-course” transaction is completed and the terminal asks the customer to enter his/her mobile code to complete the contactless payment transaction.
- At the same time an “end of transaction” notification is returned by the micro SD card to the MCP UI. The MCP UI disables the contactless interface on the micro SD card then read the MCP application log file.
- According to the last logged information, the MCP UI displays on the mobile phone the amount of the ongoing transaction and asks the customer to enter his mobile code.

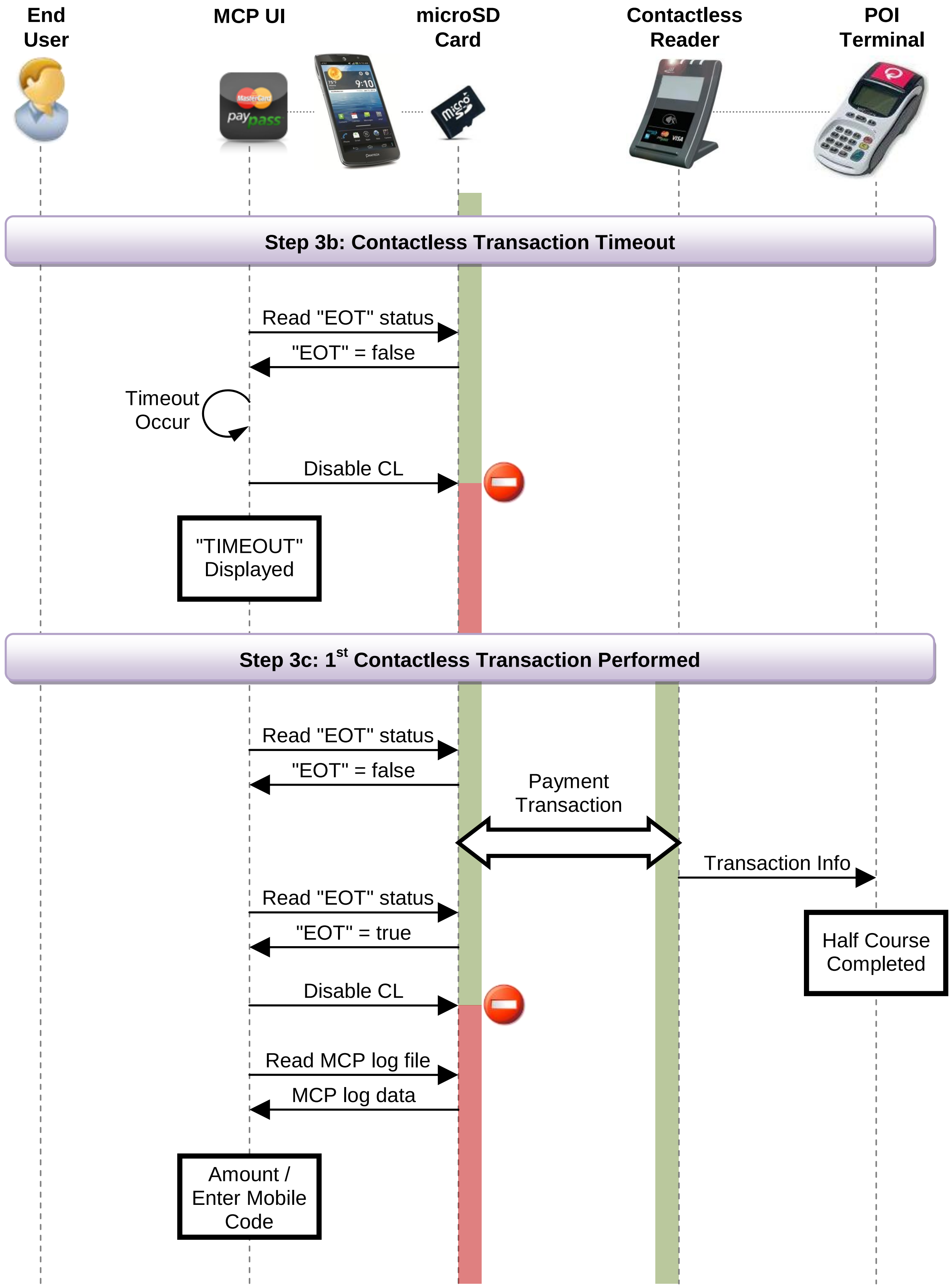
Step 4: Customer’s mobile code verification

- The customer checks the purchase amount and enters his/her mobile code on the mobile phone.
- The MCP UI transmits a mobile code verification request to the micro SD card.
- Upon successful verification of the mobile code, a message is displayed on the mobile phone requiring the customer to tap again his/her mobile phone on the contactless reader area.

Step 5: 2nd Contactless transaction performed

- The customer taps again (2nd Tap) his/her mobile phone on the contactless reader area.
- An audible tone and/or visible signal then indicate that the mobile phone – contactless reader interaction is completed. After this the mobile phone can be removed from the contactless reader area. Note, however, that the transaction processing at the POI might still continue.
- An off-line MCP application authentication/authorisation is performed by the POI.
- After processing the off-line authorisation, the merchant’s POI terminal displays an approval or decline message.
- Information about the current transaction (e.g. transaction amount) is saved in the MCP application log file.
- At the same time an “end of transaction” notification is returned by the micro SD card to the MCP UI. The MCP UI disables the contactless interface on the micro SD card then read the MCP application log file to display on the mobile phone information about the current transaction.


End User
MCP UI
microSD Card
Contactless Reader
POI Terminal
Step 1: Payment Request
Amount Displayed
Enable CL
Step 2: Enable Contactless Payment
Click "PAY" button
Enable CL
"READY" Displayed
Read "EOT" status
"EOT" = false
Read "EOT" status
"EOT" = false
Step 3a: Contactless Transaction Aborted by End User
Read "EOT" status
"EOT" = false
Click "ABORT" button
Disable CL


End User
MCP UI
microSD Card
Contactless Reader
POI Terminal
Step 3b: Contactless Transaction Timeout
Read "EOT" status
"EOT" = false
Timeout Occur
Disable CL
"TIMEOUT" Displayed
Step 3c: 1st Contactless Transaction Performed
Read "EOT" status
"EOT" = false
Payment Transaction
Transaction Info
Half Course Completed
Read "EOT" status
"EOT" = true
Disable CL
Read MCP log file
MCP log data
Amount / Enter Mobile Code

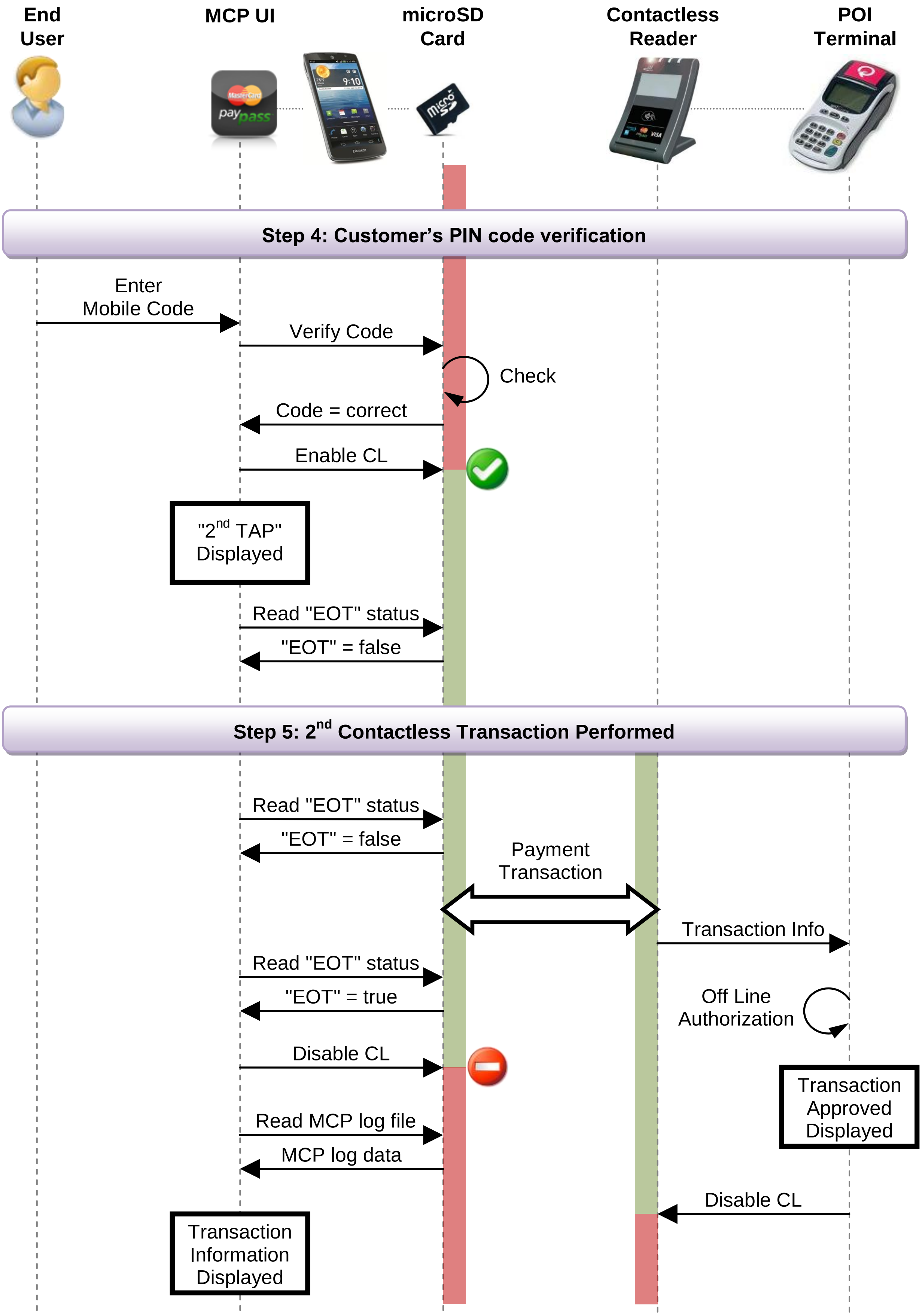


Figure 4: Off-line double tap transaction scenario flow

2.2 Requirements from Use Case 2 – Advanced access control application

2.2.1 Introduction

The access control and time attendance systems are usually using contactless cards for the identification of persons. For less secure systems we can use contactless cards on the frequency 125 kHz either without security property or with very small security. For example we can use the EM4002 or HID PROX largely deployed contactless cards. For applications where customers require a higher level of security we can use cards on the frequency 13,56 MHz with ciphered communication. These cards meet ISO standard 14443 or 15693, like MIFARE®Classic or MIFARE® DESFire®. Since MIFARE Classic cards are widespread all over the world, these cards were subject to a number of malicious attacks and the security of this card has been broken a few years ago. MIFARE DESFire cards are still resisting to attacks. Cards with the JCOP operating system provide the highest level of security, but these cards were too expensive for access control and time attendance systems. These cards were only used for banking or payment applications.

A few years ago the first NFC mobile phones appeared on the market. These mobile phones were equipped with NFC security modules based on the JCOP card. This allowed a higher level of security without additional cost of a card as it was provided by the mobile manufacturer. However this solution implicates two major drawbacks. The first issue of this solution was that the security element was soldiered inside the mobile phone with no possibility to transfer this element to another phone. The second problem was that the access right to the security element is limited to the mobile phone manufacturer and mobile network operator, who have no real interest to stimulate the integration of this mobile phone in access control solutions. The second generation of NFC mobile phones use as security element a NFC SIM card. The SIM card is transferable from one mobile phone to another as long it is equipped with the same NFC interface. The problem of access rights to the secure element application repository remains, as the SIM is controlled and managed by the mobile network operator. The mobile network operator can sell or rent space on secure element for the service provider applications. But the operator does this with large scale organisations such as bank institutions, to install and personalise their application; it is hardly doable for every organisation access control solutions as there is not a common overall system.

The secure element which will be present in the micro SD card, delivered by a third party, perfectly matches the secure identifier application requirements for very secure access control systems. The big advantage of this solution is that the space for secure identifier applications can be shared by more than one access control system. The secure identifier applications can be delivered to the secure element "over the air" by the trusted service manager. Another advantage is that the secure element can cooperate with mobile phone applications for special usage as described in the following use cases.

2.2.2 *Secure access control actors*

The Figure 5 represents the equipments required for an access control system based on a contactless technology.

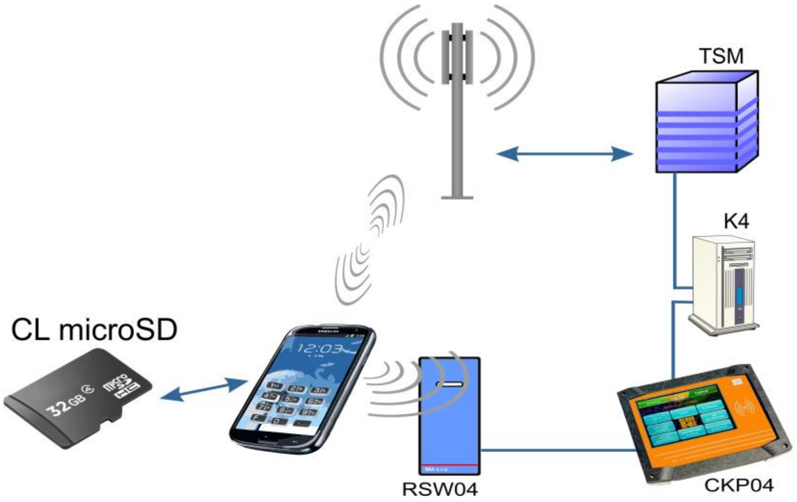


Figure 5: Contactless access control system infrastructure overview

The MATTHEW secure identifier access control application is present on the contactless micro SD inserted in the mobile phone.

The user presents this mobile phone in front of the contactless reader (RSW04) which will perform the complete card reading scenario. Once this is completed, the reader sends, via the Wiegand interface, the card identification number to the access control terminal (CKP04). The Wiegand interface arose from the popularity of Wiegand effect card readers largely deployed in the 1980s; it is used to connect a card reader to an access control system.

The access control terminal (CKP04) contains the list of enabled cards authorised to access the protected secure area, therefore the terminal is connected to the lock or the door handle.

The access control server (K4) centralises and generates the list of enabled cards for every terminal (CKP04). The access control server also generates the NFC card numbers which are send via the Trusted Service Manager (TSM) to MATTHEW contactless micro SD card.

2.2.3 *Types configuration of micro SD cards*

Three types of NFC micro SD card will be used in the following use case scenarios:

- *Simple identification NFC micro SD card*: Only identification data are present in this type of card. When the card is correctly authenticated, identification data will be send to the access control terminal.
- *Chained NFC micro SD card*: This type of cards contain the identification data, a flag indicating *Chained card, and a* list of maximum 4 chained contactless cards. Two cards of this type have to be presented consecutively to the contactless reader to get the access granted the access control terminal to the second one.
- *PIN NFC micro SD card*: Present in this type of card are the identification data and flag, indicating that the PIN must be presented to enable the sending of identification data to the access control terminal. The PIN will be present in additional data of the application.

2.2.4 Access control scenarios

This section describes the three main scenarios which demonstrate the Secure Access Control System.

2.2.4.1 Simple identification card reading scenario

- The customer taps his/her mobile phone on the RSW04 contactless reader area. (The customer holds his/her mobile phone close to the contactless reader area until an audible tone and/or a visible signal takes place).
- After detection of the NFC micro SD in the RF field, the special application for the MATTHEW project is selected.
- If this AID exists in the NFC micro SD, the authentication process is started. If not, this micro SD is deselected and the reader checks if another card is in the RF field.
- After successful selection of the application, the standard authentication process SCP02 protocol is started. The INIT UPDATE command is send to the NFC micro SD.
- After the reception of the response from the NFC micro SD to the INIT UPDATE command, the cryptograms is computed, and send to the NFC micro SD with the EXTERNAL AUTHENTICATE command.
- After successful reception of the EXTERNAL AUTHENTICATE command response, the authentication is ended.
- The special command READ DATA is send to the NFC micro SD. This command will be implemented as a special java applet.
- After the successful reception of the READ DATA command response, the identification data is extracted, and the flag indicating the type of card is checked. If the flag indicates *Simple identification card*, the received identification data is send by the contactless reader (RSW04) to the access control terminal (CKP04) via the Wiegand interface.

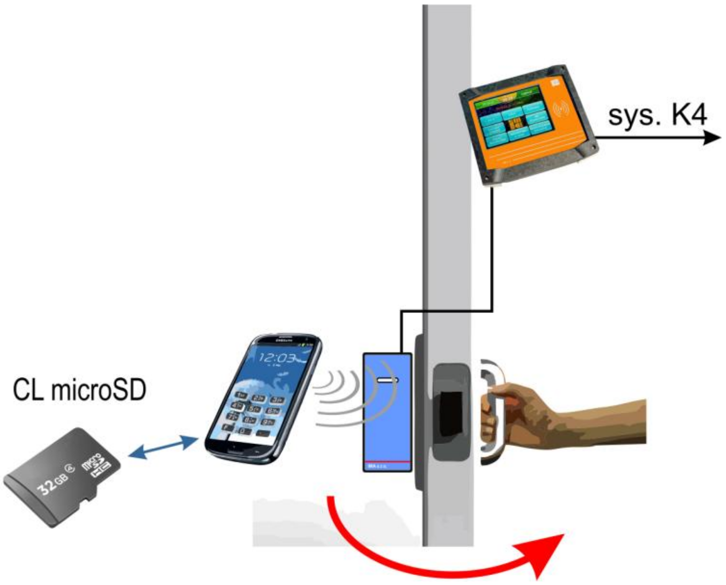


Figure 6: Simple identification card reading scenario

2.2.4.2 Chained card reading scenario

- The customer taps his/her mobile phone on the RSW04 contactless reader area. (The customer holds his/her mobile phone close to the contactless reader area until an audible tone and/or a visible signal takes place).
- After detection of the NFC micro SD, the special application identifier (AID) for the MATTHEW project is selected.
- If this AID exists in the NFC micro SD, the authentication process is started. If not, this micro SD is deselected and the reader checks if another card is in the RF field.
- After the successful application selection, the authentication process is started. The INIT UPDATE command is send to NFC micro SD card.
- After receiving a response from the NFC micro SD card to the INIT UPDATE command, a cryptogram will be calculated and the result will be send to NFC micro SD card by the EXTERNAL AUTHENTICATE command.
- After successful reception of the EXTERNAL AUTHENTICATE command response, the authentication is ended.
- The special command READ DATA is send to the NFC micro SD. This command will be implemented as a java applet.
- After the successful reception of the READ DATA command response, the identification data is extracted, and the flag indicating the type of card is checked. If the flag indicates *Chained card,* the list of chained contactless identifiers is saved temporarily to the contactless reader (RSW04).
- The contactless reader (RSW04) starts waiting for the taping of another NFC card within this list. This state is indicated on the reader by the fast blinking of a green LED for 20 seconds.
- If in this interval a mobile phone with NFC micro SD card is presented, the identification procedure is repeated.
- If the flag indicates the type of card *Chained card,* the contactless identification data of this second card is checked against the list of identifiers previously read.
- If the data matches, the identification data read from second NFC micro SD card is send to the access control terminal via the Wiegand interface of the reader.
- In all other cases the reader returns to the state waiting for card presentation.

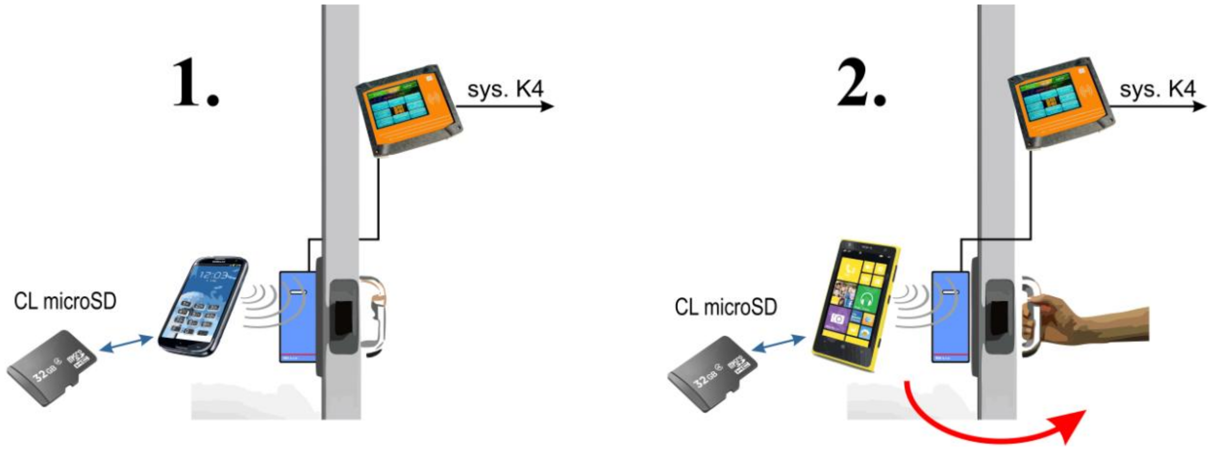


Figure 7: Chained card reading scenario

2.2.4.3 PIN card reading scenario

- The customer taps his/her mobile phone on the RSW04 contactless reader area. (The customer holds his/her mobile phone close to the contactless reader area until an audible tone and/or a visible signal takes place).
- After detection of the NFC micro SD in the RF field, the special application for the MATTHEW project is selected.
- If this AID exists in the NFC micro SD, the authentication process is started. If not, this micro SD is deselected and the reader checks if another card is in the RF field.
- After successful selection of the application, the standard authentication process is started. The INIT UPDATE command is send to the card.
- After the reception of the response from the NFC micro SD to the INIT UPDATE command, the cryptograms is computed and send to the NFC micro SD with the EXTERNAL AUTHENTICATE command.
- After successful reception of the EXTERNAL AUTHENTICATE command response, the authentication is completed.
- The special command READ DATA is send to the NFC micro SD. This command will be implemented as a java applet.
- After the successful reception of the READ DATA command response, the identification data is extracted, and the flag indicating the type of card is checked. If the flag indicates *PIN card,* the PIN is extracted from the NFC micro SD and temporarily stored in the contactless reader.
- The reader switches to the CARD EMULATION mode. In this mode the reader behaves like a standard contactless card with special NFC Data Exchange Format (NDEF) tag. The reader will wait for the connection with the mobile phone.
- The NFC interface of the micro SD is disabled and the mobile phone switches to reader mode.
- As the mobile phone is presented to the contactless reader, it reads a special NDEF; the mobile phone sends “WAIT” commands to the reader and starts the application associated to the NDEF, which will request the user to enter a PIN.
- Once the mobile phone is removed from RF field, the reader will indicate to the user that is in waiting state for PIN entry with a fast blinking of a green LED for 20s.
- The user enters the PIN on the application of the mobile phone and taps the mobile phone again on the contactless reader.
- Phone will read again NDEF from the reader, and the authentication between the Reader and the application in the mobile started.
- After a successful authentication, the PIN entered is transferred from the mobile phone to the contactless reader.
- If both PIN match in the contactless reader, the identification data is send to the Access control terminal via the Wiegand interface.
- In all other cases the reader returns to the state waiting for card presentation.

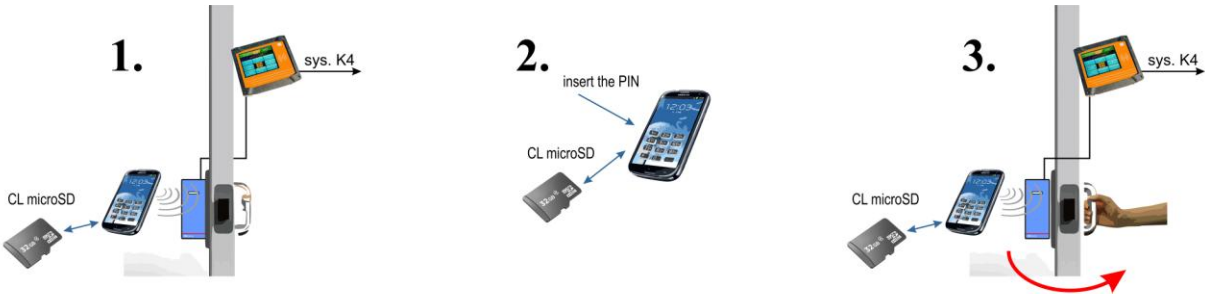


Figure 8: PIN card reading scenario

2.2.5 *Communication scheme for simple identification card reading scenario*

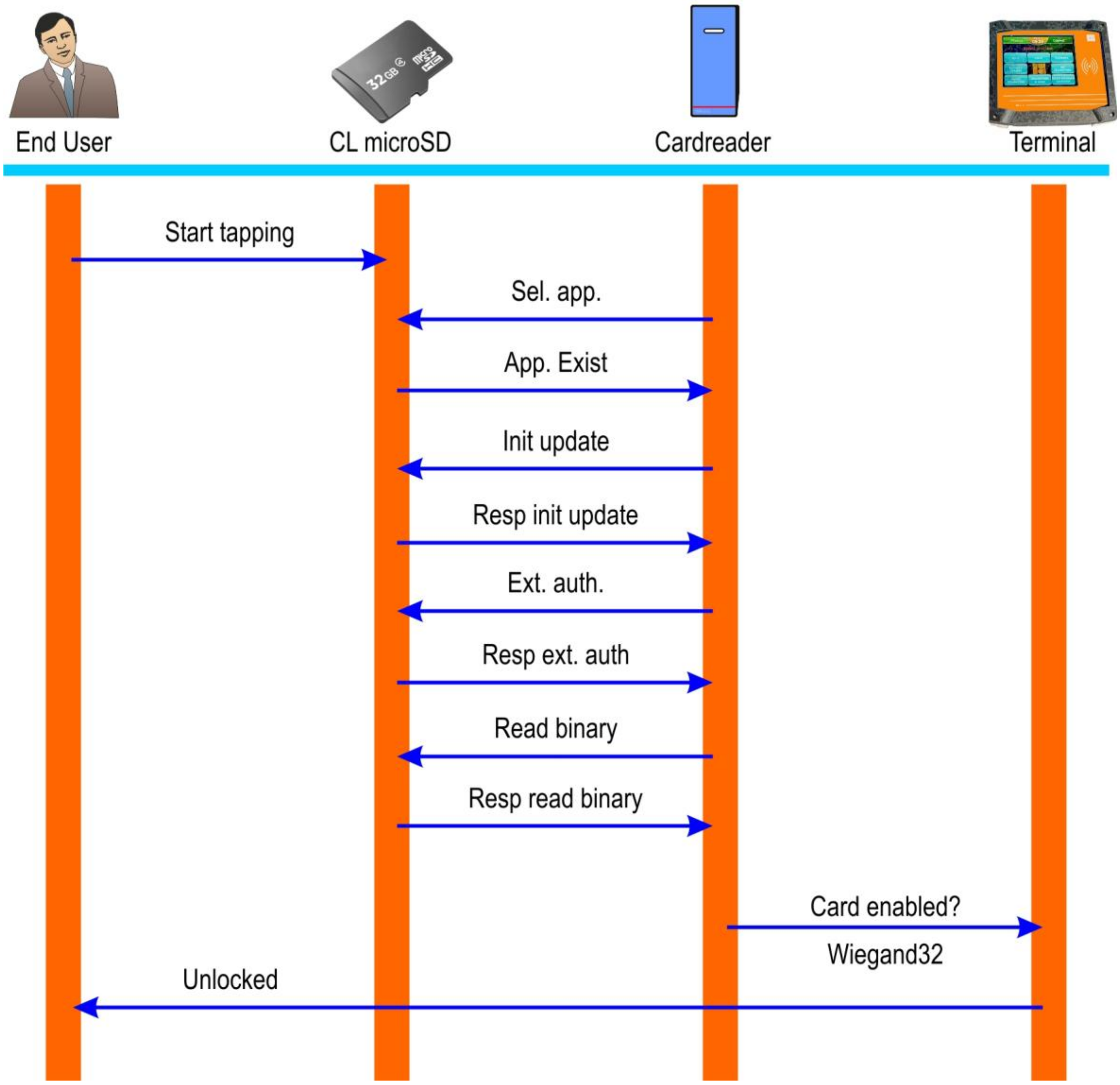


Figure 9: Simple identification card reading scenario flow

2.2.6 Communication scheme for PIN card reading scenario

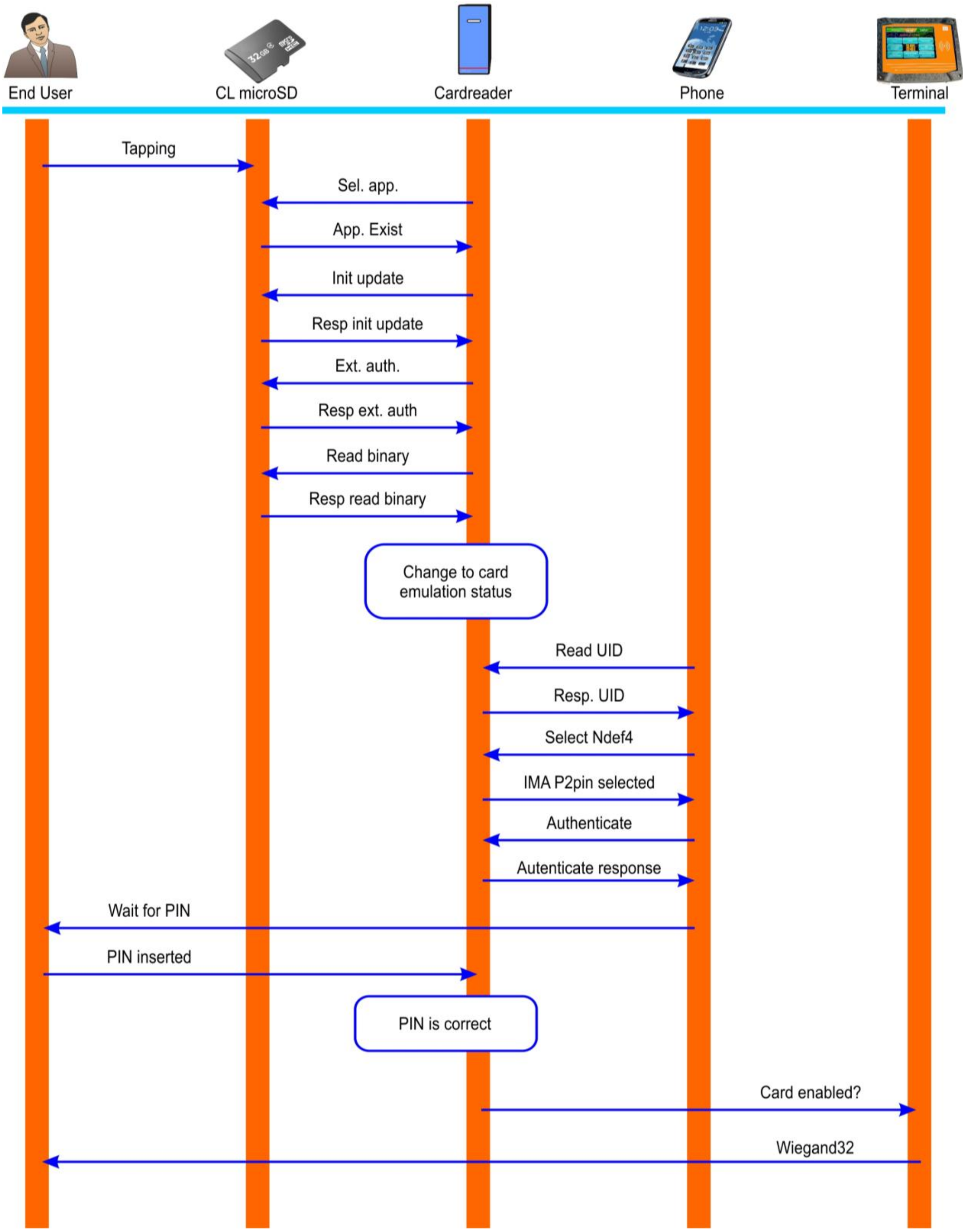


Figure 10: PIN card reading scenario flow

2.3 Requirements from Use Case 3 – Advanced ticketing application

2.3.1 Introduction

In ticketing applications, there are two major concerns that need to be tackled: ticket authenticity and customer privacy. From an operator’s standpoint, it is very important that tickets are not forgeable and that there is some methodology that can provide verifiable evidence that a ticket is valid. The only trustable way to solve this authenticity problem is to use cryptography embedded in tamper-resistant secure elements. In this context, the secure element can be a sophisticated smart card or even a low-cost RFID tag that was infused with the necessary cryptographic hardware or software implementations. In terms of cryptographic techniques, either public-key cryptography or symmetric-key cryptography can be used.

In symmetric-key cryptography, the verifier uses a unique identifier to retrieve the device's symmetric key from a central database. Then, the common symmetric key is used within a challenge-response protocol to cryptographically prove the authenticity of the ticket (the secure element). Note that all verifiers always have to be online to access the central storage of tickets (identifiers and symmetric keys). Unfortunately, this always-online requirement demands elaborate and laborious networks and back-end systems. An offline alternative consists in deriving the device's symmetric key from its identifier and a master secret key. However, the master secret key must be securely embedded into each and every verifier, thus increasing the risks of that key being compromised. When compromised, the master secret key can be used to clone any device registered in the system. These online and offline ticketing scenarios therefore offer very limited flexibility and security guaranties.

Quite nicely, when public-key cryptography is used instead, a ticket can be verified without the verifier being online and having access to the central server. The ticket comes along with a public key and a certificate that assert the validity of that public key, which can be used to authenticate the device towards any third party. Similarly to before, some appropriate challenge-response protocol can be used to verify the authenticity of a ticket held in the device.

However, both symmetric and public-key schemes come along with a downside for the customer. By having access to a network of readers, one may easily trace the route and whereabouts of each and every customer. Such a privacy concern has been a major hindering reason for not switching to cryptography-based ticketing solutions in many countries. The underlying problem for both symmetric and public-key cryptographic schemes resides in unique device identifiers. In symmetric-key cryptography, a unique identifier is used to access the symmetric key from the database or through key derivation. In public-key cryptography, the device's public key can also be interpreted as a unique identifier.

To solve these two requirements, ticket authenticity and user privacy, more advanced algorithms are needed. Such algorithms might also be based on pairings (a.k.a bilinear maps on appropriate elliptic curves), a powerful design tool that gives rise to protocols that are otherwise inefficient, unknown or unlikely to exist using classical, symmetric or public-key cryptography. However, pairing-based cryptography remains a comparatively young technology with significant overhead costs when implemented in (lightweight) secure elements.

2.3.2 Entity roles in Anonymous Ticketing

Although the exact algorithms and protocols that will be demonstrated in this advanced ticketing application are subject to investigation within WP2, a high-level view of all involved parties is provided below.

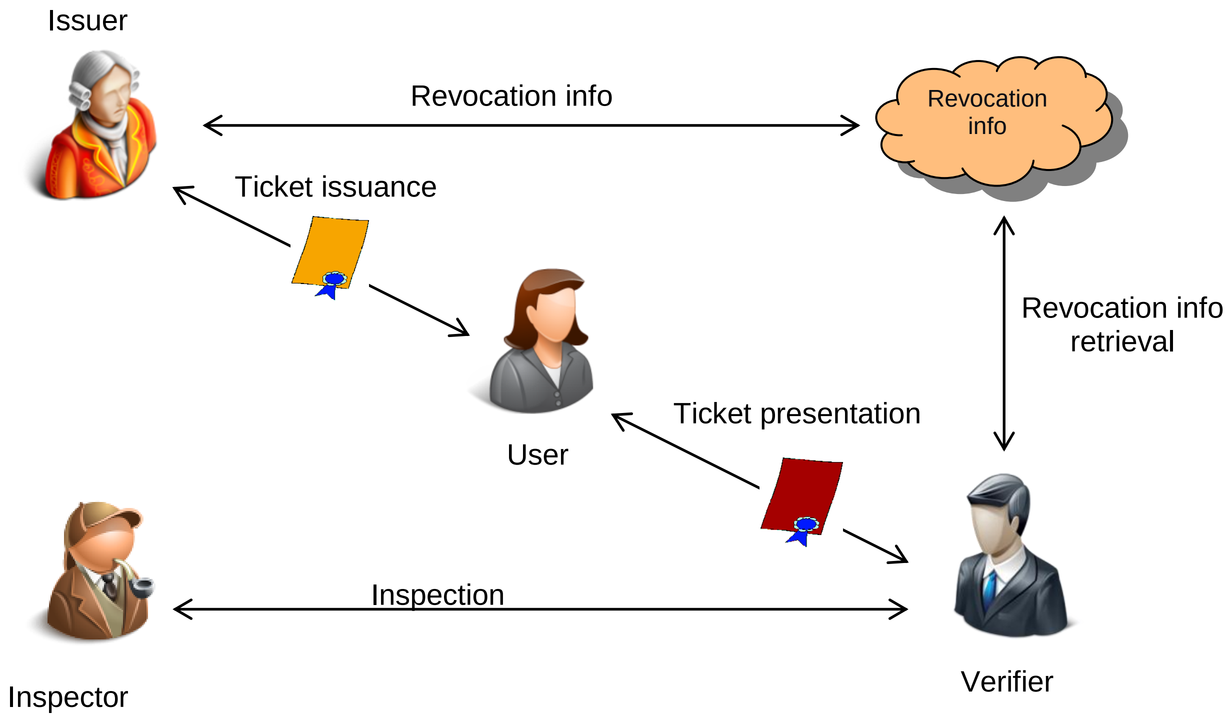


Figure 11: Advanced ticketing system overview

Issuer: the Issuer is an entity that can deliver a valid ticket to a user on request. The ticket itself can be seen as a credential that is downloaded and securely held in the user's device. The Issuer is typically a server operating a Web API over HTTP connections.

User (device): the User is represented by her/his device, which contains one or more tickets and performs cryptographic operations on her/his behalf. In this use-case, the device can typically be the µSD card supporting active NFC. However, it is intended to resist logical and physical attacks (side-channel and fault-based attacks) aiming at compromising its internal secret keys or tickets. Even though the device typically embeds cryptographic accelerators, its computational power and energy consumption remain limited. The device is capable of performing the ticket issuance protocol together with the Issuer through an online connection, as well as the ticket presentation protocol with Verifiers through a near-field transaction.

Verifier: a Verifier is a key-less entity that performs ticket presentation interactively with user devices. The presentation either succeeds or fails, depending on the characteristics of the presented ticket: it must be authentic and unrevoked. A Verifier may get online from time to time, for instance to update its revocation information. However, ticket presentation in itself shall not require the Verifier to be online. The Verifier is basically a component of the turnstile, equipped with an appropriate NFC reader, and shall not require a security module to perform verification. Its computational resources are somewhat limited and comparable to the ones of a smartphone.

Inspector: the Inspector (a.k.a identity opener or re-identification authority) is a trusted authority capable of revealing the identity of Users involved in ticket presentations. The inspection algorithm takes as input the transcript of an anonymous presentation and a master private key known as the inspection (or opening) key. A unique device identifier is recovered from inspection. The Inspector may be implemented as a Web service that responds to inspection requests based on an appropriate access control policy.

2.3.3 Life Cycle and User Experience

Tickets have the following life cycle:

1. **Issuance:** the Ticket Issuance protocol is played between the Issuer and the User's device with the following characteristics:
    a. The Issuer possesses a public/private key pair and the private key is required to create a new ticket, thus ensuring unforgeability.
    b. The device contains a Ticket Store that is, a memory zone where tickets are stored. Only one ticket is selected at any given time; however the device can switch between tickets upon request through a dedicated application on the smartphone/tablet.
    c. Beside tickets, the device is equipped with a PUF that has been enrolled beforehand.
    d. Ticket issuance does not require anonymity, so that both the Issuer and the User/device can also authenticate each other by any classical means at issuance time if necessary, typically if the User has a customer account on the Issuer's website.
    e. The output of the issuance protocol is the ticket itself, which is a cryptographic element produced together by the Issuer (using the issuing private key) and the device (using the PUF key).

Requirement	Requirement Type	Comment
Unforgeability of tickets	Security proof	Tickets cannot be created without the Issuer's private key
Undeniability of devices	Security proof	Only devices with enrolled PUF can issue tickets together with Issuer
Issuance < 800 ms (objective)	Performance	Performance objective for issuing a ticket on a device, excluding transmission delays

2. **Presentation:** ticket presentation occurs at verification time. The protocol is played between the User's device and a key-less Verifier in a challenge-response style. It shall observe the following features:
    a. At the end of the presentation stage, the Verifier can make sure that the ticket residing on-board the device, even though it remains unknown, is authentic in the sense that it was produced by the Issuer.
    b. However, ticket presentation does not reveal any information about the User's identity. More specifically, presentations shall be untraceable (the User's identity remains hidden) and unlinkable (the Verifier cannot, nor anyone else for that matter,

tell apart presentations of the same ticket from presentations of different tickets). Untraceability and unlinkability provide 2 different levels of anonymity. Unlinkability implies untraceability but the converse may be untrue.

c. Using a dynamically evolving revocation information (some dedicated crypto elements), the presentation protocol establishes that the ticket is or is not currently revoked, with total certainty.

Requirement	Requirement Type	Comment
Untraceability of presentations	Security proof	Provably infeasible to tell whether a presentation matches a given user/device
Unlinkability of presentations	Security proof	Provably infeasible to tell whether 2 presentations involve the same user/device
Presentation < 300 ms (objective)	Performance	Performance objective for presenting a ticket to a verifier e.g. entrance gate in the metro
Security against implementation-level attacks	Tamper-resistance	Design the algorithm such that certain implementation attacks (DPA, FA, etc.) are not possible and perform practical evaluations to ensure resistance against those attacks.

3. **Inspection:** the Inspector performs inspection alone, using a private system-wide inspection key. The input of the inspection algorithm is a presentation transcript. It returns the unique identifier of the User's device that was involved in the transaction. Optionally, ticket inspection can be verifiable, in which case the Inspector issues some cryptographic evidence that the identified user device is correct given the presentation transcript. Verifiable inspection ensures that Users cannot be framed illegitimately by the Inspector and therefore that identified Users cannot deny ticket presentations they were involved in.

Requirement	Requirement Type	Comment
Unframeability of inspected users	Security proof	Provably infeasible for the Inspector to forge a proof that a given transcript matches an incorrect User
Inspection < 500 ms (objective)	Performance	Performance objective for inspecting a ticket.
Security against implementation-level attacks	Tamper-resistance	Design the algorithm such that certain implementation attacks (DPA, FA, etc.) are not possible and perform practical evaluations to ensure resistance against those attacks.

4. **Revocation:** The revocation algorithm is performed by the Issuer and makes use of the Issuing key. Revocation takes as input the identifier of a User's device, possibly auxiliary User-

dependent information stored by the Issuer, an older version of the revocation information, the private issuing key and returns an updated version of the revocation information. Revocation information refers to a dedicated crypto element that is used by Verifiers to ascertain that a ticket is unrevoked at presentation time. The revocation information is public and can typically be stored at a prescribed location in the Cloud. A revoked ticket is useless and should therefore be removed from the device's Ticket Store.

Requirement	Requirement Type	Comment
Unforgeability of revocation info	Security proof	Provably infeasible to create a valid revocation info without the issuing private key
Revocation < 500 ms **(objective)**	Performance	Performance objective for revoking a ticket

5. **Transfer (optional):** the transfer protocol allows a ticket to move from one device to another with the mutual consent of their owners. Transferring a ticket shall take place online, potentially with the participation of the Issuer.

Requirement	Requirement Type	Comment
Uncloneable tickets	Security proof	Provably infeasible to clone a ticket
Transfer < 900 ms **(objective)**	Performance	Overall performance objective for transferring a ticket
Security against implementation-level attacks	Tamper-resistance	Design the algorithm such that certain implementation attacks (DPA, FA, etc.) are not possible and perform practical evaluations to ensure resistance against those attacks.

Chapter 3 System Architecture Requirements

3.1 Framework Description

The purpose of this section is to clarify the System Architecture requirements for the Matthew project, and define the scope addressed by the work packages in the project.

The MATTHEW application framework, see Figure 1, and the use case described in chapter 2, allowed to identify clearly the elements involved in the MATTHEW System Architecture outlined in the Figure 12.

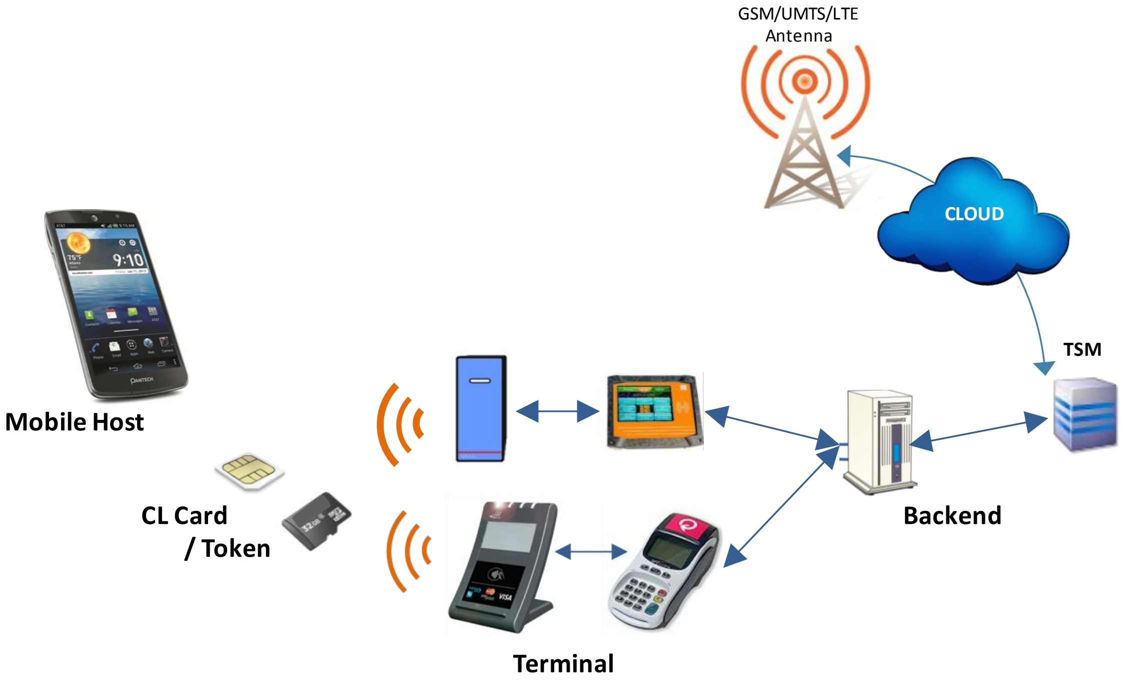


Figure 12: MATTHEW System Architecture

The following sections will detail the requirements for each of these elements, and indicate in which Work Package they will be further investigated. The requirements detailed in this section will also serve a base for the MATTHEW platform specification.

3.1.1 Elements not covered by the MATTHEW project

The security of the mobile handset platform itself is not in the scope of the MATTHEW project. The mobile and terminal requirements applicable to the MATTHEW project are detailed in the sub sections below.

3.2 Software Requirements

This section focuses on the software requirements of the different elements of the MATTHEW system architecture for the applications targeted in the project. These software requirements will serve as a base for the application development done in WP4 for the implementation of the application.

The current stage of the project allowed the identification of the requirements for the software on the MATTHEW token, the mobile host, Contactless terminal and the Backend Terminal for the two existing applications being transferred and enhanced on the MATTHEW platform: **Mobile Payment and Access Control applications**.

As the protocols and algorithms used for the **advanced ticketing application** are still subject of investigation and research, the requirements are not yet clearly defined. We can however summarize that all involved parties (ticket, mobile host, terminal, and back-end) must be capable of computing strong public-key cryptography and maybe even pairing-based cryptography. As it is currently unknown which type of cryptography is required for each component, as this is highly dependent on the to-be-defined algorithms, we for now assume that each component must be capable of computing both public-key and pairing-based cryptography in runtimes that allow interactive protocols.

The following sections identify the software requirements for the equipments requested for the Mobile Payment and Access Control use case described previously.

3.2.1 CL Card/Token Software Requirements

Card/Token Software Criteria	Mobile Payment Requirement	Access Control Requirement
Platform	M : Java Card 2.2 or higher	M : Java Card 2.2 or higher
Standards	M : Global platform 2.1.1 or higher	M : Global platform 2.0.1
Applications	M : Mastercard Mobile Paypass (MPP 1.0) O : Visa Mobile Payment Application (VMPA 1.4.1)	M: IMA Secure identifier application

M – Mandatory O – Optional

3.2.2 Mobile Host Software Requirements

Mobile Host Software Criteria	Mobile Payment Requirement	Access Control Requirement
Supported OS	M: Android 2.3 or higher	M: Android 4.0. or higher
Accesses	M : Mass storage access O : Device identification O : Network access O : NFC interface status inquiry	M : Mass storage access O : Device identification O : Network access O : NFC interface status inquiry
Capabilities	M : Enable/Disable contactless interface O : Read historical file from MCP application	M : App. For PIN insertion O : App. For transferring ID Applet to MicroSD

M – Mandatory O – Optional

3.2.3 Terminal Software Requirements

Terminal Software Criteria	Mobile Payment Requirement	Access Control Requirement
Embedded Applications	M : Mastercard Paypass O : Visa PayWave	M : Standard CKP04 IMA firmware
Capabilities	M : Offline authentication	

M – Mandatory O – Optional

3.2.4 Backend Software Requirements

Backend Software Criteria	Mobile Payment Requirement	Access Control Requirement
System server		M : IMA K4 server.
Remote access / provisioning		O : Application for transferring ID applet to the phone MicroSD (TSM)

M – Mandatory O – Optional

3.3 Hardware Requirements

Similar to the software requirements, the hardware requirements for the **advanced ticketing application** are still subject of investigation. Nevertheless, the current standing of research identified the computational capabilities and the interfaces of the involved secure elements as two of the most critical hardware components. To bring the computational requirements into perspective, it is important to realize that public-key cryptography already is many times more laborious than symmetric-key cryptography. On top of that, the algorithms that are based on pairing-based cryptography take even longer to compute. Also, the hardware requirements must involve countermeasures against implementation attacks. And as implementer it is important to be secure against all type of attacks, e.g., novel attacks on pairing-based cryptography or differential-power-analysis attacks on symmetric cryptography. Note that even pairing-based cryptography heavily takes advantage of symmetric-key primitives.

This section details the hardware requirements of the MATTHEW system architecture for the two application domains with clearly defined protocols and algorithms: Mobile Payment and Access Control applications.

The first part of the requirements, in section 3.3.1, is relevant to the MATTHEW contactless card / token (microSD, nanoSIM) identify the constraints to be addressed in WP3 by the active transmission technology and the antenna designs for the small form factor. The 2nd part of the requirements, in section 3.3.2 and 3.3.3, focus on the mobile host that will embedded the MATTHEW contactless card / token, and the Terminal with which the two applications will have to interact to complete the scenarios of chapter 2, and requested for the applications implemented in WP4.

3.3.1 CL Card/Token Hardware Requirements

3.3.1.1 Card Interface Requirements

Card Interface Criteria	Mobile Payment Requirement	Access Control Requirement
Form Factor	M : microSD card	M : microSD card
Compliancy	M : SD 2.0 or lower	M : SD 2.0 or lower
Power class	M : 3.0v – 3.6v	M : 3.0v – 3.6v
Memory size	M : Flash less O : 2 GBytes or higher	M : Flash less O : 2 GBytes or higher

M – Mandatory O – Optional

3.3.1.2 Contactless Interface Requirements

Contactless Interface Criteria	Mobile Payment Requirement	Access Control Requirement
Protocol	M : ISO14443-4A or ISO14443-4B	M : ISO14443-4A
Bit rate	M : 106 Kbit/s	M : 106 Kbit/s
Identifier Size	M : 4, 7 or 10 bytes (Type A) and 4 bytes (Type B)	M : 4, 7 or 10 bytes
Identifier Type	M : Fix or Random	M : Fix or Random
Operational distance	M : 0 – 2cm	M : 0 – 2cm
Transaction Time	M : 0,4s (768bits RSA key)	M : 0,5 sec

M – Mandatory O – Optional

3.3.1.3 3D RF Characteristics Requirements for transferable NFC components

To overcome the limitations of the metallic environment in a mobile phone like metal socket or battery, on the one hand active transmission is necessary to achieve enough performance for transmission from the microSD card or nanoSIM to the reader. On the other hand, the antenna designs of such form factors shall consider the spread of the RF field with respect to the metallic influence.

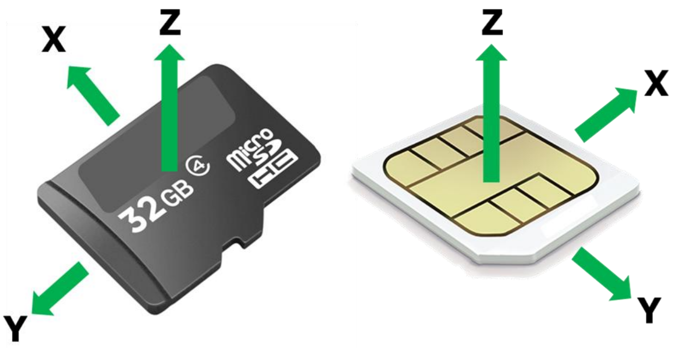


Figure 13: RF field vectors of microSD and nanoSIM

Figure 13 shows three vectors of the RF field for microSD card and nanoSIM. Based on investigations on state of the art mobile phones the antenna design should be optimized for two of these components, e.g. Z and Y component to handle a broader set of mobile devices.

Measurements on a Vivopay payment reader, which is a probable reader device for the small form factors researched within MATTHEW, have shown that a modulation strength according ISO/IEC 10373-6 test specification (test assembly 2 used) of 6 mV_{peak} in a reference socket leads to an maximum operating distance of approximately 3.5 cm, and 3 mV_{peak} leads to a maximum operating range of 2 cm.

	Minimum Modulation Strength in a Reference Socket (measured acc. ISO/IEC 10373-6 test specification)	
	microSD card	nanoSIM
Z component	6 mV_{peak}	6 mV_{peak}
Y component	3 mV_{peak}	3 mV_{peak}
X component	-	-

Based on these requirements, simulation models of the reference socket and the transferable device like the microSD card will be developed to optimize the 3D RF characteristics of the antenna designs. Due to the large selection of available metal sockets in mobile phones or tablets, these activities are necessary to achieve an adequate system performance of such transferable devices.

3.3.2 *Mobile Host Hardware Requirements*

Mobile Host Hardware Criteria	Mobile Payment Requirement	Access Control Requirement
Operating system	M : Android 2.3 or higher	M : Android 2.3 or higher
Card extension slot	M : microSD card	M : microSD card
Card compliancy	M : SD 2.0 or higher	M : SD 2.0 or higher
Card slot location	M : Lateral/External slot O : Internal slot	M : Lateral/External slot O : Internal slot
Back cover	M : No metallic material	M : No metallic material
Handset Model	O : Sony Xperia Z or similar	O : Samsung Galaxy S3 or similar

M – Mandatory O – Optional

3.3.3 Terminal Hardware Requirements

Terminal Hardware Criteria	Mobile Payment Requirement	Access Control Requirement
Contactless Protocol	M : ISO14443-4A and ISO14443-4B	M : ISO14443-4A In contactless reader RSW04 connected to CKP04 terminal
Features	O : Keypad O : Display	O : Keypad O : Display
Connectivity	O : RS232 or USB2.0	O: Wiegand interface, RS485 interface
Model	O: Vivopay	O : CKP04 IMA terminal

M – Mandatory O – Optional

3.4 Security Requirements

The requirement for security is an omnipresent constraint that has to be considered in each and every design step. Both logical and physical security threats have to be regarded. While logical security threats misuse given interfaces, e.g., to perform buffer overflows or access restricted memory, physical security threats are much more delicate. In physical threat scenarios an attacker could measure timings, measure power consumptions, or actively induce faults. Those threats have to be considered during the specification, design, implementation, and evaluation phases.

In our research, we consider both logical and physical threat scenarios in all three use cases. In particular, there are several research questions that are of interest within the scope of MATTHEW:

- There is this subject of transferability that involves **multiple secure elements**. Those secure elements can be housed within a single smart-phone or multiple smart-phones. The yet to be answered question is how those secure elements communicate with each other in such a way that the security of the whole system cannot be compromised.
- The to-be-designed algorithm and its specific requirements on the functionality of a secure element are yet to be answered open questions. Especially when it comes to the requirements to allow **interactive protocols**, the secure element must provide **sufficient performance** to compute pairings and elliptic curve group operations within reasonable times.
- Additionally, the secure elements must compute all parts of the protocol (issuing, presentation, inspection, transferability) in a secure manner. However, as they pose a target for an adversary that is capable of performing **timing-analysis, power-analysis,** and **fault-analysis attacks**, they have to be secured against such types of attacks. To assure resistance it is necessary to practically perform such attacks on all parts of the underlying primitives (Pairing, ECC, AES, SHA, …).
- Additionally to the cryptographic primitives, the CPU of a secure element is still one of the most vulnerable targets. It is especially vulnerable against, e.g., fault attacks with which it is possible to skip instructions, function calls, or branch instructions. Therefore when designing a secure element, it is not only important to **guard the cryptographic primitives**, but also the **CPU** and the **system as a whole**.
- How to securely store all involved keys is still an open research question. **PUFs** not only allow to derive unique private keys but also to store private keys in a secure manner. However, as PUFs are still a comparatively young technology, there are still a lot of open research questions to be answered.

Chapter 4 Conclusion

This document concludes the work done by the partners in task 1.1 on MATTHEW Use Case Requirements and in task 1.2 on MATTHEW System Architecture Requirements. By describing these requirements for three different application domains with different maturity level this work sets the bases for the work to be addressed in the following MATTHEW Work Package and Tasks.

These requirements will be further completed for the Advanced Ticketing application once the WP2 progressed on the protocols and algorithms used. The current state of the requirements allows starting the work of task 1.3 in WP1 for the MATTHEW platform specification, the WP3 - Component Hardware development, and soon the WP4, Application development.

Chapter 5 List of Abbreviations

AID	Application Identifier
MCP	Mobile Contactless Payment
TSM	Trusted Service Manager
POS	Point of Sale
CVM	Cardholder Verification Method
EOT	End of transaction
JCOP	Java Card OpenPlatform
NDEF	NFC Data Exchange Format
NFC	Near Field Communication
IC	Integrated Circuit
SIM	Subscriber Identification Module
PIN	Personal Identification Number
SD	Secure Digital
POI	Point of Interaction
AES	Advanced Encryption Standard
PUF	Physically Unclonable Function
GSM	Global System for mobile communications
ECC	Elliptic Curve Cryptography
SHA	Secure Hash Algorithm