



Mobile Proximity Payment Issues and Recommendations

Mobile Payment Configuration and Maintenance

Version 1.0
October 2006

Table of Contents

1. PREFACE	5
1.1 ABOUT MOBILE PAYMENT FORUM	5
1.2 DOCUMENT PURPOSE AND SCOPE	5
1.3 REFERENCES	6
1.4 DEFINITIONS	6
1.5 TERMINOLOGY	8
1.6 PROXIMITY PAYMENT ARCHITECTURE	8
1.7 TECHNOLOGICAL FOCUS	9
1.7.1 Proximity interface	10
1.7.2 Wide-Area communication	10
1.7.3 Interface from the application to smart cards	10
1.7.4 Application environment	10
2. PROVISIONING AND PERSONALIZATION	11
2.1 INTRODUCTION	11
2.1.1 Possible Options for Device Provisioning	11
2.1.2 Guiding Principles and Assumptions	11
2.2 HIGH LEVEL ARCHITECTURE FOR PROVISIONING	11
2.3 ROLES IN PROXIMITY PAYMENT PERSONALIZATION	12
2.3.1 User	13
2.3.2 User Device	13
2.3.3 Account Issuer	13
2.3.4 Application owner	14
2.3.5 Application developer	14
2.3.6 Application Loader	14
2.3.7 Account data generator	14
2.3.8 Account data loader	14
2.3.9 Application Security Domain Manager	14
2.3.10 Application Security Domain Controller	15
2.3.11 SE Platform Controller	15
2.3.12 SE Platform Manager	15
2.3.13 Registration Manager	15
2.4 MODELS	16
2.4.1 SE Platform Controller centric model	16
2.4.2 Multi-application providers model	17
2.4.3 Delegated model	18
2.4.4 Application provider centric model	19

2.4.5 Controlling authority model	19
2.5 KEY MANAGEMENT	21
2.5.1 Parties	21
2.5.2 Distributed Model	21
2.5.3 Centralized Key Management Authority Model	23
2.6 PROVISIONING SECURITY REQUIREMENTS	25
2.6.1 Protection of Payment Assets	25
2.6.2 Protection of Transaction Data	26
2.7 PROVISIONING SECURITY ISSUES	27
2.7.1 SE Platform Controller and Manager	27
2.7.2 User Registration	28
2.7.3 Authenticity / Integrity of the Payment Application / Credentials	30
2.8 PROVISIONING PROCESS ISSUES	31
2.8.1 Payment Application / Credentials Installation Process	31
2.8.2 Update of Payment Application and Credentials	33
2.8.3 Payment Application and Credentials Deletion	35
2.8.4 Subscription Change	36
2.8.5 Device Change with Embedded Secure Element	37
2.8.6 Device Change with Removable Secure Element	38
2.8.7 Backup / Restore of Payment Application and Credentials	39
3. ADDITIONAL SECURITY ISSUES	42
3.1 INTRODUCTION	42
3.2 MOBILE DEVICE ARCHITECTURE	42
3.2.1 Logical Architecture	42
3.2.2 Implementation Options for the Secure Element	44
3.2.3 Secure Element Security Level	44
3.2.4 Secure Element Support for Post-Distribution Personalization	45
3.2.5 Standardisation of Secure Element Application Environment	46
3.3 MOBILE PLATFORM SECURITY ISSUES	46
3.3.1 Mobile Platform Application Environment Choice	47
3.3.2 Mobile Platform Security Level	47
3.3.3 Access to the Secure Element	49
3.3.4 Access to Proximity Modem	49
3.4 SECURE ELEMENT ISSUES	50
3.4.1 SIM/USIM	50
3.5 APPLICATION SECURITY ISSUES	52
3.5.1 Isolation of Payment Application	52
3.5.2 PIN Code	52
3.6 APPLICATION AND PLATFORM APPROVAL ISSUES	54
3.6.1 Approval and Approving Bodies	54
3.6.2 Lifecycle for Mobile Devices	56
3.6.3 Analogue Approval	56
3.6.4 Digital Approval	57
3.6.5 Approval of Secure Element	57
3.6.6 Conveying Approval Level of Secure Element to Issuers	58

4. CUSTOMER CARE – OPERATOR AND CARD ASSOCIATION	59
4.1 INTRODUCTION	59
4.1.1 Guiding Principles and Assumptions	59
4.2 CUSTOMER SERVICE ISSUES	59
4.2.1 Customer Service Process for Problem Resolution	59
4.2.2 Customer Support/Education	60
4.2.3 Auditable Customer Service Trail	61
5. USABILITY	62
5.1 INTRODUCTION	62
5.1.1 Guiding Principles and Assumptions	62
5.2 TRANSACTION EXPERIENCE USABILITY ISSUES	62
5.2.1 User Authentication	62
5.2.2 Payment Credential Selection Process	64
5.2.3 Payment Credential Availability with Device Off	65
5.2.4 Payment Interaction with Phone Operation	67
5.3 NETWORK INTERACTION ISSUES	68
5.3.1 Provisioning and Personalization Process Requirements for the User	68
5.3.2 Payment Credential Availability Off-Network	69
5.4 FORM-FACTOR ISSUES	70
5.4.1 Physical Positioning of a Mobile Device for Payment	70
5.5 BRANDING	71
5.5.1 Branding on the Proximity-enabled Mobile Device	72
5.5.2 Branding via the User Interface	72
5.6 RECEIPTS	72
5.6.1 Delivery of Receipts	72
6. INTERWORKING	72
6.1 INTRODUCTION	72
6.2 APPLICATION ISSUES	72
6.2.1 Multiple Payment Applications on A Mobile Device	72
6.2.2 Anti-collision	72
6.2.3 Conflicting Analogue Requirements	72
6.2.4 Multiple Proximity Applications on a Single Device	72
6.3 ENVIRONMENTAL ISSUES	72
6.3.1 Use where Mobile Telephony is prohibited	72
6.4 MULTI-TECHNOLOGY ISSUES	72
6.4.1 Support of multiple Contactless RF technologies in a single device	72
7. CONCLUSION	72
7.1 SUMMARY	72
7.2 RECOMMENDATIONS	72

1. Preface

1.1 About Mobile Payment Forum

The Mobile Payment Forum brings together leading organizations from the mobile and financial industries to create a foundation for standardized, secure, and authenticated mobile payments. A global, member-driven organization, the Forum seeks to achieve broad industry cooperation to standardize the building blocks needed to deploy secure and convenient mobile commerce solutions.

Membership includes key financial institutions, telecommunications operators, wireless-device manufacturers, merchants, content providers, and software and hardware developers and vendors. The Mobile Payment Forum is a non-profit membership organization, incorporated in the state of Delaware in the United States.

1.2 Document Purpose and Scope

The Mobile Payment Forum engaged on a study of proximity payment in mobile devices. This activity involved a number of steps, of which this document is the last.

Firstly, the proximity payment trials and deployments which were then taking place were identified and analysed.

Secondly, an assessment of applicable technologies for mobile proximity payment was undertaken. The results of this assessment are detailed in *Proximity Payment Technology Assessment* [PPTA].

This document is the next deliverable in this activity and highlights issues which need to be addressed in the area of mobile proximity payment. Recommendations are made where appropriate, and in certain cases these have been expanded to provide requirements.

This document is written for consideration by those implementing mobile proximity payment systems, or components of a system. It also contains information of use to forums defining technology which may form a basis for implementation of a mobile proximity payment system.

1.3 References

- [14443] ISO/IEC 14443: *Identification cards -- Contactless integrated circuit(s) cards -- Proximity cards*, www.iso.org
- [MPAF] *Mobile Payments Architectural framework and Use Cases*, Version 1.0, Mobile Payment Forum, www.mobilepaymentforum.org
- [NFC] ISO/IEC 18092: *Information technology -- Telecommunications and information exchange between systems -- Near Field Communication -- Interface and Protocol (NFCIP-1)*, www.iso.org
- [PPTA] *Proximity Payment Technology Assessment*, Version 1.0, Mobile Payment Forum, www.mobilepaymentforum.org
- [RADP] *Requirements for Application Download and Personalization*, *Mobile Payment Configuration and Maintenance*, Version 1.0, Mobile Payment Forum, www.mobilepaymentforum.org
- [SATS] *Security and Trust Services API (SATSA) for Java™ 2 Platform, Micro Edition*, Version 1.0, JSR 177 Expert Group, Java Community Process (JCP), www.jcp.org
- [STIP] *STIP Core Framework Technology, A Technical Specification*, Version 2.2, GlobalPlatform, www.globalplatform.org

1.4 Definitions

BIP	Bearer Independent Protocol
CDMA	Code Division Multiple Access
Contactless RF	Generic term for multiple varieties of short-range RF communication technology
GP	GlobalPlatform
GPRS	General Packet Radio System
GSM	Global System for Mobile telephones
JCP	Java Community Process
JSR	Java Specification Request
LAN	Local Area Network
MMI	Man-Machine Interface
MMS	Multimedia Messaging
MNO	Mobile Network Operator

NFC	Near-Field Communication
OMA	Open Mobile Alliance
OTA	Over-the-Air
PAN	Personal Area Network
PDA	Personal Digital Assistant
PIN	Personal Identification Number
POS	Point of Sale
Proximity Payment	A payment where the consumer is physically present at the point of sale.
RF	Radio Frequency
RFID	Radio Frequency IDentification
RUIM	Removable Universal Identify Module
SATSA	Security And Trust Services API
SE	Secure Element
SIM	Subscriber Identity Module
SSO	Single Sign On
STIP	Small Terminal Interoperability Platform
STK	SIM Tool Kit
SMC	Secure Multimedia Card
UI	User Interface
UMTS	Universal Mobile Telephone System
URL	Uniform Resource Locator
USIM	Universal Subscriber Identity Module
WLAN	Wireless Local Area Network

1.5 Terminology

The use of terminology surrounding proximity technologies is often not well defined, with for example the terms RFID, contactless and proximity being used interchangeably.

For the purposes of this document, proximity refers to the relationship of the user to the point of sale, that is, a proximity payment is a payment where the consumer is physically present at the point of sale.

The proximity interface is the interface which is used to transfer transaction credentials between the mobile device and the point of sale terminal. The proximity interface may be implemented using a range of technologies (as discussed in the “*Proximity Payment Technology Assessment*” [PPTA]) and these include contactless technologies (such as those defined by ISO14443 [14443] or FeliCa), Near Field Communications [NFC], infra red and others.

The term “contactless RF” is used in this document to describe short range radio frequency communication technologies.

RFID (Radio Frequency Identification) is a particular use of contactless RF in which an identifier is transmitted from a tag to a reader. Typically the tag is a “dumb” tag, incapable of processing data. While it is possible to implement a proximity payment system based on RFID¹, in practice most proximity payment systems make use of smartcards or other devices capable of data processing in order to implement security features in the payment system. This distinguishes proximity payment from RFID.

1.6 Proximity Payment Architecture

The architecture used by the Mobile Payment Forum to describe mobile payment systems is described in detail in the Mobile Payment Forum document “*Mobile Payments Architectural Framework and Use Cases*” [MPAF]. The architecture is enhanced for proximity payment as shown in Figure 1. For reference, brief descriptions of each functional block are provided below.

¹ For example, systems described in sections 7.1.1 and 7.1.2 of the *Proximity Payment Technology Assessment* [PPTA] could be implemented in this way.

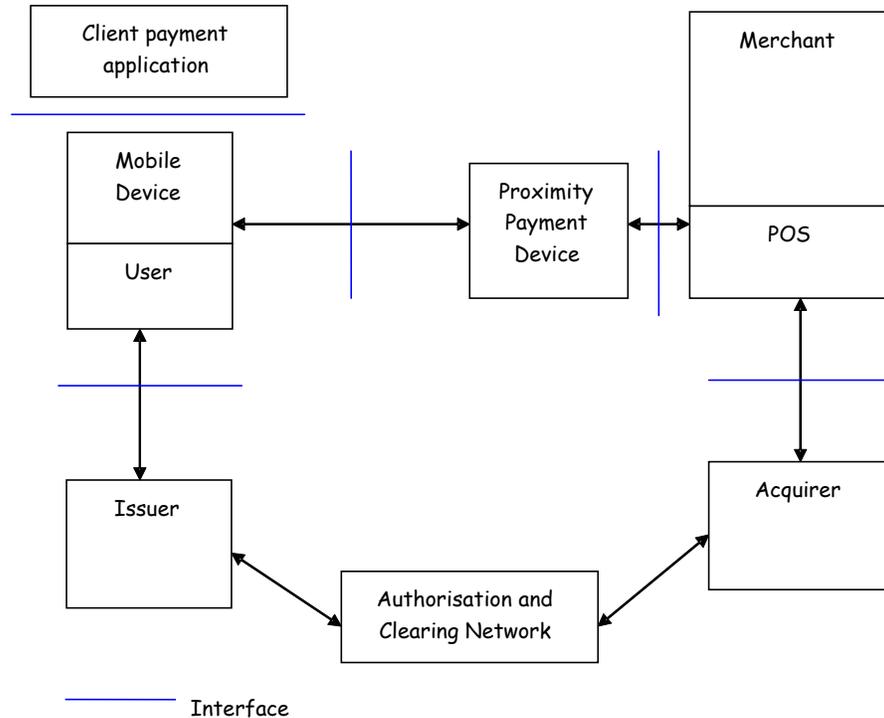


Figure 1 Proximity Payment Architecture

Client Payment application stores the payment credentials. The client payment application interfaces with the terminal.

Mobile Device communicates with the proximity payment device attached to the merchants POS using Bluetooth, IR, Contactless RF, other some other bearer technology.

Proximity Payment Device communicates with the client's terminal using infrared, Bluetooth, Contactless RF etc. to receive payment credentials and transmits them to the POS. The proximity payment device may be integrated with POS.

POS. The POS communicates with the acquirer (credit card acquirer or the wallet data store) to exchange transaction details. For credit card transactions the ISO8583 defines this interface in many countries.

1.7 Technological focus

A variety of technologies could be used as components in the architecture for proximity payment; many of these are detailed in the Mobile Payment Forum document "*Proximity Payment Technology Assessment*" [PPTA]. However, in order to narrow down the available options and to keep the work practical and focused, a set of technologies needs to be chosen.

2. Provisioning and Personalization

2.1 Introduction

Device provisioning involves the delivery of the payment application(s) and credentials necessary for a mobile device to be used as an instrument for making valid proximity payments. To provision a mobile device with the necessary information, a variety of security- and privacy-related issues must be addressed, in addition to other issues related to the process of managing the lifecycle of the payment credentials. This chapter details requirements and issues with discovery, installation and management of the payment application(s) and payment credentials that must be addressed to allow a mobile device to be used for proximity payment.

2.1.1 Possible Options for Device Provisioning

Different solutions for securely provisioning the payment application in mobile device are certainly possibly, depending upon the bank and mobile network operator business models, including:

- Provisioning during the post-personalization of the card (done by the SIM manufacturer, the operator or the bank in their offices).
- OTA provisioning offered via the mobile network operator by the operator or a trusted third party (bank or not).

A part of the provisioning could also be done during the production of the card; but this could lead to significant costs, and certain post-personalization steps would still essential to bind the card to the user account.

2.1.2 Guiding Principles and Assumptions

The provisioning issues detailed below assume that a mobile device will be provisioned using an Over-the-Air (OTA) approach, which is the approach which is deemed to be the most scalable.

2.2 High level Architecture for Provisioning

A high level logical architecture for provisioning a multi-application mobile device for payment applications was developed in the Mobile Payment Forum document, “*Requirements for Application Download and Personalization*” [RADP]. This architecture has been adapted for the case

of mobile proximity payment, and is shown in Figure 2: High Level Provisioning Logical Architecture below.

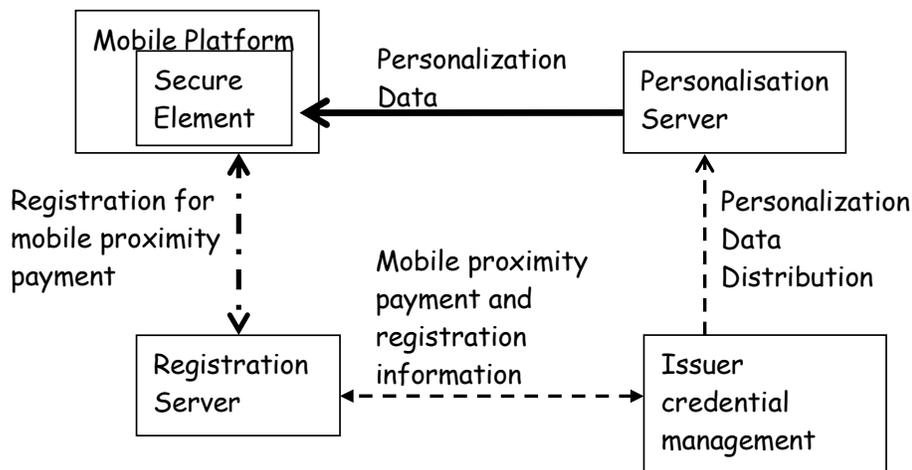


Figure 2: High Level Provisioning Logical Architecture

The elements of the provisioning architecture consist of the mobile device (the mobile platform with associated secure element), the *personalization server*, the *issuer credential management* system, and a *registration server*.

The mobile platform and secure element are as described above, and host the payment application and provide the proximity capability.

The *personalization server* is responsible for transferring the personalization data (application and/or payment credentials) into the secure element.

The *Issuer credential management* system is responsible for the distribution of the payment application (if necessary), and for the generation of the payment credentials in a form for the particular user and the particular secure element associated with that user.

The *Registration server* is responsible for the association of a user and their payment account with a particular mobile device and secure element. This includes associating the payment account with the address of the mobile device, authenticating the user to ensure they have the right to the payment account, and ensuring that the user is in possession of the registered mobile device.

2.3 Roles in Proximity Payment Personalization

This section refines the architecture and roles initially defined in section 2.2. The different roles which are played in personalization of mobile

devices for proximity payment are identified, as are the interactions between these different roles.

The following assumptions have been made in preparing this:

- The secure element on the mobile device is managed using cryptographic keys which are required in order to install an application (and possibly to personalize the application);
- The applications are personalized via the wide area modem (generally over the air).
- The mobile device requires a proximity payment application, which will control the flow of data over the proximity link between the mobile device and the point of sale terminal.
- This application needs to be personalized with account data specific to the customer.

Note that this section identifies the roles required, not the entities which perform the roles. One business entity may perform multiple roles in an implementation.

2.3.1 User

The user has a proximity payment account with an account issuer, and owns a mobile device to which the customer wishes the account to be provisioned.

2.3.2 User Device

The mobile device owned by the user which will be enabled for proximity payment.

2.3.3 Account Issuer

The Account Issuer has a relationship with the user. The account issuer holds the proximity payment account for the user, and is responsible for initiating the generation of account data (application personalization data). The account issuer is responsible for instructing the account data loader to load the account data to a user's mobile device. The account issuer is responsible for communicating with the application owner in order to ensure a compatible proximity application is available on the device.

The account issuer is responsible for obtaining permission from the application security domain controller (and possibly also the SE platform controller) to place account data on the secure element.

Security Domain Controller to install and personalize applications within the application security domain. The application security domain manager is responsible for communicating with the SE platform manager as required to perform these tasks.

2.3.10 Application Security Domain Controller

The application security domain controller is responsible for negotiating the creation and terms of usage for the application security domain with the SE Platform Controller.

The application security domain controller is responsible for approving what applications may be loaded into the application security domain.

2.3.11 SE Platform Controller

The SE platform controller is responsible for approving what applications may be loaded onto the secure element. The SE Platform Controller may delegate this responsibility for a class of applications to an application security domain controller.

2.3.12 SE Platform Manager

The SE platform manager is responsible for managing the SE platform cryptographic keys, and generating cryptograms as necessary to allow parties authorised by the SE Platform Controller to install and personalize applications on the SE. The SE Platform Manger is responsible for creating Application Security Domains under the instructions of the SE Platform Controller. The SE Platform Manager works with the Application Security Domain Manager to allow the management of an application security domain.

2.3.13 Registration Manager

The registration manager is responsible for the interface with the customer to request mobile proximity payment functionality on his or her mobile device. The registration manager is responsible for the correlation of a user's mobile device information and the user's account information. The registration manager provides this correlation to the account issuer, and optionally to the application owner.

The diagram in Figure 3 shows the relationships between the roles defined above. Black lines indicate data transfer, with optional data transfer relationships being dotted. Blue dashed lines indicate a business relationship. Optional business relationships are indicated by a dashed-dotted blue line.

The optional relationships indicate that depending on the implementation of the system, these relationships may be in place.

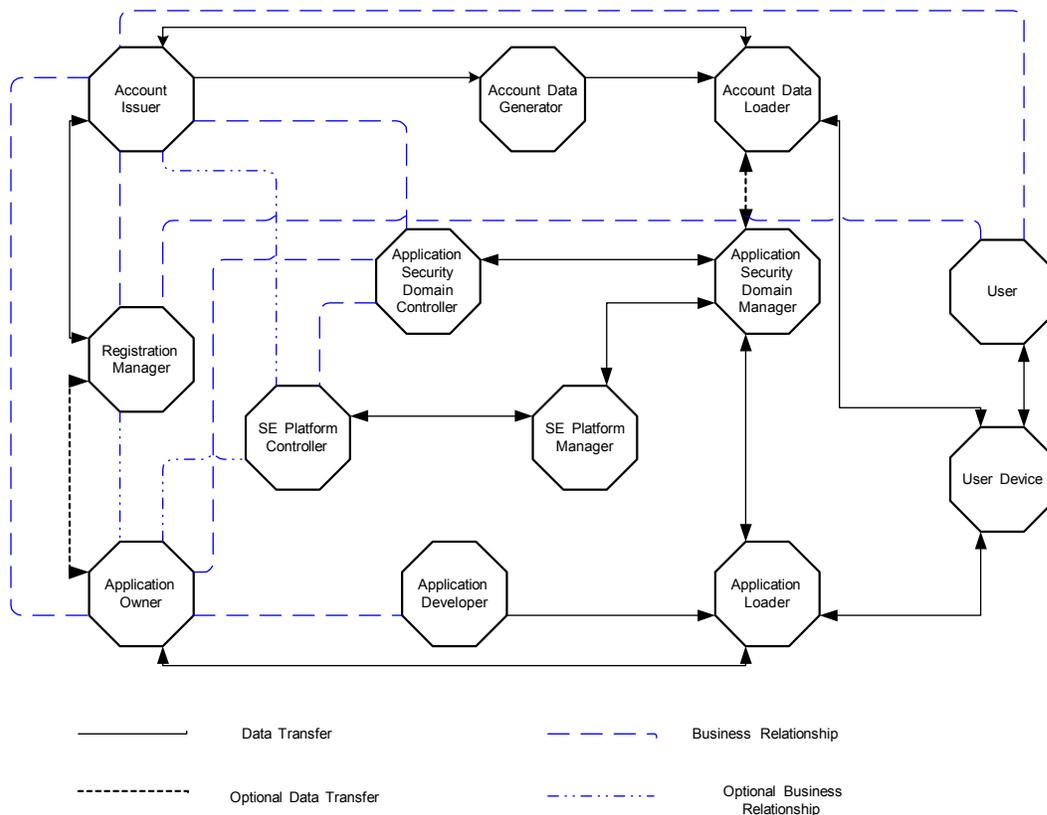


Figure 3 Personalization Architecture

2.4 Models

Given the personalization roles described above, several different models are possible. These models are described in detail below.

2.4.1 SE Platform Controller centric model

The simplest model with only one security domain is that of the SE Platform. All the applications are either the SE platform controller’s own applications, or under its control. In this model the card management framework (which manage secure application load, install, deletion) is performed only by the SE Platform controller, who also manages the platform. The application loader and account data loader roles are also played by the SE platform controller. This is illustrated in Figure 4.

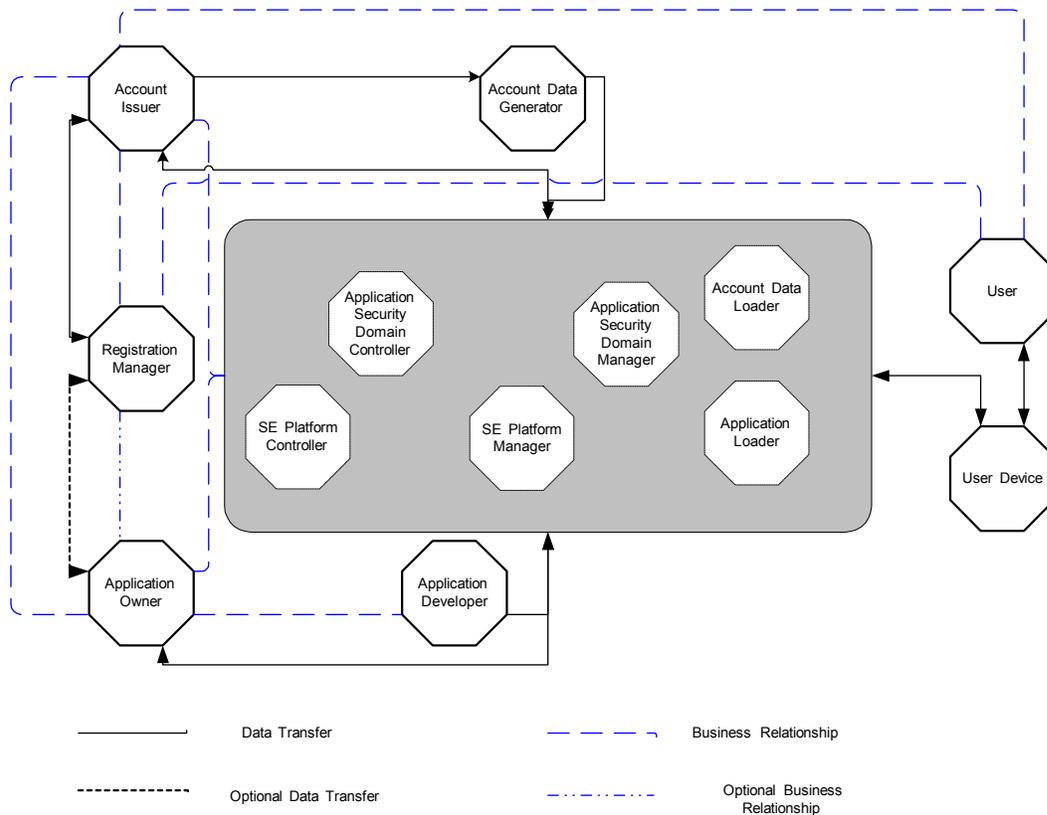


Figure 4 SE Platform Controller Centric Model

2.4.2 Multi-application providers model

This model is a slightly more sophisticated configuration with multiple Security Domains: the SE Platform Security Domain and Security Domain for Application Providers: the business partners of the SE Platform Controller sharing the card's real estate. Some applications are the SE platform controller's own applications or under its control while other applications are under the control of the different Application Security Domain Controllers. In this model, card management (creation of security domains, key management) is still performed only by the SE platform controller, who acts as the SE platform manager. The Application Security Domain controllers are allowed only to execute transactions with their applications but are not allowed to directly manage them. The application provider security domain verifies application load, integrity and authenticity. An option of this model is for the SE Platform Controller to be able to guarantee to the application Provider the integrity and authenticity of an application load operation: the Digital Authentication Pattern (DAP). This model is represented in Figure 5.

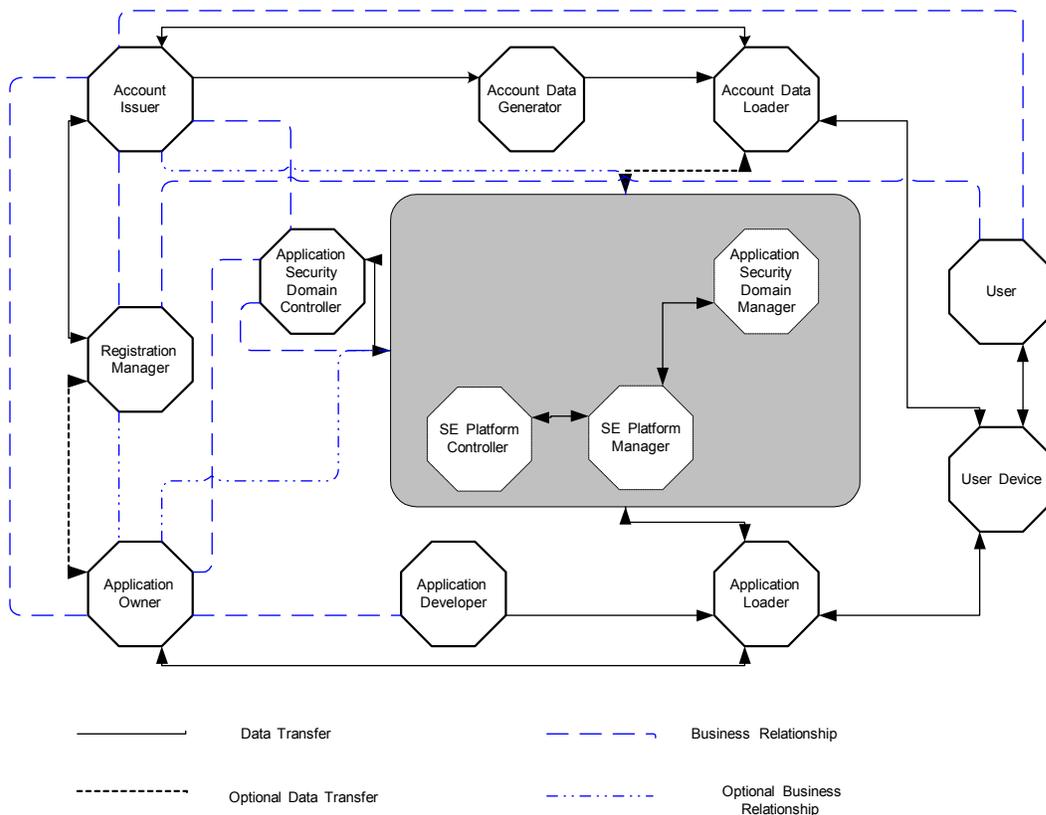


Figure 5 Multi-application Provider Model

2.4.3 Delegated model

This model is also based on the shared Security Domains between the SE platform controller and Application Security Domain controller; the difference is that not only the card's real estate is shared but also the management of the card. SE Platform management (creation of security domains, key management) is still performed by the SE platform controller but is also partially delegated to the Application Security Domain manager. In this model, the Application Providers (issuer, application controller) are allowed not only to execute transactions with their applications but are also allowed to directly manage them, but only their own applications. In this case the application provider is the application security domain controller and can delegate loading to an application loader. The ultimate control still remains in the SE platform controller) hands: the card management delegation is a specific privilege assigned by the Card Issuer (SE platform controller to a few Applications Security Domain Managers. Furthermore, each management operation, e.g. loading a new application, requires an explicit pre-approval from the SE Platform Manager (a Token). This model is represented by Figure 3,

where the application security domain manager needs to obtain the load token from the SE platform manager for each load.

2.4.4 Application provider centric model

In this model, application security domain is completely independent from the SE Platform security domain. It may have a full management capability of its domain (No token is needed). It even may define sub Security Domains in the primary Security Domain. This model is only possible with PKI infrastructure to manage the keys.

This model is illustrated in Figure 6. Note that the SE Platform manager is responsible for setting up the Application Security Domain (shown as a dashed arrow in Figure 6), and after that does not participate in the management of the application security domain.

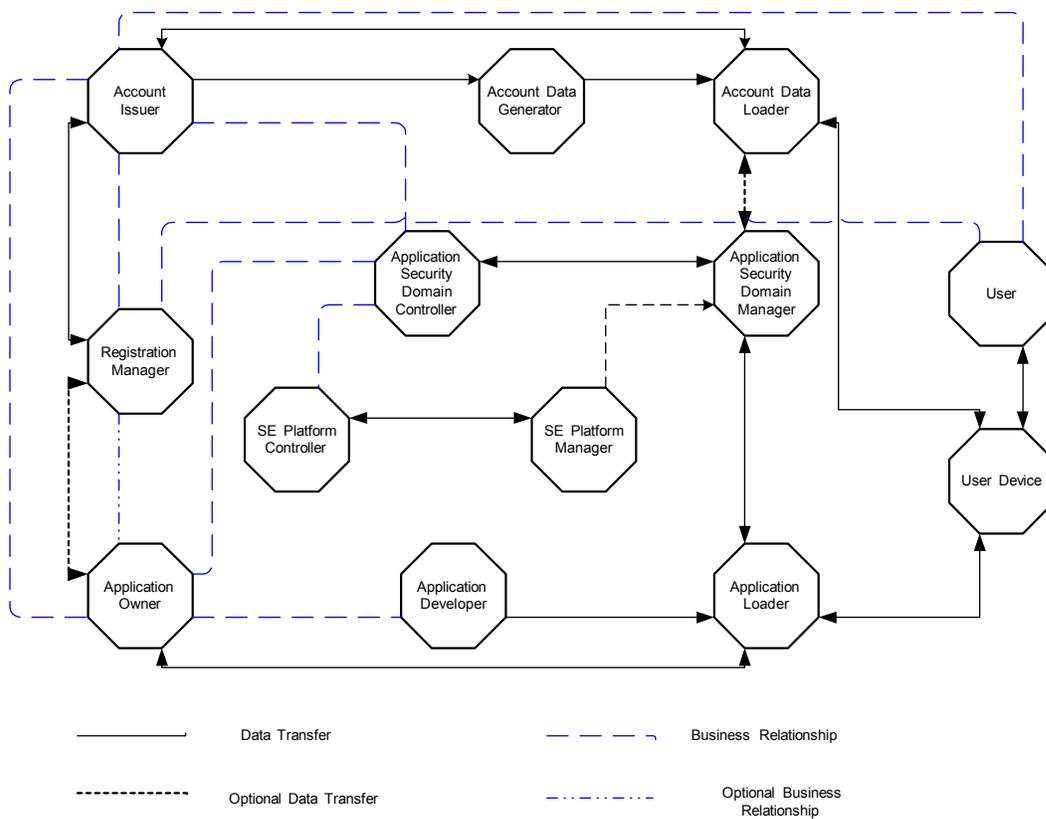


Figure 6 Application Provider Centric Model

2.4.5 Controlling authority model

In the controlling authority model there are multiple Security Domains: the SE Platform Security Domain, Security Domain for Application Providers, and a very special Security Domain dedicated to a Controlling Authority, a scheme wide authority. In this model, SE platform management is still performed by the SE platform manager and optionally by the Application Security Domain manager with Delegated Management privilege.

In both cases, the SE platform manager as well as the Application Security Domain manager are only allowed to load new applications to the card with the Controlling Authority's explicit pre-approval (Mandated Digital Authentication Pattern). The ultimate control for loading applications has shifted from the SE platform manager's hands to a scheme wide authority. This Controlling Authority's control only applies to load operations (verify and load of the applets).

This is illustrated in Figure 7, where a new entity, the Controlling Authority has been added. The SE Platform Manager and Application Security Domain Manager both communicate with this controlling authority to obtain the needed approvals for loading.

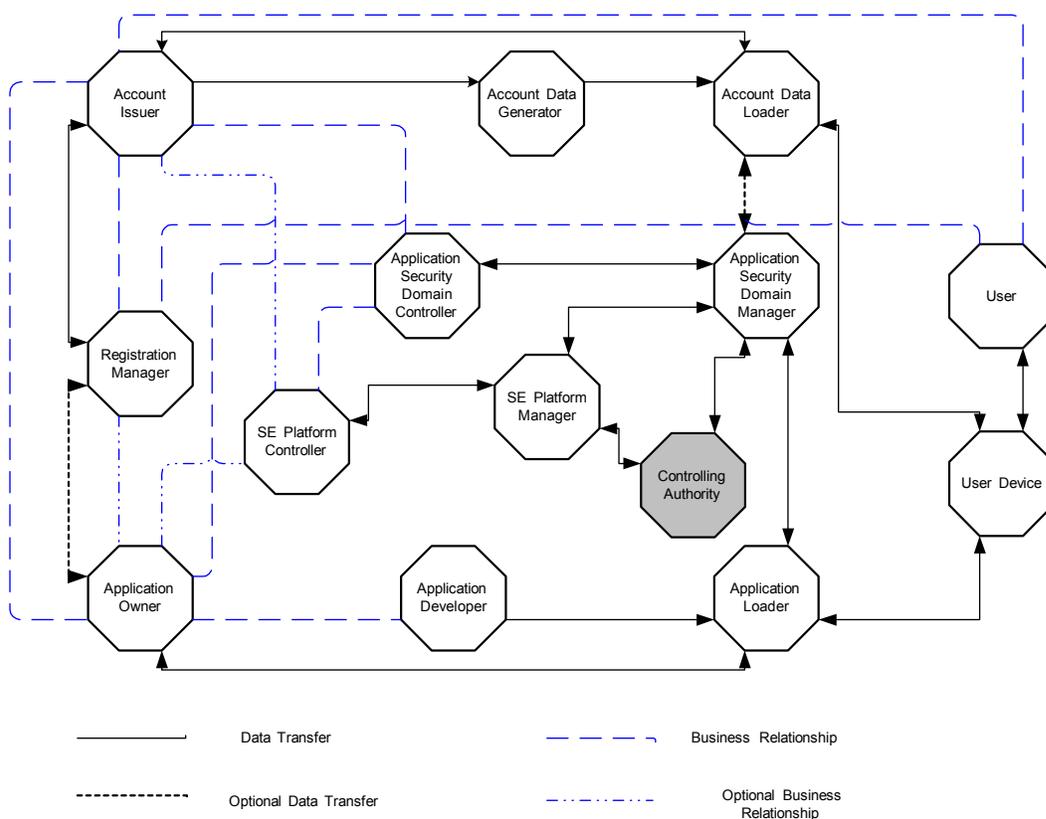


Figure 7 Controlling Authority Model

2.5 Key Management

The models presented above define the roles of controllers and managers. Controllers have logical control over what applications and/or data is installed into the secure element. The managers perform the operations necessary to carry out the instructions of the controllers.

The assumption is that the secure element is secured by cryptographic means, and thus the managers perform the cryptographic operations required for management of the SE. In order for the managers to perform these functions they must have knowledge of the cryptographic keys stored in the SE for management purposes. This section discusses how the keys may be distributed to the appropriate entities whilst ensuring the required levels of separation between parties.

Two models are presented, a distributed model and a centralized model. In the following models the discussion concentrates on the keys used to allow control of the platform, in particular loading and management of applications. There may also be other sets of keys for encryption which also need to be managed, however for simplicity at this point these are not discussed.

2.5.1 Parties

The following parties are assumed to be a part of the distribution and control chain for a secure element.

- The *silicon manufacturer*, who manufactures the physical silicon for the SE.
- The *secure element vendor*, who packages the silicon and sells it as a secure element
- The *mobile device manufacturer*, who (for embedded secure elements) embeds the secure element in the device
- The *SE platform manager*, who manages the SE platform as discussed above
- The *Application security domain manager*, who manages an application security domain as discussed above

As will be elaborated below, not all parties are always required. In a particular instance there may be more parties, however for the purposes of this discussion this classification serves to illustrate the issues surround key distribution and management.

2.5.2 Distributed Model

The distributed model involves the management keys for the SE cascading from one party to the next in the distribution and control chain. This is

illustrated in Figure 8. Although the example assumes symmetric keys are used, the model may also accommodate asymmetric keys.

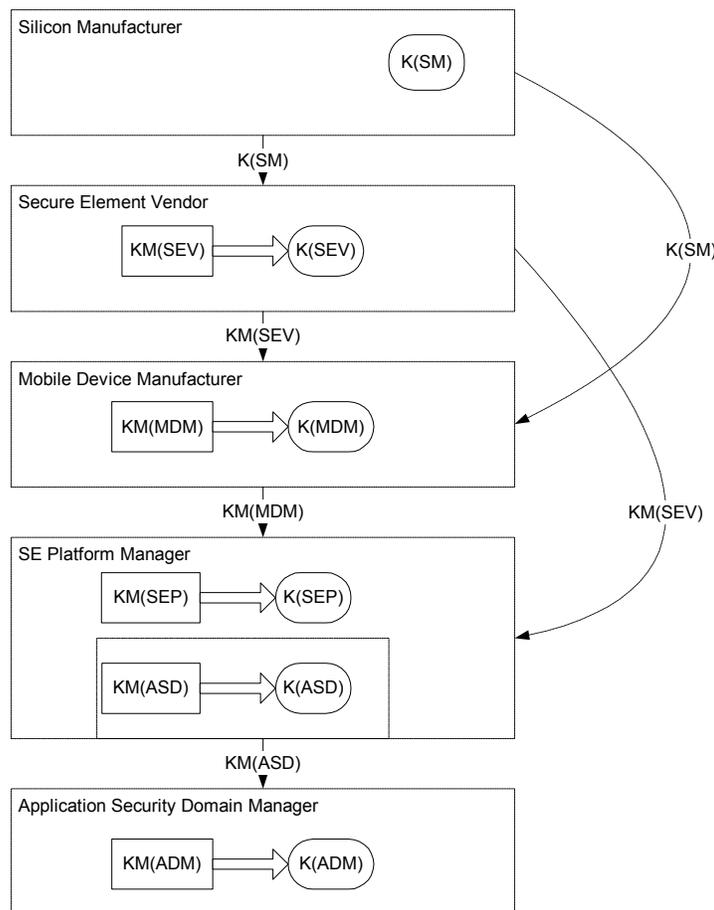


Figure 8 Flow of keys for distributed key management

The Silicon manufacturer produces the silicon, at which time each of the chips has the same key, $K(SM)$.²

The chips are provided to the SE Vendor along with $K(SM)$. The SE vendor changes the management key to $K(SEV)$, generally by diversifying³ the keys from a master key, $KM(SEV)$. At this point the change in keys means that the silicon manufacturer can no longer manage the SE.

The SE Vendor provides the SE to the mobile device manufacturer for embedding in the mobile device (for removable elements this step may be

² The value of $K(SM)$ is usually constant for a batch of chips. It may be varied between batches, and may also be varied for different customers.

³ Diversification of symmetric keys is a process by which individual keys are generated by encrypting a chip specific value with a master key. The chip specific value is typically a serial number (such as the ICC_ID). The use of key diversification allows all individual keys to be regenerated from the master key, without the need to store and handle a large number of individual keys.

skipped). The SE vendor also provides $KM(SEV)$ ⁴ to the mobile device manufacturer. This allows the mobile device manufacturer to change the keys to $K(MDM)$, again diversified from $KM(MDM)$.

When the devices are shipped, $KM(MDM)$ will be supplied to the SE platform manager for the devices⁵.

The SE Platform Manager will change the keys to $K(SEP)$, providing separation from the SE Vendor or mobile device manufacturer.

When an application security domain is set up by the SE platform manager, it will have an initial management key, $K(ASD)$. The SE platform manager provides the corresponding $KM(ASD)$ to the application security domain controller. The application security domain manager may then change the keys to $K(ADM)$, diversified from $KM(ADM)$ to provide separation from SE platform manager.

Note that as illustrated in Figure 8 the mobile device manufacturer may source the silicon directly from the silicon manufacturer without going through the SE vendor.

Of the models presented in section 2.4 above, it is most applicable to the SE Platform Controller centric model, the Multi-application providers model, the Delegated model and the Application provider centric.

2.5.3 Centralized Key Management Authority Model

In the centralized authority model, key management is performed by a centralized authority. While control is passed from one controller to another, the centralized authority maintains the keys. Secure channels are established between the centralized authority and the controllers, and the centralized authority operates under the orders of the controllers.

The model is illustrated in Figure 9. This model is often applied using asymmetric key management using a public key infrastructure, as this is more amenable to providing confidentiality for applications for the different parties; however, it is possible to establish such a model using symmetric keys.

⁴ Note that the SE Vendor will use a different $KM(SEV)$ for each mobile manufacturer to whom the SEs are sold. This provides separation between the different manufacturers.

⁵ Typically the mobile device manufacturer will use a different value of $KM(MDM)$ for each SE platform manager.

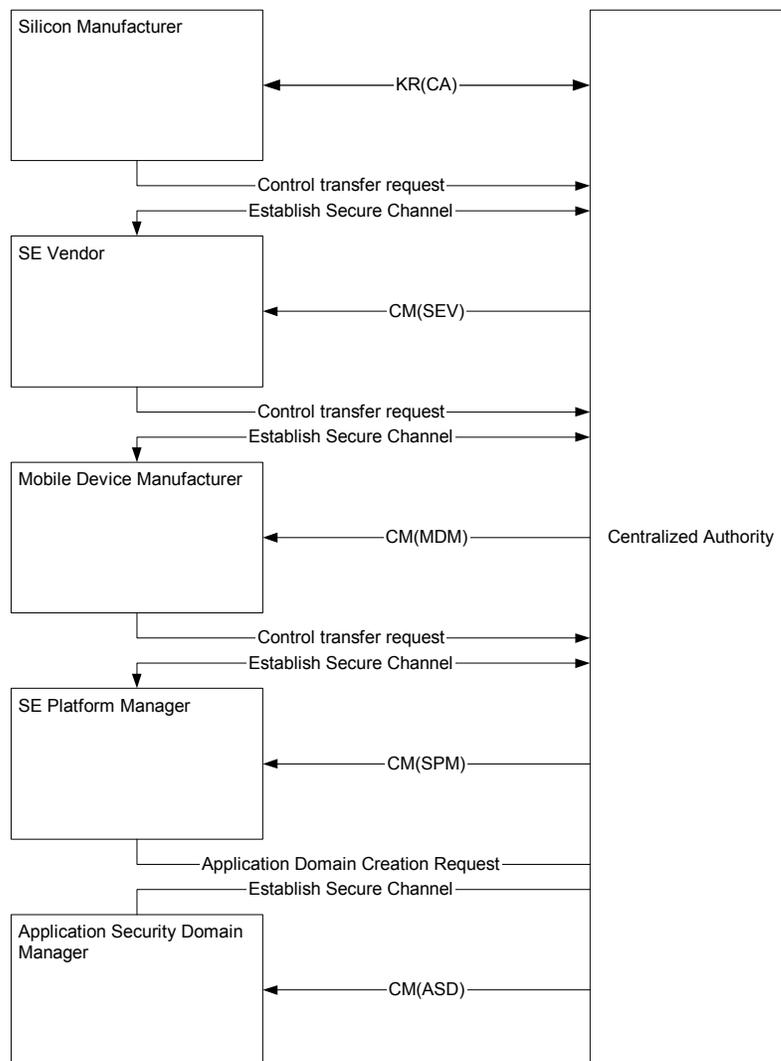


Figure 9 Centralized Authority Distribution Model

In this model, at the time the chip is manufactured a management root key, $KR(CA)$, is embedded in the device, with the corresponding key held by the centralized authority⁶.

When the chip is sold to an SE Vendor, the silicon manufacturer informs the Centralized Authority, who establishes a secure channel for communication with the SE Vendor. When the SE Vendor wishes to perform a management function, it sends a request to the Centralized authority which issues a management certificate, $CM(SEV)$, to allow the SE Vendor to perform the function.

A similar process is undertaken to allow transfer to the Mobile Device Manufacturer and the SE Platform Manager.

⁶ Note that this may be done either by the Centralized Authority providing the key to the silicon manufacturer, or by the chip generating the key and this being registered with the Centralised Authority.

When the SE Platform Manager wishes to establish an application security domain, the SE Platform manager requests the Centralised Authority to establish the security domain, providing the Centralised Authority with the parameters of usage for the application security domain. The Centralised Authority establishes a secure channel with the application security domain manager. The Centralised authority will issue a management certificate, *CM(ASD)*, to the application security domain manager to allow the application security domain manager to perform management functions which have been allowed by the SE platform manager.

This model corresponds to the Controlling authority model outlined in section 2.4 above.

2.6 Provisioning Security Requirements

This section provides a number of security requirements for mobile proximity payment related to provisioning and use of payment credentials and information. It is not an exhaustive list of requirements.

Overall, this document concentrates on the capabilities of a mobile device being used for proximity payment, and assumes that the point of sale accepting infrastructure meets all industry and other appropriate standards and certifications.

The security requirements for mobile payment can be classified into two areas:

- Protection of Payment Assets
- Protection of Transaction Data

2.6.1 Protection of Payment Assets

2.6.1.1 Description

Payment assets are assets particular to the mobile proximity payment application. These include the payment application code and the payment credentials, including such things as account numbers, user information (such as name), and payment secret keys.

The payment assets need to be protected throughout the lifecycle of the application. For the purposes of discussion, this is broken down into a provisioning phase and an active phase⁷.

2.6.1.2 Requirements

⁷ A third phase may consist of a decommissioning phase – this document does not discuss this so it is not included, however it may be useful to include this in the Issues and Recommendations document.

Protection of Payment Assets during provisioning: Payment assets must be protected for confidentiality and integrity during transit between the provisioning server and the security element.

Payment assets must be protected by keys which reside in the secure element.

Protection of Payment Assets during active phase: The protection of the payment assets during the active phase is largely the responsibility of the secure element. However the secure element must be able to interact with the mobile platform, and as such certain requirements fall on the mobile platform.

The mobile platform should not allow the entry of PINs for mobile proximity payment applications to be eavesdropped by other applications residing in the mobile device.

The mobile platform should prevent attacks on the secure element from applications running in the application environment on the mobile device.

2.6.2 Protection of Transaction Data

2.6.2.1 Description

The transaction credentials are by nature not secret as they are exchanged between the payment application and the POS equipment in order to complete a transaction. There is the potential for eavesdropping when transmitted over a proximity interface, and the design of any proximity payment system must take this into account.

While the transaction credentials are not secret, neither are they considered public information; therefore they should be protected against unauthorised access for both privacy and security reasons. Additionally, as a programmable device with a wide area communication channel, a mobile device offers attack vectors not available in traditional proximity devices which must also be taken into account.

For example, attacks over the proximity channel (such as eavesdropping or generation of transaction data via rogue readers) require the attacker to be in proximity with the device under attack. This provides a degree of protection due to the requirement of proximity, and limits the scope of any attack to areas in which the attacker may install attack equipment. In contrast, on a mobile device it may be possible to install attack software via the wide-area channel which mimics the proximity

attacks, thus removing the need for the attacker to be in proximity with the device under attack, and also widening the scope of the attack to any device which may be infected with this malicious software.

2.6.2.2 Requirements

The following security requirements have been formulated in order to mitigate this issue:

- Applications running in the mobile platform or secure element should not be able to monitor transaction data generated by the payment application unless explicitly allowed by the payment application.
- The payment application should have exclusive access to the proximity modem during a mobile proximity payment transaction.
- Transaction data should only be generated by a valid request, that is, via the proximity modem⁸.

2.7 Provisioning Security Issues

This section details the security issues related to the discovery, installation and management of payment application(s) and credentials into the mobile device. Additional issues related to the provisioning process will be addressed in subsequent sections of this document.

The ecosystem necessary to securely support proximity payment will include a variety of functional roles, each of which is critical for proximity payment using mobile devices to be broadly deployed. Examples of these roles include: secure payment application development, payment application and credentials provisioning on the mobile devices, management of the keys used to secure the payment and authorization credentials, and over-the-air service provisioning and maintaining the payment application(s) and credentials.

As this ecosystem is defined, determinations will need to be made regarding who plays each of these functional roles, how the interfaces between each role are established and maintained, and the mechanisms for securely exchanging the necessary data between ecosystem partners.

2.7.1 SE Platform Controller and Manager

2.7.1.1 Description

⁸ This may be performed by the mobile platform preventing the generation of transaction request from any other source, or by providing sufficient information about the source of a request to the payment application such that it can determine the validity of any request.

The issues of key management and distribution have been discussed in section 2.5. The roles of SE platform controller and SE platform manager play an important part in the security of the system. Some of the security issues which need to be considered include:

- Which business entities play the roles of SE platform manager and controller? What are their security policies and data management practices?
- What are the policies of the SE platform controller with respect to separation of applications from different application owners, for example, do they allow separate application security domains.
- What process will be used to manage these keys throughout their lifecycle, include replacement of outdated keys, and the revocation of keys for payment credentials that are no longer valid (expired, lost, stolen, etc)?

2.7.1.2 Recommendation

The distribution of roles is a business decision, and is likely to vary from market to market, and indeed may vary between competitors in a single market. However if there is a great variance in models, security policies and key management practices it will lead to fragmentation in the market, and create barriers to widespread deployment of mobile proximity payment.

It is recommended that the industry discuss different models which may be deployed in order to understand any technical requirements to reduce the dangers of fragmentation.

2.7.2 User Registration

2.7.2.1 Description

One of the key security issues to be addressed prior to enabling proximity payment on a mobile device is the need and methods for registering the user prior to installing the payment application(s) and credentials, and providing the user with the necessary authorizations to effect a secure installation. For traditional payment cards, a card is usually mailed to the address that the user has on file with the account issuer. Once received, the card is activated either via a telephone call from the users home telephone, or through some other authentication method which assures the issuer that the card has been received by the proper account holder.

If a token will be required for personalization, it is recommended that it is generated and delivered to the user during the registration process. This will be discussed further in Section 2.8.1

The authentication process could allow for Single-Sign-On (SSO) to enable subscribers to access multiple applications using SSO supported by strong authentication described earlier, either using a digital certificate, One-Time-Password or biometric authentication schemes.

The process should be able to record all significant authentication events to provide an audit trail.

2.7.2.3 Remaining Considerations

Additional questions which must be addressed include determining what types of information must be used in order to properly establish the identity of the user and how to link the mobile device to be provisioned with the account identity of the user.

2.7.3 Authenticity / Integrity of the Payment Application / Credentials

2.7.3.1 Description

To provision a mobile device for proximity payment, the proximity payment application and the credentials must be installed in the mobile device. It should be noted that the proximity payment application may consist of two pieces, one a secure application which resides in the secure element, and the other an application which resides in the mobile device which provides a user interface. Depending on the implementation, one or both parts of the application may need to be installed via an over the air mechanism. The authenticity of the payment application must be verified prior to installation, to ensure that it is a valid payment application, and not a piece of malicious code attempting to infect the mobile device. The integrity of the payment application must also be verified prior to installation, to ensure that there were no data errors during the OTA transfer.

Likewise, when the a set of payment credentials is ready to be downloaded and installed in a mobile device, the authenticity of the payment application must be verified, to be sure that it is a valid payment application from the proper source. If the application is not already on the mobile device, then the appropriate application should be installed prior to installing the credentials. If the existing payment application is found to

be invalid, then a process must be in place to initiate the delivery and installation of the correct payment application.

Once the payment application is verified and the payment credentials are delivered, they too must be verified to ensure that the data was not corrupted during transfer, so that they will be valid for use with the payment application.

The personalization system must also ensure that the personalization information will only be downloaded to a valid payment application. It must not be downloaded to a rogue application masquerading as a payment application.

2.7.3.2 Recommendations

A process must be identified for verifying the validity of the payment application to be installed.

It is recommended that for applications which reside in the secure element the keys for managing the SE be used for authenticity and integrity of the application during installation. Thus validating the authenticity of the application will be the responsibility of the platform manager or the application security domain manager.

The code of applications which reside in the handset should be signed. If certain privileges are required for operation, then the owner of those privileges will need to sign the code.

In any case, the signer of the code should provide information to the user regarding the source of the application.

The authenticity and integrity of the payment credentials must be verified. The verification process may make use of the keys for managing the SE, or may be verified by the payment application itself, rather than by the device itself. If verification is performed by the payment application, then this process may also be leveraged to ensure that the payment credentials are available only to a valid payment application (for example by encrypting the credentials with an application specific key).

2.8 Provisioning Process Issues

This section details the issues related to the process of managing the lifecycle of the payment credentials, apart from those directly related to security.

2.8.1 Payment Application / Credentials Installation Process

2.8.1.1 Description

2.8.1.2 Recommendations

Overall, the interfaces to be used for proximity payment will depend largely upon which model is used, and the actors involved. While the proximity payment ecosystem could benefit from standardization of these interfaces to support a scalable, interoperable solution, this cannot be done until it is clear what model(s) will be adopted, and who the actors will be.

To address the issue of which medium to use, the preferred approach for provisioning a mobile device for proximity payment is to do so via the wide-area cellular network. This should allow for the most flexible, scalable solution for mobile device provisioning, while maintaining the lowest overall system cost.

An OTA personalization process and protocol also needs to be defined, ideally based on current industry standards. If the SIM card is used as the S.E., GSM 03.48 could be used. Other possible standards include Global Platform secure channels, or Bearer Independent Protocol (BIP), which is based on an end-to-end security model. This is one area of standardization which needs to be further explored by the MPF.

The process needs to provide the triggers for any required user interaction to perform the personalization process (e.g. informing the user to go to a particular URL, requesting the user to enter authentication information etc). If a time delay is expected between the registration of the mobile device and the personalization of that device, then some sort of token should be used to assist in authentication during the personalization process.

This process needs to provide an audit trail, to enable any required certifications.

The user should be informed of when the payment application and credentials are ready for use.

2.8.2 Update of Payment Application and Credentials

2.8.2.1 Description

As with traditional credit cards, the payment instruments installed in mobile devices for proximity payment will need to be updated as their expiration date nears.

For payment information updates, the process may be made transparent to the user, with the account issuer delivering updated credentials and/or payment application(s) to the device

as needed. As with the initial installation, the account issuer may or may not indicate to the user that the payment information has been updated, although doing so in a manner similar to the process used for updating traditional credit cards could be used.

The process that is defined for updating the payment application(s) and credentials, whether it is to delete the existing payment information and reinstall new information, or to simply overwrite the existing information, will need to ensure that the correct link is made between the old account details and the updated account details.

It is also desirable to have a mechanism to advertise to the user's mobile device that a download of updated payment credentials is available. Such a notification could take the form of a message to the user requesting that a download be initiated, and providing the necessary instructions for doing so, or it could be a message which could trigger an existing payment application to initiate a download automatically

It may also be possible for the user to query their bank via their mobile device, to determine if such a download is available. The level of authentication necessary to enable a user to initiate an unadvertised download of payment information will need to be determined.

2.8.2.2 Recommendations

Push changes should use well known push mechanisms such as OMA Push.

If the user is not going to see a change in the payment functionality on their mobile device, then notifying the user that such a change is happening is not necessary, as it could confuse the user more than anything else. However, if the user is going to notice a change in payment functionality, then it will be best to notify the user about the change, to avoid customer service calls; notification of the change does not mean the user should have approval over making the change.

Depending on the situation the user may be asked to authenticate; however, for critical updates this may need to be done without confirmation from the user.

The underlying security system should not allow updates from unauthorized sources.

2.8.3 Payment Application and Credentials Deletion

2.8.3.1 Description

There needs to be a mechanism for deleting the payment credentials, and possibly even the payment application, from the mobile device remotely. Such a deletion might be initiated by the user (“cutting up a credit card”), or it may be a revocation initiated by the account issuer.

Deletion of the payment information is an important aspect in the lifecycle of a payment instrument. There needs to be mechanism for the account issuer to remotely revoke the payment credentials, and even delete the complete payment application, if needed. This “push deletion” could be invoked for a variety of reasons, including the mobile device being reported lost or stolen, or due the payment history of the account holder.

Whatever the reason, a mechanism for push deletion needs to be defined, to allow the payment instrument to be deleted by the account issuer as needed. However, the requirements for user interaction during a push deletion need to be considered. If the deletion is because the phone was lost or stolen, then requiring user interaction defeats this as a security measure. On the other hand, deletion without informing the user may also cause confusion.

There may also be a desire to provide the user with a mechanism for removing their payment credentials from the mobile device if so desired. Such a mechanism would need to provide the user with the means to select the specific payment credentials that they want to delete, and to even delete the payment application in its entirety, if the user no longer wants to have access to the proximity payment functionality.

On the other hand, providing the user with a mechanism to delete their payment credentials or application could result in accidental deletion, which could certainly lead to customer service issues. In order to get payment credentials restored which have been accidentally deleted, it may be unclear to the user whether they need to call their account issuer, or their mobile network operator. The level of visibility that a mobile operator has into what applications the user has on their device can vary on the model that is in use for allowing downloads. For example, some operators allow the user to download any application, while others allow it only through their portal. So support from the mobile network operator for reinstalled a deleted payment application may be limited.

An alternative to allowing the user to delete their payment credentials is to provide the user with the necessary contact information to request the deletion of the credentials, and then to have the deletion initiated by the account issuer, or other responsible party.

2.8.3.2 Recommendations

A mechanism to delete payment credentials needs to be present.

It is recommended that this be performed over the air via the personalization and lifecycle management system. There may be interfaces provided from the payment application to initiate this deletion, either by the account issuer, or by the user.

If the application cannot be deleted over the air, then a mechanism should be included to allow the user to perform⁹ the deletion of the payment credentials, and the payment application, if so desired. If the application cannot be deleted, then the payment application should provide a mechanism to disable itself.

Care must be taken to prevent the accidental deletion of the application or credentials. In particular, if multiple account issuers may personalize a single application, then extra care must be taken to ensure the application is not deleted. For example, the application may not be able to be deleted unless all credentials have been deleted first.

2.8.4 Subscription Change

2.8.4.1 Description

A subscription change could occur for several reasons. In some cases, the user will have changed mobile provider, and will not be returning to the old account. Alternatively, the user may have multiple mobile subscriptions, and routinely move the mobile device between accounts.

If the secure element is the SIM, then a change of mobile subscription would also require a change of the secure element.

If the secure element is not the SIM, then issues to be addressed include:

- Addressability (mobile identification may change)

⁹ This mechanism may require the user to notify the payment account issuer, in order to obtain the required information to perform the deletion.

- Platform management (installing/updating needed applications)

Depending upon the use case, such a change may be an indication that the mobile device has been stolen, or transferred to a different user.

2.8.4.2 Recommendations

There needs to be a mechanism for the payment application to determine what mobile subscription is being used, in order for the payment application to be enabled only when the registered subscription is being used, and otherwise disabled if the mobile subscription details change, according to the business rules of that application.

The account issuer should consider disabling the payment application upon the detection of a change of the mobile subscription, unless there is a valid reason not to do so.

2.8.5 Device Change with Embedded Secure Element

2.8.5.1 Description

A device change could occur for a variety of reasons, including the user upgrading to a new mobile device, or replacing a device which has been lost or stolen. In some cases, this change could also include a change of mobile subscription, as detailed above.

Such a change could impact not only the mobile device user, but also others, including the existing (and possibly new) mobile network operator, and any relevant 3rd parties, such as one or more account issuers, and even a back-up service provider (if one is present).

If the secure element is embedded in the mobile device, then such a change may require the entire personalisation process to be repeated for the new mobile device, including both the payment application, and the payment credentials issued to the device user. This would follow the personalisation process described in Section 2.8.

Under ideal circumstances, any information related to the payment credentials will be deleted from a mobile device before any attempt is made to reinstall them on a replacement mobile device. Sometimes this may not be possible, particularly if the existing mobile device has been lost or stolen, or the user has deactivated the existing mobile device before the issuer is notified of the device change. In these

instances, the existing device may not be on the network, so it may not be possible for the payment credentials residing in this device to be deactivated.

Therefore, there should be a mechanism for deactivating any payment credentials through the issuer, to ensure the credentials will not be used any further. This would require that the payment instrument information that is issued to the new device should be distinguishable from the payment instrument information issued to the old device.

When credentials are to be reissued and installed on a new mobile device, it is important that the re-issuance occurs in a secure manner, and may only be performed by an authorized party (that is, the issuer or the issuer's agent). The issuer must be properly authenticated, and confidentiality and integrity of the new data must be ensured (as with the initial issuance).

For this type of device change, the mobile operator will be in the best position to manage such a process, particularly if multiple accounts on a single device are affected.

2.8.5.2 Recommendations

The payment instrument information should be unique to each installation on a device, so that payment credentials from the same account but installed on different mobile devices can be distinguished. This should be true for both the re-issuance of credentials in the case of a device change, or with updates to valid credentials in the same device.

2.8.6 Device Change with Removable Secure Element

2.8.6.1 Description

Mobile devices utilizing a removable secure element are also subject to routine replacement/upgrading, but because the secure element is removable, in some instances it can be transferred by the user to the new mobile device, assisting in the transition.

Of course, if a device is lost, the situation would be similar to a device change with a non-removable secure element, e.g. loss of both the mobile device and the secure element. Likewise, if the device change is due to a change in the user's mobile subscription, then the situation would be as detailed in Section 2.8.4. This section discusses the situation where the mobile device itself is being changed, and the secure element is being transferred to the new device, but the user's mobile subscription is not changing.

The advantage of having a removable secure element is that a complete repersonalization of the mobile device may not be required when a device change is made. However, any time the secure element is transferred from one device to another, the compatibility of the new device will need to be verified.

Compatibility includes not only support for the APIs involved, on both the secure element side and the mobile device side, but also support for the correct MMI, to ensure the applications still function, as well as the form factor of the secure element itself. If any of these critical components is not compatible, then the secure element will also need to be replaced.

However, if the secure element from the existing mobile device is compatible with the new mobile device, then the secure element can be transferred. If this is desirable, there will need to be a mechanism for the payment application to determine if all of the necessary components are installed, and if not to download and install them, after proper authentication and authorization, so that the stored payment credentials can be used.

2.8.6.2 Recommendations

When a removable secure element is being used, there must be a mechanism in place to authenticate the user when the secure element is installed in a new device, prior to enabling the payment application and credentials on the new device. This is needed to ensure that a stolen secure element cannot be installed and used in a new device by an unauthorized user.

2.8.7 Backup / Restore of Payment Application and Credentials

2.8.7.1 Description

It may be desirable for a trusted 3rd party, possibly the mobile operator or some other entity, to offer as a service to the user a network back-up of the contents of their digital wallet – basically, a complete set of their currently installed payment applications and credentials. The provisioning logical architecture, including the necessary interfaces for a backup/restore function, is shown in Figure 10.

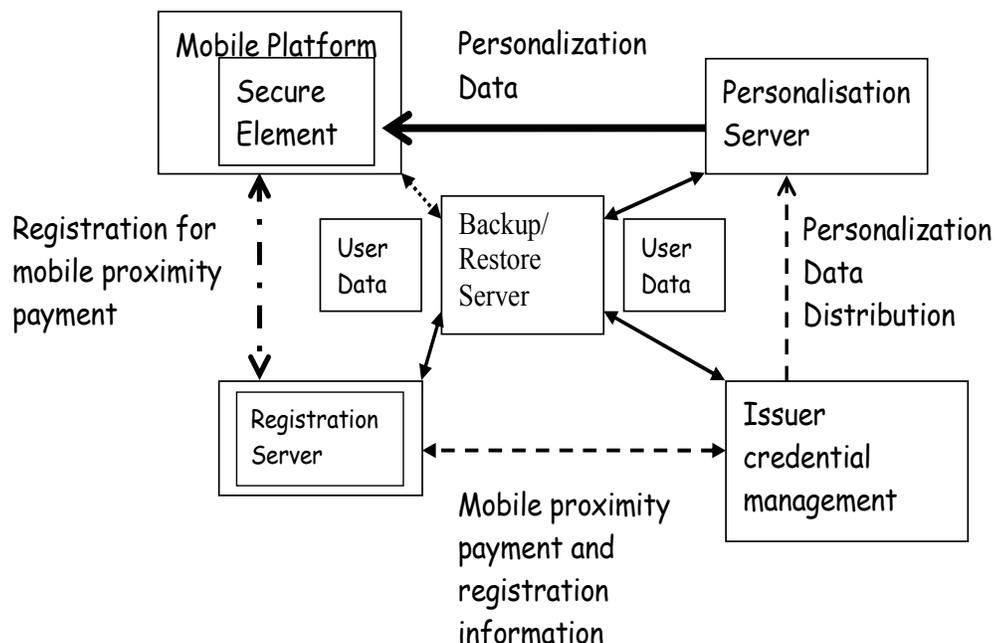


Figure 10: Backup/Restore Function in Logical Provisioning Architecture

If a mobile device is being replaced, either due to the loss of the old device, or simply because the user is upgrading to a newer model, this Personalization Back-Up Service provider could then, following the appropriate authentication of the user, reinstall the existing set of payment credentials, along with any necessary payment applications, on the user’s new mobile device.

The data maintained by the Back-Up Service provider may include other information, including personalization preferences/setting, which require a direct link to the mobile device.

Of course, it is important that the re-issuance occurs in a secure manner, and is performed by an authorized party (that is, the issuer or the issuer’s authorized agent). This agent must be able to ensure the authentication of the issuer, and the confidentiality and integrity of the new data (as for the initial issuance). There is also a need to ensure that the correct link is made between the old account details and updated account details.

If the Personalization Back-Up Service provider does not also act as the primary OTA personalization entity for the mobile device in question, there must be a mechanism for the back-up service to acquire and maintain an accurate set of the user's payment credentials; this must either be done by contacting the issuing entity as each credential is issued to the user; or by a mechanism whereby the personalization back-up service provider can read the credentials from the user's device, in order to maintain a current image of all of the existing payment applications/credentials.

In order to insure that all of the credentials to be restored to the user's device are valid and up-to-date, the user may need to be identified to a number of different parties – the application provider, the account issuer, the mobile network operator, the back up service provider and potentially other parties. It needs to be considered whether a standard identity may be used for each of these, or if the back up service provider must maintain a mapping of identities for each of the different parties.

The entities which may provide the personalization back up service will need to be determined. Possibilities could include the mobile operator, the handset manufacturer, or some other third party service provider.

2.8.7.2 Recommendations

There will be some credentials that cannot be restored directly from a backup, since some payment credentials will need to be updated by the issuing bank prior to being reinstalled in a new device. These credentials, even if backed up by the backup service provider, will need to be refreshed by the account issuer, to ensure that the users account information reflects any updated account information.

3. Additional Security Issues

3.1 Introduction

This section addresses issues and recommendations for security issues that are not directly concerned with provisioning/personalization. This includes issues such as platform security, application security, and issues relating to the secure element in a mobile device. Security issues specific to provisioning/personalization are addressed in Chapter 2: Provisioning and Personalization.

3.2 Mobile Device Architecture

3.2.1 Logical Architecture

The logical architecture for the mobile device is shown in Figure 11.

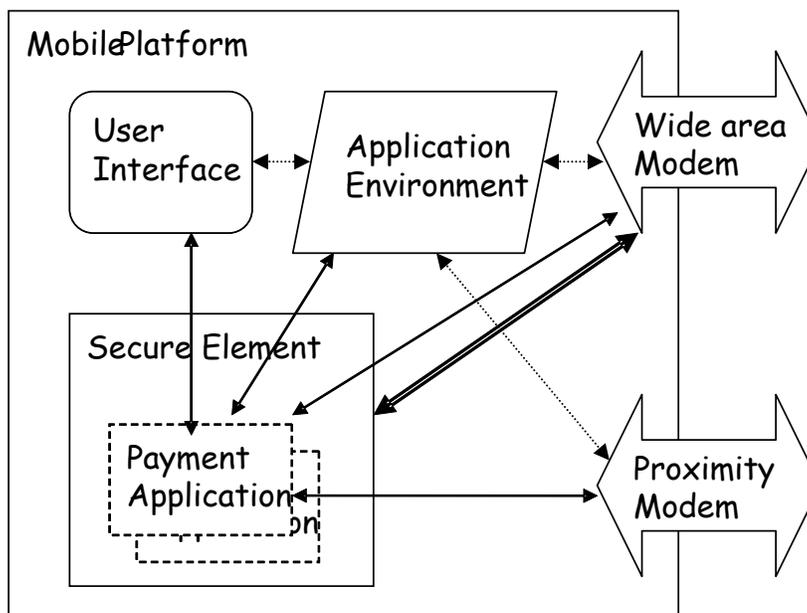


Figure 11: Logical Architecture of a Mobile Device

The mobile device consists of two logical components, a *mobile platform*, and a *secure element*.

The *secure element* is where one or more payment applications are hosted, and provides a secure area for the execution of the applications and protection of the payment assets (e.g. payment data, keys, the application

code). The secure element may also host applications which are not related to payment.

The *mobile platform* is the device in which the secure element is hosted, and also contains other components such as a user interface (UI) for both input and output, an application environment (such as a Java or other virtual machine), a proximity modem, and a wide-area modem. In the context of the Issues and Recommendations document, the wide-area modem is likely to be a cellular modem, however it could also be a wired modem, or wireless LAN.

The secure element and the mobile platform provide logical interfaces between the payment application and the user interface, between the payment application and the application environment, between the payment application and proximity modem, and between the payment application and the wide area modem. There is also a logical interface between the wide-area modem and the Secure Element which allows personalization.

In practice, the logical interfaces may be implemented via other elements within the mobile platform. For example, the payment application to UI interface may be implemented via a controlling application present in the application environment on the mobile platform. This situation is illustrated in Figure 12 where the interfaces between the payment application and the user interface and payment application and wide area modem are established via applications running in the application environment.

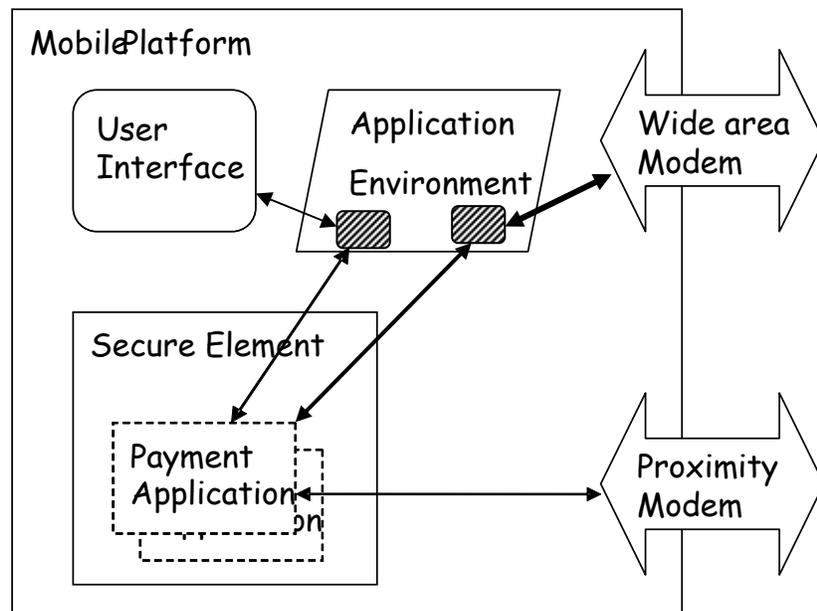


Figure 12 Example implementation of logical interfaces

3.2.2 Implementation Options for the Secure Element

Four implementation options for the secure element have been identified at this point. It is possible that additional implementation options will be available to device implementers in the future; if so, these additional implementations will need to provide appropriate security services for supporting proximity payment.

3.2.2.1 Embedded Hardware Secure Element

The secure element may be a hardware tamper resistant module which is embedded in the mobile device. For example this may be a component such as a smartcard chip which is soldered on the circuit board of the mobile platform.

3.2.2.2 Secure Element on USIM

The secure element may be in the USIM (or SIM or RUIM), the smartcard used for identification for mobile telephony purposes.

3.2.2.3 Removable Hardware Secure Element

The secure element may be a removable hardware element, such as smartcard, or a secure core on a multimedia card. In many ways this is similar to the removable USIM option described above, however it is not associated with the particular mobile subscription, nor does it need to support telephony services for the mobile device.

3.2.2.4 Secure Element in Mobile Device Baseband Processor

The secure element may be hosted as part of the baseband processor (that is, the multipurpose processor which powers the mobile device) using portions of secure memory and processing. This is potentially a longer term solution.

3.2.3 Secure Element Security Level

The environment (both hardware and software) into which a payment application can be downloaded and run will have a significant impact on the overall user experience. The most satisfying user experience will likely be achieved through a combination of a mobile device's inherent capabilities (its MMI, its friendly usage, and its enhanced services allowed by collateral applications), and a secure element for providing the necessary security services.

This distribution of functionality between the secure element and the mobile platform will place security requirements on both the secure element and the mobile platform. This section addresses the requirements for the secure element.

3.2.3.1 Description

The secure element is responsible for protecting the assets of the payment application. These include the application code, any application cryptographic keys and other confidential payment related information, both in persistent storage and also at runtime.

The secure element must have a tamper resistant application environment which provides a demonstrably secure environment that enables applications to be run without the risk of being intercepted by another application. This would include:

- mechanisms for keeping applications separate;
- protecting different areas of memory from being modified by unauthorized applications;
- prevention of modification of application code;
- secure access control to application persistent storage

3.2.3.2 Recommendations

The secure element should be implemented in a manner which may be independently evaluated to demonstrate a given level of security.

Typically payment applications expect a rating of very high security, for example, EAL4+.

3.2.4 Secure Element Support for Post-Distribution Personalization

3.2.4.1 Description

The distribution model for mobile proximity payment applications and personalization data is likely to be one where the applications and data are loaded to the mobile device after the device is in the hands of the user.

As such the secure element application environment must support the secure installation and personalization and lifecycle management of applications in the field. The management of such application environments is discussed further in Chapter 2.

3.2.4.2 Recommendations

The secure element application environment should be a well defined environment allowing post-distribution installation and personalization of applications.

As described above, the most satisfying user experience will likely be achieved through a combination of a mobile device's inherent capabilities (its MMI, its friendly usage, and its enhanced services allowed by collateral applications), and a security element for providing the necessary security services.

Many of the security requirements for mobile proximity payment can be pushed to the secure element, however the mobile platform provides a number of key functions. As such there need to be a number of security features in the mobile platform. This section addresses the

3.3.1 Mobile Platform Application Environment Choice

3.3.1.1 Description

An issuer wishing to deploy mobile proximity payment needs to target their customer base. In many cases the mobile devices which are owned by their customers will be diverse. If there is a proliferation of application environments used for the mobile platform this will cause fragmentation, and increase the development and support effort required.

This also includes the APIs offered to control the proximity functionality and the interaction between the mobile platform and the secure element.

3.3.1.2 Recommendations

There are a small number of well-known application environments in use for mobile devices, for example J2ME MIDP, Symbian, Windows Mobile. However the interfaces to a secure element are not well defined, with many manufacturers implementing proprietary interfaces.

In order to minimize the development of the portion of proximity payment applications which need to reside on the mobile platform there needs to be a standardization of these interfaces.

For J2ME it is recommended that this be done along the lines of JSR 177, but standardizing the way in which JSR177 being used.

3.3.2 Mobile Platform Security Level

3.3.2.1 Description

The mobile platform needs to provide a level of security to prevent an attacker being able to download malicious code

which could compromise the proximity payment application – this includes both the portion residing the secure element as well as the parts of the payment application residing in the mobile device.

The level of security required is not at the same level as that expected of the secure element. As mobile devices are primarily personal devices, and generally kept in a user’s possession, the need to protect against physical attacks is somewhat reduced. The mobile platform primarily needs to protect against attacks from other applications which reside in the device, in particular those which may be loaded onto the device.

3.3.2.2 Recommendations

The mobile platform should provide a trusted, provable environment that enables applications to be run without the risk of being intercepted by another application. This would include mechanisms for keeping applications separate, and for protecting different areas of memory from being modified by unauthorized applications.

The link between the secure element and an application running on the mobile platform should be secured so that no other applications may eavesdrop or inject traffic on the link.

The mobile platform should provide mechanisms for verifying the code that it is running in order to prevent against malicious tampering with the operating system and core software.

The mobile platform should provide immunity (either through hardware or software) to virus-type attacks of the mobile platform’s stored data.

Compliance with the Small Terminal Interoperability Platform (STIP) specification may be appropriate for mobile devices that may be used as payment terminals. STIP defines a Java-based payment terminal solution for use on resource-limited devices, such as mobile devices. The STIP specifications are under the control of the Global Platform Consortium.

The business case for hardware assisted security within terminals is unclear. Terminal manufacturers will not necessarily accept the concept of a trusted device.

Keys and certificates for verifying signed code should be protected against modification.

In addition, the mobile platform should support privacy to enable subscribers to access and pay for services from 3rd

parties and communities that the subscriber may want to access without having to disclose personal information

3.3.2.3 Remaining Considerations

The mobile platform application environment should be controlled and managed by a unique responsible entity to ensure trusted operation. However, it is possible that certain rights related to the security and/or functionality of the mobile platform may be delegated to other entities, such as the mobile network operator. The nature of, and process for, delegating and managing such rights must be determined.

3.3.3 Access to the Secure Element

3.3.3.1 Description

The secure element protects the payment credentials, and in particular payment application keys and other secret data. However in order for the proximity payment application to operate, it must generate and release transaction data, and respond to PINs and so on.

If arbitrary applications residing on the mobile platform are able to access the secure element, then these may launch attacks on the payment application in the secure element. For example, an application could request the generation of payment transaction credentials, which are then sent from the device to be used in another place, or an application could launch a denial of service attack by sending PIN verification commands to the proximity payment application until the PIN is locked.

3.3.3.2 Recommendations

It is recommended that access to the secure element is restricted to privileged applications. For example, the APIs for communication with the secure element may be available only to applications which have been signed with specific keys.

3.3.4 Access to Proximity Modem

3.3.4.1 Description

It is very possible that mobile devices with an integrated proximity modem could also have a variety of other modems integrated as well. For example, a single device could include the wide-area cellular modem, the proximity modem, and a PAN or WLAN modem, such as a Bluetooth or IEEE 802.11 modem.

Since different applications will have access to these modems, it could be possible for a rogue application to mimic use of the proximity modem, in order to get access to payment credentials. Some method of controlling access to each modem could be implemented, but this could be difficult to enforce in a multi-application environment. Alternatively, mechanisms might be put in place to allow the verification of which modem interface is being used by an application, if feasible.

3.3.4.2 Recommendations

Applications executing in the Secure Element should be able to distinguish which interface is being used to communicate with the application, to limit spoofing of the proximity payment modem.

Restrictions should be placed on which applications are able to communicate with the secure element (for example using the J2ME domains).

It may be appropriate to restrict the access to the proximity modem to certain classes of applications in the mobile device.

3.4 Secure Element Issues

A secure, tamper-resistant element is needed to enable the execution of secure applications such as required for proximity payment. Some options for the secure element raise issues specific to their use. These issues are detailed below.

3.4.1 SIM/USIM

3.4.1.1 Description

The SIM/USIM card is one good candidate for use as the secure element for several reasons, including:

- SIM/USIM cards have been designed to be secure against logical and physical attacks, making the SIM/USIM card a tamper-resistant component.
- Thanks to its small size and small OS, the application code can be proven secure.
- Applications can be maintained under the control of a trusted actor; in the case of the SIM/USIM, this would be the mobile network operator (MNO).
- Several mechanisms, such as the use of a PIN code, already exist in the SIM/USIM to allow application management or security mechanisms.

On the other hand, the SIM/USIM must be able to perform its telephony functions in addition to any functions related to proximity payment, and the real time requirements for communications and mobile payment must not interfere with or degrade each other in order for the SIM/USIM to function as the secure element for mobile payment.

3.4.1.2 Recommendations

Different standards and specifications already exist to ensure that isolation between the proximity payment and telephony functions is maintained. One or more of these standards (depending upon the implementation) should be followed. These examples include:

- Global Platform (GP), which defines a standard that enables isolation between applications inside the SIM. GP, which documents many issues related to the implementation of Secure Elements, specifies an open environment on the card with:
 - Protected Domains: a mix of personalized and highly-secure application-based services.
 - An authority that controls the card: allows the sharing of portions of the card with partners.
 - Secure Application Manager: allows the downloading of new cardlets, then new (or updated) payment applications remotely.

3.4.1.3 Remaining Considerations

The SIM/USIM card is clearly one possible solution for the secure element required to allow the execution of secure applications such as that required for proximity payment. However, to do so, the SIM/USIM needs to trust the mobile platform, which means that the security requirements for a mobile handset to allow proximity payment application must still be evaluated and defined. For example, the SIM/USIM payment application must be non-alterable and fully isolated from the others in order to:

- Forbid any external or internal attack (by the user or someone else).
- Prevent any collateral effect from any other SIM/USIM card application (including another payment application).

3.5 Application Security Issues

3.5.1 Isolation of Payment Application

3.5.1.1 Description

The card payment application must be non-alterable and fully isolated from the others in order to prevent external or internal attacks (by the user or not). This is necessary to prevent any collateral effect from another card application (including another payment application), which could result in the unauthorized access of payment credentials.

This isolation could be enforced using hardware security mechanisms, software security mechanisms, or a combination of the two, depending upon the particular platform implementation.

3.5.1.2 Recommendations

It is recommended to make use of application security domains when available to provide further separation of the applications.

The payment application should not rely solely on the support of the SE or mobile platform to provide isolation. The application should be coded defensively to minimize the risks of unauthorized access and modification.

This includes encrypting sensitive data, using checksums to ensure the validity of data and code, deleting any sensitive values from memory after use etc.

3.5.2 PIN Code

3.5.2.1 Description

Use of a PIN code is one security mechanism already being used for authentication in many different applications, including for enabling access to the SIM in GSM mobile devices. Use of a PIN code for securing access to the proximity payment application is a viable solution, although issues regarding whether PIN codes may be shared between applications, or must be specific to an application, will need to be addressed. Some implementations, such as those based on Global Platform, may allow a single PIN to be used for securing multiple applications. This single PIN code could be used to activate the wallet or the specific used credit/debit card. However, this raises support issues, including who issues the PIN, how it is shared between applications, and who must

provide user support if the PIN code is locked due to too many retries, or forgotten by the user.

Having different PIN codes for each application simplifies support (each application issuer supports the PIN related to their application), but then the user must remember and manage a potentially large number of PINs.

3.5.2.2 Recommendations

There should be a certain minimum length defined for a PIN code of at least four digits. A mechanism should also be provided which would not allow the use of obvious Pin codes (e.g. 0000, 1234, etc.).

The PIN code should be locked after a certain number of incorrect attempts to enter the PIN. The same PIN locking mechanisms as those used for credit cards can easily be applied for proximity payment, for example locking the PIN after 3 consecutive incorrect entries.

Because a PIN can be locked inadvertently by the user, there should be a mechanism for unblocking a locked PIN code, with conditions for doing so well-defined.

The order of in which the PIN code retry counter is incremented should be implemented carefully, making sure that the PIN code retry counter is incremented using a sequence which will prevent power cycling attacks to be used to allow unlimited retries.

PIN code entry via the keypad should be hidden from other applications, to prevent eavesdropping by a rogue application.

The wallet or proximity payment application PIN code must be dissociated from the SIM PIN, in order to avoid any side effects. (why and more detail) To separate the telephony lock and payment application lock. This will not prevent the user from using the same value for each PIN; it will just provide separate management mechanisms for each PIN.

3.5.2.3 Remaining Considerations

Issues regarding the use of a single PIN code for multiple applications must also be addressed. Questions regarding the management of multi-application PIN codes will need to be resolved, such as who unlocks the PIN if it is locked; and what is used to identify the user prior to unlocking the PIN.

3.6 Application and Platform Approval Issues

There are two main areas of approval for payment applications and platforms. The first is type approval, where the payment application and device are tested to ensure that the application and device behave correctly in accordance with the specifications to ensure interoperability. The second area of approval is to provide assurance that the application and device offer a sufficient level of security.

Type approval for proximity devices covers two areas:

- Analogue testing to check the RF properties of the device, such as read range.
- Digital testing to check that the application conforms to the specification at the level of data exchange.

Security testing covers both the platform and application environment (usually a smartcard and its operating system) in which the payment application will run, and the application itself. Testing of the platform covers both physical and logical security and gives a level of assurance regarding the tamper resistance of the platform. Testing of the application for security is usually a code review, and checks for defensive coding techniques and that insecure constructs are not used.

In addition, other testing will be required, including usability testing to ensure an acceptable user experience, and intra-handset interoperability testing, to ensure that the payment application does not negatively impact the functionality of other applications on the mobile device.

As part of the product development process for mobile devices which support proximity payment in addition to a variety of other functions, testing of this nature should be included as a natural extension of the manufacturer and/or operator approval process for new devices, rather than as part of the proximity payment development process.

3.6.1 Approval and Approving Bodies

3.6.1.1 Description

Payment associations generally set interoperability requirements, and define a minimum level of security for applications and platforms which carry their brand. Each issuer may also have further requirements on applications and platforms for what they issue.

Generally speaking, a vendor must get a certificate of approval for products from the particular payment association. The testing is typically managed by the payment associations; however, the tests may be carried out in laboratories approved by the associations.

The requirements of an individual issuer are typically managed between the particular issuer and the particular vendor.

In the traditional card world a particular card would carry only one payment brand, and products from only a single issuer, and the issuer would choose the vendors from which the platform and application are sourced.

In the case of payment through a mobile device this is no longer the case. The platform is likely to be chosen by the user, and may hold multiple applications from multiple associations and issuers, as well as applications unrelated to payment using the same equipment. Additionally, at the time of manufacture it is likely that the applications, payment associations and issuers are unknown.

This makes it more difficult to determine what approvals are required and from whom.

3.6.1.2 Recommendations

There will be individual certifications required on an ad-hoc basis. However, to meet the broader needs of the market, a clear understanding across the industry of what is required to achieve certification is needed. Once this is done, cross-certification then becomes possible. However, cross certification should be achieved at an acceptable level based upon the established level agreed to between the respective parties in conjunction with periodic approval.

There is also a need to determine who would perform the certification testing, if done by a 3rd party.

Self-certification may also be possible, and would be desirable from the perspective of the mobile device manufacturer and/or the application developer. However, to make this possible, there will need to be stringent requirements for maintaining the separation of design and testing, to ensure that any self-certification being used is valid.

Some testing will be required of the UI implementation with the application running on the mobile device, to ensure that the specific UI correctly exercises the application API.

Operator approval of the proximity payment application should be consistent and incorporated into each operator's current process to the extent deemed appropriate by the operator. However, it must be understood that not all devices will require operator approval – especially samples and prototypes.

3.6.2 Lifecycle for Mobile Devices

3.6.2.1 Description

The development cycle for a mobile device is considerably different than that of a smartcard. Approval of every new model of mobile device could cause considerable impact on the release process for mobile devices.

3.6.2.2 Recommendations

Mobile devices should be designed in a modular fashion such that the elements of the mobile device that require approval are unchanged between devices based on common platforms.

3.6.2.3 Remaining Considerations

Boundaries need to be defined for elements requiring approval in order to determine when elements have changed sufficiently to require a new approval.

3.6.3 Analogue Approval

3.6.3.1 Description

Analogue type approval tests that the RF properties of a mobile device are within the specifications of the payment scheme. The analogue characteristics may be heavily dependent on the physical properties of the device, such as antenna size and form factor, and indeed the properties of the chip which is driven by the antenna. This last issue may be of less importance in devices which have a power source, as compared with devices where power is derived from the field of the reader.

While standards do specify some of the RF characteristics, these are not always as rigidly defined as is required for interoperability within a given system. Therefore payment schemes (and other contactless systems) may have additional more precise requirements. Given the differing nature of these systems, the requirements may not always align.

Analogue testing is generally performed on the final solution (platform and application) however for a mobile device the application is unknown.

There may be other industry testing in place (for example, the NFC Forum) which may possibly be leveraged if it is sufficiently strict to provide the level of interoperability required for the payment schemes.

3.6.3.2 Recommendations

Particularly for removable secure elements, the mobile devices should be defined such that the RF characteristics of the solution are independent on which secure element is used.

Where possible, approval bodies should seek to reuse approvals of other organizations with compatible requirements.

Industry wide forums defining Contactless RF technology for mobile devices should take into account the requirements for payment in any testing that they perform.

3.6.4 Digital Approval

3.6.4.1 Description

Digital approval tests that the parts of the communications between the application and reader conform to the specifications. This is largely a property of the payment application; however, the testing is generally performed on the complete solution (application plus platform). Due to this, most test benches are designed to test the digital conformance via an RF interface.

Whereas for analogue approval the platform was known but not the application, in this case the application is known but not the platform. Indeed, the application may need to run on multiple platforms.

3.6.4.2 Recommendations

Investigate the possibility of defining reference platforms for the testing of applications.

Standardisation of the application environments used will simplify this issue.

3.6.5 Approval of Secure Element

3.6.5.1 Description

Approval of the Secure Element involves testing that the secure element is well behaved (behaves according to its specifications) and its level of tamper resistance.

If the secure element is a separate module from the baseband mobile device then there is the possibility for this to occur independently of the mobile device.

However interfaces between the mobile device and the secure element offer possible attack points. This may require some level of approval of the handset.

If the secure element is integrated with the baseband processor then the boundary between the secure element and the rest of the processor capabilities needs to be defined.

3.6.5.2 Recommendations

Initially better to use well known and understood secure elements.

Development of new secure elements should take into account the need for evaluations to enable approvals.

3.6.6 Conveying Approval Level of Secure Element to Issuers

3.6.6.1 Description

Traditionally an issuer will source an approved platform and application, personalize this and send it to a user. There is full traceability through the process, and the business process ensures that the platform is approved.

In the case of mobile payment, it is more likely that the user has sourced the platform (the mobile device), and the user then request the issuer to personalize a payment application on this device. The issuer needs to have a means of determining that the platform is approved before the personalization takes place.

3.6.6.2 Recommendations

There must be sufficient traceability within the distribution and personalization system to provide assurance to the issuer that the Secure Element to which data is being personalized is approved.

4. Customer Care – Operator and Card Association

4.1 Introduction

This chapter considers issues related to customer care related business issues. The purpose is to ensure that the infrastructure put in place to support mobile proximity payment meets the requirements for the business. The intention is not to define the business and customer care processes, but to ensure that this area is not neglected, and that any applicable requirements are recognized.

Areas considered include disputes (and traceability for liability issues) and points of contact for customer care (who should the customer contact – bank, operator, payment association, handset manufacturer etc).

4.1.1 Guiding Principles and Assumptions

One basic assumption is that the customer does not know the source of the problem being experienced. The problem could be due to the user interface, or due to a specific set of payment credentials, with the user's credit card account, or even caused by a hardware failure.

When a problem is encountered, a user will call whomever they decide would be appropriate. Therefore, a process is needed for routing the query to the appropriate party for resolving the issue, along with the necessary information for identifying and resolving the issues as they are reported.

4.2 Customer Service Issues

4.2.1 Customer Service Process for Problem Resolution

4.2.1.1 Description

A process for routing customer service issues to the correct party, depending upon the nature of the problem, will be needed to ensure that the problem can be dealt with by the appropriate organization.

As there will be many entry points into the system; there will need to be a clearly defined management strategy for how the problem is addressed, so the customer is passed on to the proper entity to address the problem.

Depending upon the roles in the proximity payment ecosystem, it is possible that the operator may address many of the issues that are raised, and route the remaining issues to the proper entity to address.

For operators who do not take an active role in addressing these issues, then the account issuers should also provide an initial point of contact for the consumer in case problems are encountered.

4.2.1.2 Recommendations

When a subscriber encounters a problem and initiates a service call, it is likely that the subscriber will either call their mobile operator or the issuing bank for resolution.

When this occurs, one recommendation would be for the subscriber to be routed to a system which will help to quickly identify the nature of issue, and to assist in routing the issue accordingly. To ease in the initiation of this process, an initial point of contact could be identified as a display function in the user interface portion of the application.

The application itself should be designed, if possible, in such a way to assist in determining if the problem is with a particular set of payment credentials, the secure payment application, or the user interface.

4.2.2 Customer Support/Education

4.2.2.1 Description

Education of both consumers and customer service representatives will be an important factor in the successful deployment of proximity payment-enabled devices and services. This will assist the consumer to become comfortable with proximity payment services, and aid the relevant customer service centers to more quickly identify and resolve problems.

4.2.2.2 Recommendations

The party that registers the users should provide collateral (i.e. FAQs) which can help the subscriber to understand the proximity payment functionality of their device.

A list of common problems or standard error codes including suggested resolutions that any customer service center could deal with, to try and quickly resolve the issue, should be developed. This information should also be made directly available to the customer where appropriate (User Manual, etc).

At the retail points of sale where proximity payment-enabled devices are sold, the sales representatives should be able to assist the customer with purchase and activation of the device and functionality of the device. All other service issues should be directed to the appropriate resource depending on the nature of the problem.

4.2.3 Auditable Customer Service Trail

4.2.3.1 Description

In order to assist in the ongoing effort to manage the resolution of problems in the proximity payment space, there needs to be a way for what problems do get address, and how effective the solutions proposed are in resolving customer issues.

4.2.3.2 Recommendations

There needs to be an auditable trail throughout the customer service process, to identify when/where the process breaks. This will assist in determining the source of the problem, and how to address it.

5. Usability

5.1 Introduction

Usability addresses concerns with the experience of the user who wants to use a mobile device for proximity payment. Many aspects of the proximity payment process, from provisioning and personalization of the mobile device to selecting the desired credentials for payment, to performing the payment transaction, have usability concerns associated with them. The goal in identifying these issues, and in making recommendations to address these issues, is to ensure ease-of-use, minimize confusion regarding device functionality, and to in general achieve a level of consistency in the proximity payment user experience that will help promote broad acceptance.

5.1.1 Guiding Principles and Assumptions

The issues described in this chapter are related more to the experience of the user when performing the transactions being described than to the processes themselves. In some cases, more details regarding the specific processes being discussed are addressed in other sections of this document.

5.2 Transaction Experience Usability Issues

Transaction Experience issues are those issues which involve the user interaction with the mobile device when making a proximity payment. These include the process for selecting and managing payment credentials, and how the device will function when interacting with the PoS terminal.

5.2.1 User Authentication

5.2.1.1 Description

To prevent unauthorized use of a mobile device for proximity payment, it is expected that some type of authentication process will be optionally available for enabling the proximity payment functionality. Although it is likely that most users may not make extensive use of this process to secure their proximity payment credentials, just having it available will provide the user with peace of mind that their device can be secured.

One likely solution for this would be through the entry of a PIN code, set by the user, which would enable the payment application. Using a PIN code as the authentication mechanism to enable the proximity payment application is something that many users of today's mobile devices would already be familiar.

As currently used for some credit cards, and for authentication on GSM-based mobile devices, the PIN code gives the user the capacity to personalize his/her own security context (even if in the absolute, this security mechanism is quite poor, allowing only one combination out of 10E4). However, use of a PIN means that part of the security of the device is based on the user's own knowledge, which gives the user a strong (albeit subjective) trust in the service delivery.

Depending upon the requirements of the primary actors in the proximity payment transaction (account issuers, merchants, card associations, etc), authentication of the user could be required at different frequencies, including:

- Rarely, to simply enable use of the payment application
- Each time the user enables a particular set of payment credentials for making payments
- On a transaction by transaction basis

However, while this type of authentication process is certainly desirable, if it is too complex, or requires the user to authenticate themselves to their device too often, then it could be a disincentive for using the mobile device as a payment device.

5.2.1.2 Recommendations

Any sort of authentication process, whether it uses a PIN, biometric information, or some other mechanism, should be user-configurable. If possible, it should allow the user to disable the authentication process, if so desired, or at least reduce the frequency at which re-authentication must be done to the minimum level of authenticating the payment application, or when selected a new set of payment credentials (if multiple sets of credentials are present).

Also, the authentication process must utilize a secure entry method, and the secure element as needed, to ensure that the authentication information can not be recorded by another application running on the mobile device.

5.2.1.3 Remaining Considerations

In future payment systems, if PINs are ever used as a signature replacement for issuer authentication, then the user will no longer be able to switch off the requirement for PIN entry for authenticating payment.

5.2.2 Payment Credential Selection Process

5.2.2.1 Description

While the first mobile devices which are enabled for proximity payment will contain only a single set of payment credentials, in future implementations it is expected that each device will be able to hold multiple sets of payment credentials. These multi-credential implementations will require some mechanism for allowing the user to select which set of credentials to use for completing a specific transaction. This mechanism will likely involve some sort of complex payment application (e.g. “digital wallet”), which can access each set of payment credentials, either directly or through a set of individual payment applications, and allow the user to choose which set of credentials is to be used, either on a transaction-by-transaction basis, or based on a preconfigured set of user preferences.

If so desired, this sort of digital wallet application could possibly be enabled to perform a variety of automatic credential selection functions on behalf of the user, based on the amount of information available for each specific transaction, the types of payment credentials accepted by the merchant, and a set of predefined user preferences. For example, the user could set preferences such as:

- Default set of credentials to be used
- Priority order in which to use credentials for payment
- Credentials to use based on amount of transaction

The digital wallet could then automatically select the appropriate credentials for a payment application, from those credentials which are acceptable to the merchant.

Some potential digital wallet functions, such as the last one on the above list, is only possible if the amount of the transaction is made available to the device prior to the exchange of payment credentials; at this time, such a process is not supported by most POS devices currently in use.

5.2.2.2 Recommendations

Any payment application supporting multiple payment credentials should provide the user with the ability to select the set of credentials to be used on a transaction-by-transaction basis. It would also be desirable to allow the user to select a default set of credentials to be used, unless an alternate set of credentials is selected for a particular transaction.

The interaction between the mobile device and the POS should not allow the merchant to override the user preferences (for example by initially only offering the merchant's preferred payment method, and then expanding the range if the user does not have that).

It would also be useful if the payment application allows for the selection of a default set of credentials, to be used for any transaction in which the user has not chosen a specific set of credentials for completing that transaction.

5.2.2.3 Remaining Considerations

In the future, if additional POS interactions with the mobile device become possible, then a more intelligent selection process could be implemented in the digital wallet application. For example, if the POS terminal could communicate to the mobile device the amount of the transaction before payment, the application could allow the user to set preferences for using different credentials, depending upon the amount of the transaction. The payment application would then offer the desired set of payment credentials, depending upon the amount of the transaction.

5.2.3 Payment Credential Availability with Device Off

5.2.3.1 Description

One key issue with using a mobile device as a proximity payment instrument is whether the payment functionality could be available even if the device is powered off, and whether or not operation should be allowed in this case.

Technically, it is feasible to support proximity payment when the mobile device is not powered; the nature of the proximity modem technology is such that the POS device can provide the power necessary to enable a transaction to be completed. Whether or not such operation should be allowed, however, is a more contentious issue, with arguments being made both for and against doing so.

Potential concerns and issues that have been identified regarding having the mobile device proximity payment

capabilities operational when the device is not powered include:

- Since off-network lost or stolen device credentials can't be deleted OTA, the user could have concerns that unauthorized transactions could be made, even if the liability is limited, which might cause them to avoid using the device as a payment instrument. This could be a valid concern for transactions that do not require a signature, or for off-line POS transactions.
- If the device is powered off, there would be no way for a user to designate that a specific set of payment credentials be used (in multi-credential systems); unless a default set of credentials has been previously selected for use, the device would not be usable as a payment instrument.
- There may be account issuers who want more control over the management of credentials stored in a mobile device, who may argue against allowing credentials to be used unless the device is on-line, and some level of routine authentication is performed.
- It would not be possible to immediately indicate to the user that a transaction has taken place (e.g. no sound or flashing lights), so the user would need to rely on only the indication from the POS device that the transaction was successful.
- Any related branding opportunity that relied on the mobile device being powered up would be lost.

Arguments for having the proximity payment capabilities enabled when the device is powered are just as compelling. These include:

- If the mobile device cannot be used if powered down, then users will not rely on it as a payment instrument; instead, they will rely on more traditional payment mechanisms, rather than having to worry about whether their mobile device will "work" if power is lost.
- If a device is lost or stolen, the credentials could be revoked using the existing process for managing payment credentials, which would limit the liability for unauthorized payments made before the loss is reported to the account issuer, and the credentials are revoked.

- Other form factors used for proximity payment are passive, so the user's expectation might be that the same should be possible with mobile devices.
- While phone-off operation is not a requirement for proximity payment, there are other mobile-related applications, such as transit ticketing, which will require phone-off operation. Therefore, there will likely be requirements for some level of phone-off operation, even if the proximity payment capability is not available.

5.2.3.2 Recommendations

There is not a hard requirement that proximity payment functionality must work when the device is powered off. However, there are many benefits for the proximity functionality to work even if the device is powered off. It is recommended that devices be designed to allow the proximity functionality to work with the device powered off.

The mobile device should support the ability for an application to determine the device's power state, so that the payment application may restrict the level of functionality, based on the power state of the device. This could be done for security, branding or other reasons.

5.2.3.3 Remaining Considerations

In mobile device implementations that rely on the network to directly support some portion of the transaction process, either through on-line authentication, or via a server-based wallet which provides the credentials to the POS device, the device will need to be powered on to support proximity payment transactions.

In addition, if methods are devised to allow digital receipts to be delivered to the mobile device, either through the proximity modem, or via the wide-area network, these methods would be limited if the device is not powered.

5.2.4 Payment Interaction with Phone Operation

5.2.4.1 Description

Mobile telephones are one type of mobile device that are likely to have proximity payment capabilities integrated into them. Of course, since the primary function of these devices is to support voice calls, it will be important for the payment application to not interfere with the normal operation of the telephone. One

example of this sort of interference might be interrupting the telephone call with an audio indication that a transaction has completed.

Such interruptions of the user experience could potentially occur at any point during the calling process, including

- During a telephone call
- When ending a telephone call
- When initiating a telephone call

While the last two are less likely to contribute to a user negative experience than the first, it is important to account for any of the possibilities when designing the payment application, in order to avoid a negative payment experience for the user.

Also to be considered is when the phone is being used in alternate ways, such as with a hands-free speakerphone, or with a wired or wireless headset. In these instances, alternate methods of user interaction may be appropriate.

5.2.4.2 Recommendations

Proximity payment functionality must be able to work when a telephone call is in progress. This is necessary to ensure that a payment could be made without the user having to first end a telephone call.

In addition to the proximity modem working, the UI must still be available for configuring a device to make a payment, or to authorize a payment, in case some user intervention is needed while a telephone call is in progress.

5.3 Network Interaction Issues

5.3.1 Provisioning and Personalization Process Requirements for the User

5.3.1.1 Description

Chapter 2 detailed the process for provisioning and personalization of a proximity payment-enabled mobile device, and identified multiple issues related to that process. However, there are also usability questions related to this process.

If the process for initiating the OTA provisioning and personalizing of his/her mobile device, or to authenticate the user and the mobile device to the issuing bank so that the

provisioning process can be successfully completed is too complex or cumbersome, the user may be disinclined pursue it.

Also, for devices that are already provisioned for proximity payment, the extent to which the user should be involved in the update of a payment application is unclear. For example, should the user have to accept an update which is being pushed to the device, or should it happen transparently to the user? While it is often good practice for the user to have control over what is happening on their mobile device, an unsolicited update is likely to cause confusion if there has not been prior notification. If the user is confused by such an update, they could reject the update due to security concerns, which in turn could cause problems in the roll out of new or updated applications. Requiring user interaction may also limit the times at which the application can be delivered to those when the user is able to interact.

5.3.1.2 Recommendations

The registration and provisioning process needs to balance security requirements with ease of use, in order to encourage uptake.

In general, users should be notified of payment application updates, in particular when updates were made without the user's knowledge, or where the user will notice changes in appearance or functionality of the application.

5.3.2 Payment Credential Availability Off-Network

5.3.2.1 Description

Similar to the issue described in 5.2.3, there is also a question as to whether a mobile device should be enabled for use in a proximity payment transaction if the device is not connected to the mobile network. This issue revolves mainly around the concerns over unauthorized use of a lost or stolen mobile device, and the desire to be able to delete credentials from a mobile device OTA as soon as it is reported as its loss is reported by the user.

One advantage of requiring a network connection to use credentials installed in a mobile device for proximity payment is that it would provide the account issuer with a mechanism to remotely manage, and delete if necessary, a user's payment credentials when necessary. So, if a device is lost or stolen, the issuing bank would be able to delete the payment credentials at

- Must a mobile device touch the POS device to complete a transaction, or is “close” good enough? If so, how close?
- Can the device be waved in front of the POS device, or should it be held still near the POS device?
- For one-piece mobile devices, should the front or back be touched to the POS device?
- For flip mobile devices, should the flip be opened or closed?
- For flip mobile devices, should the flip be touched to the POS device, or the main body of the device?

Without clear instructions on how to handle each mobile device, the user could quickly become frustrated with trying to use their device as a payment instrument, and instead choose a different form factor for making proximity payments.

5.4.1.2 Recommendations

It is strongly recommended that each mobile device manufacturer provide clear instructions for each proximity payment-enabled device on how to hold the device when making a proximity payment, and what spot on the device must be touched to the POS device to ensure a successful transaction.

The user should be able to hold the device in a convenient manner when making a proximity payment transaction.

When making a transaction, the device should be positioned in such a way that the user will see and on-screen indications or branding.

A target mark, if available, would assist the user in understanding what spot on the mobile device should be touched to the POS device to make a proximity payment.

5.5 Branding

This section examines some of the issues with branding of technology and services on proximity-enable mobile devices.

In general, there are three types of branding:

- A target or usability mark specific to the contactless function, such as the NFC Target Mark. This might be a static brand or mark on the mobile device, indicating that it includes a specific technology.

- The branding of the overall class of service that a device is capable of, such as the ability to make a proximity payment, or the ability to read an RFID-enabled smart poster.
- The branding of a value-added service, such as a mobile operator-branded e-wallet service or a payment-instrument issuer or payment association brand.

Which level of branding is used with a mobile device will depend primarily on the business relationships involved, and on the functionality of the specific device. In some instances, only the value-added services might be branded, while in others a device's specific capabilities might also be branded, to meet specific market needs.

The form factor of the device will also need to be taken into account when considering branding issues. The placement of the target mark, and the ability to display branding via the User Interface will be impacted by the form factor of the device.

5.5.1 Branding on the Proximity-enabled Mobile Device

5.5.1.1 Description

Branding on the device is possible in a number of ways. The Secure Element itself might be branded, but this is limited to the size and the visibility of the Secure Element to the user. In these instances, a removable Secure Element (e.g. SIM/USIM, or SMC) would be more suitable to branding.

Branding on the mobile device is also possible, and this is often done to some degree today. Such branding could be limited to a target mark, or could include more extensive branding on the device itself. However, there can also be security concerns with this type of branding; for example, it may not be appropriate to brand a mobile device with a specific payment brand, or even as "proximity-payment enabled", since it could make the device more attractive to thieves.

5.5.1.2 Recommendation

Branding on the Secure Element is not of significant importance, due to its limited visibility to the user.

Branding on the mobile device itself could be of greater value, but the allowed branding on the physical device will ultimately be a product of the business relationships between the mobile device manufacturer, the mobile operator, and any related service providers.

In general, more focus will be placed on branding via the mobile device's user interface than on branding on the physical device itself.

5.5.2 Branding via the User Interface

5.5.2.1 Description

Service providers (banks, petrol stations, etc.) have two possible opportunities to dialog with the user and display their brands:

- During the operation of the application (e.g. loading of tickets, reloading of the e-purse, checking of the loyalty points left)
- During the contactless transaction itself

Some action by the user may be needed, depending on the application (selecting a payment instrument, enter PIN code). At the end of the transaction, a brand could be displayed to the user via the UI.

There could also be different branding requirements from different service providers. For example, branding could be visual and/or audio.

Depending on the technology used for the UI, it may or may not be possible to display a brand. For example, the SIM Tool KIT is very limited, and might not support the display of a brand, whereas a browsing solution or JAVA solution provides better ergonomic ability for branding.

5.5.2.2 Recommendations

As the primary means of branding is going to be via the UI, it is important that the UI elements are visible to the user during the transaction.

It is recommended that Browsing and Java technologies are used for the UI. STK can be used where the application doesn't need more than text to dialog with the user.

In any case, branding should not be displayed when the user is making or receiving a telephone call.

Delivery of the receipt to the mobile device via the wide-area cellular network would require a mechanism for the merchant POS system to acquire an address for delivery of the receipt, and any information beyond the transaction details (amount, date/time) would also require modifications to the POS system. In addition, as already mentioned the delivery of any detailed information to the mobile device might require the use of more enhanced broadband services, such as MMS. The benefit of this approach would be to enable the merchant to also provide various after-market services, such as targeted thank you messages, or electronic coupons, to encourage the user to return on a later date.

Delivery of the receipt information to a web site, whether maintained by the account issuer or the mobile operator, would provide the benefit of being able to collect as much information on the transaction as the merchant could provide. However, access to this information by the user might not be possible in real time, and collection of the information by the user for tracking/archiving purposes would require a further download, possibly through a different access mechanism.

5.6.1.2 Recommendations

If the receipt is delivered to the user via the proximity modem, then the level of detail to be transmitted should be limited to the amount of data that can be transmitted over the proximity interface in a brief period of time, to avoid an unacceptable user experience.

6. Interworking

6.1 Introduction

Mobile proximity payment is not a stand-alone application. It exists in the context of other mobile services and uses infrastructure which is not dedicated to mobile proximity payment.

The mobile device is a multipurpose device. For example, a mobile phone is primarily a telephony device, and may also incorporate a number of other features such as games and PDA functionality, as well as proximity applications.

The proximity payment application may be just one of a number of proximity applications, and these may include applications where the mobile device acts as a reader and others where the mobile device emulates a card.

The mobile proximity applications may operate in an environment specifically designed for mobile devices, however the greater opportunity exists where the mobile device is used with existing acceptance infrastructure. Where the existing infrastructure is used the mobile payment system must conform to the requirements of the legacy systems.

This section considers a number of issues which arise due to the interworking of the mobile proximity payment with existing infrastructure and services.

6.2 Application Issues

6.2.1 Multiple Payment Applications on A Mobile Device

6.2.1.1 Description

Many of the existing proximity payment systems have been designed around the payment credentials being stored on a dedicated device, such as contactless payment cards, key fobs or tags. These devices are supplied by the issuer, and hence the number of payment credentials on the device is constrained in a well defined manner.

Typically there will only a single set of credentials, or possibly a very small number (for example, debit and credit), and these will be issued by a single issuer for a single payment association. Under these conditions, payment credential

selection is performed by the user presenting a particular device to the reader. The POS may cycle through the payment brands which are accepted at the merchant until that available on the device is encountered. In the cases where multiple credentials exist, these will usually be selected on the POS device.

This method of selection will not work in the case where multiple credentials exist on a single device. Conceptually, the mobile device now acts like a physical wallet in which multiple cards are stored. Bringing the mobile device into proximity with the POS terminal is analogous to putting a wallet in proximity and expecting the POS to correctly choose the correct card.

The mobile device needs to convey to the POS terminal information regarding which of the payment credentials has been selected. One method of achieving this is to present the available credentials, with an indication of the selection priority. This would assist where not all of the available payment credentials on the device are accepted by the merchant. The POS may select the highest priority credentials which are accepted. This however requires that the POS terminal currently expects the possibility of a multiplicity of credentials, and also that the method of selection between the different payment applications and brands is compatible. If protocols for exchange of the transaction credentials differ greatly between payment applications it may be that if the presence one payment application (or set of payment credentials) has been identified by the POS it will be impossible to select a different payment application (or payment credentials). In this case, the mobile device may need to implement a filter in order to only make the selected payment credentials visible to the POS terminal.

6.2.1.2 Recommendation

The mobile device must provide a mechanism for a user to select the desired payment credentials. This selection must be reflected on the interface between the mobile device and POS terminal.

It is recommended that the mobile device either filter the list of available payment applications and credentials to only expose the selected credentials to the POS terminal, or that if the presence of multiple applications and credentials is exposed, that sufficient information about the user selections is

conveyed to ensure that the credentials intended by the user are read by the POS terminal.

6.2.1.3 Remaining considerations

In order to provide information regarding the selection of payment credentials to the POS terminal requires that the mobile device is aware of the relationship between the various payment applications and credentials. There may be a need to standardize an interface for the registration and management of payment applications and credentials within the mobile device.

6.2.2 Anti-collision

6.2.2.1 Description

The selection of payment credentials in existing payment proximity payment systems on the basis of which physical device is brought into proximity with the POS terminal is discussed above in section 6.2.1. One of the consequences of this method is that if multiple devices supporting payment are simultaneously in the field of the POS reader the POS cannot determine which should be used. This is resolved by removing one of the devices from the field.

Due to the complexity of determining if a device supports payment (due to potentially needing to check for multiple types of payment), and also due to concerns about the reliability of anti-collision procedures, some existing schemes require that if the POS terminal detects multiple proximity devices in the reader field the transaction should be aborted, regardless of whether more than one device supports payment.

A device supporting multiple different proximity applications, some of which may use different selection mechanisms (for example, an RFID tag as opposed to a contactless microprocessor card), may expose the different applications by responding as a collision of multiple logical devices. This may cause the transaction to be aborted as described above. In this case it is impossible to remove the extra devices from the reader field.

The mobile device must be capable of being configured to appear as a single device in order for the transactions to proceed.

6.2.2.2 Recommendation

If the mobile device can expose the proximity applications by means of a collision of multiple logical devices, then it must be configurable such that it can appear as a single payment card.

6.2.2.3 Remaining considerations

Certain applications may make use of the anti-collision feature, or low level physical identifiers, for example, some access control systems are coded to the physical identifier of the Contactless RF device, rather than making use of an application residing on the device. In such a case, the physical identifier of the card is required, and it may be required to expose such applications via a collision.

This implies that the mobile device needs to understand the context in which the mobile device is being used in order to determine the correct behaviour and configuration of the collision feature. This may limit the extent to which proximity applications can automatically be used without user interaction.

6.2.3 Conflicting Analogue Requirements

6.2.3.1 Description

The mobile device which supports multiple proximity applications (including payment applications) may be subject to a number of interoperability standards, particularly on the analogue Contactless RF interface. These include such issues as read range.

As each application may define its own requirements around the analogue characteristics, there is the possibility that these will conflict in such a manner that it is impossible for the mobile device to meet all of the requirements simultaneously. Although some of the digital characteristics may be altered depending on context, it is much more difficult to modify the analogue characteristics.

6.2.3.2 Recommendation

Industry Standards groups that are defining the standards for testing/certification for analogue requirements, such as the NFC Forum, should take into account the analogue requirements for mobile proximity payment when establishing these standards.

If groups defining requirements for other service areas also published requirements for analogue performance, it would assist in the development of common standards across multiple application spaces.

There are locations in which the use of mobile telephones is prohibited. For example, in many countries it is prohibited to use mobile phones on the forecourts of petrol (gasoline) stations. In other situations, it is desirable that mobile phones are turned off (for example in a cinema) due to possible disturbance to others.

This will affect the ability to pay with the mobile device. It is possible to switch off the features which cause the prohibition, for example in a cinema the phone may be muted, and many (although by no means all) phones now offer a “airplane” mode, where cellular transmissions are switched off.

Another possibility is for the mobile device to allow payment when the phone is switched off. While this may be technically possible, any use of the user interface of the mobile device will be disabled when mobile device is off.

The need to put the phone (or mobile device) into a state other than the default in which it may validly be used may be confusing to users. It may also introduce enough complexity to make it sufficiently inconvenient to users to make use of it.

Additionally, the mobile device would need to provide some indication, for example to petrol station attendants, that the mobile device is in a state where it may validly be used.

6.3.1.2 Recommendation

Clear guidelines should be provided on the use of mobile devices in places where cellular transmissions are prohibited. These should take into account the practicality of indicating the state of the device to those who may need to enforce the usage policy.

6.4 Multi-technology issues

6.4.1 Support of multiple Contactless RF technologies in a single device

6.4.1.1 Description

There are a number of variants of Contactless RF, for example, ISO14443 Type A, ISO14443 Type B, FeliCa and NFCIP-1 to name a selection. In systems in which Contactless RF devices can be of multiple types, generally the Contactless RF device can be one of a number of variants, and the reader is required to support multiple variants in order to work with all valid devices.

A mobile device, on the other hand, will be expected to support multiple variants as a Contactless RF device, and possibly also multiple variants as a reader. As such, it is possible that the mobile device may respond in combinations which are unexpected by a reader.

An example would be a transit application operating on FeliCa technology, and a payment application operating on ISO14443 Type A RFID technology. A multi-application reader may be used to both accept proximity payment, and to provide tickets for the transit application. It would be possible that the reader may constantly query for type A and FeliCa in turn. In dedicated Contactless RF devices, if the device was a type A RFID card, the reader would know that it is not a transit device (as that would be a FeliCa device) and not proceed if was looking for a transit device (and vice versa). The mobile device, as a multi-application, multi-technology device, may support transit on FeliCa, but also an ISO14443A application¹¹. It may respond to the ISO14443A query, and hence the reader may assume it is not a transit device, and not register it. This could cause indeterminate behaviour, as if the reader had queried for FeliCa first, then the mobile device would have responded correctly.

6.4.1.2 Recommendation

It is recommended that this issue be considered in more detail. It may be necessary to enable the response to particular technologies to be turned off based on the application which has been selected for use.

¹¹ Of course the reader may be designed in a manner which can deal with the multi-technology nature of the mobile device. However the mobile device may be interacting with an existing infrastructure which has been designed with assumptions which exclude the possibility of multi-technology devices, therefore this is a concern which should be considered.

7. Conclusion

7.1 Summary

This document details a variety of issues, grouped under a number of broad categories, which must be addressed to ensure that the deployment of proximity payment technology in mobile devices will be successful. It provides a variety of recommendations which should be followed in order to achieve this goal, as well as a limited number of recommendations which will need to be met for the key members of the proximity payment ecosystem to support deployment of this technology.

The categories detailed in this document include provisioning, personalization, security, customer care, usability, and interworking, all of which are relevant to the proximity payment applications space.

With such a large number of issues detailed, it is natural that some sort of prioritization should be appropriate, so that readers of the document will understand what issues require the most immediate attention.

7.2 Recommendations

While all of the categories and specific issues detailed in this document are important, some of the issues have been identified as those which are both important, in terms of how they must be addressed, and urgent, in that they must be a priority to address from an industry standardization perspective, since they will have an immediate impact on the proximity payment devices and supporting infrastructure which is being developed today.

Two areas in particular are of priority for the development of the infrastructure to support mobile proximity payment:

1. Development of a personalization infrastructure;
2. Development of an approvals process for mobile devices for proximity payment.

These areas are identified as critical to scalability of mobile proximity payment systems across mobile operators, issuers, payment associations, secure elements and handset types. As such, these are activities which need the participation of many areas of industry

Other areas which are important, in particular to those designing mobile proximity payment systems and components include:

- Security
 - 2.6.1 Protection of Payment Assets
 - 2.6.2 Protection of Transaction Data
 - 2.7.1 SE Platform Controller and Manager
 - 2.7.2 User Registration
 - 2.7.3 Authenticity / Integrity of the Payment Application / Credentials
 - 3.3.1 Mobile Platform Application Environment Choice
 - 3.3.2 Mobile Platform Security Level
 - 3.3.3 Access to the Secure Element
 - 3.3.4 Access to the Proximity Modem
 - 3.4.1 SIM/USIM
 - 3.5.1 Isolation of Payment Application
 - 3.5.2 Pin Code
- Standard Operating Environment
 - 3.2.5 Standardisation of Secure Element Application Environment
- Swapping Removable Secure Element
 - 2.8.6 Device Change with Removable Secure Element
- Form Factor
 - 5.4.1 Physical Positioning of a Mobile Device for Payment
- Branding
 - 5.5.1 Branding on the Mobile Device
 - 5.5.2 Branding via the User Interface