MobilePay AppSwitch Implementation Guide

Version 2.1

Table of contents

MobilePay AppSwitch Implementation Guide

1 Change log

Version	Date	Amendment
1.0	01.10.2014	Document created
1.1	08.10.2014	Error scenarios added.
1.2	31.10.2014	More details added by Danske Bank and Trifork.
1.3	28.11.2014	More details added by Danske Bank and Trifork.
1.31	15.01.2015	Chapter 10 updated.
2.0	26.03.2015	TKI: Document updated to support latest AppSwitch release 1.6
2.1	27.05.2015	TKI: Document edited to treat AppSwitch only (SDK)

MobilePay AppSwitch Implementation Guide

2 Purpose of the implementation guide

This implementation guide explains the design and implementation process of MobilePay AppSwitch. This includes descriptions of MobilePay AppSwitch communication, error scenarios, security, and test setup.

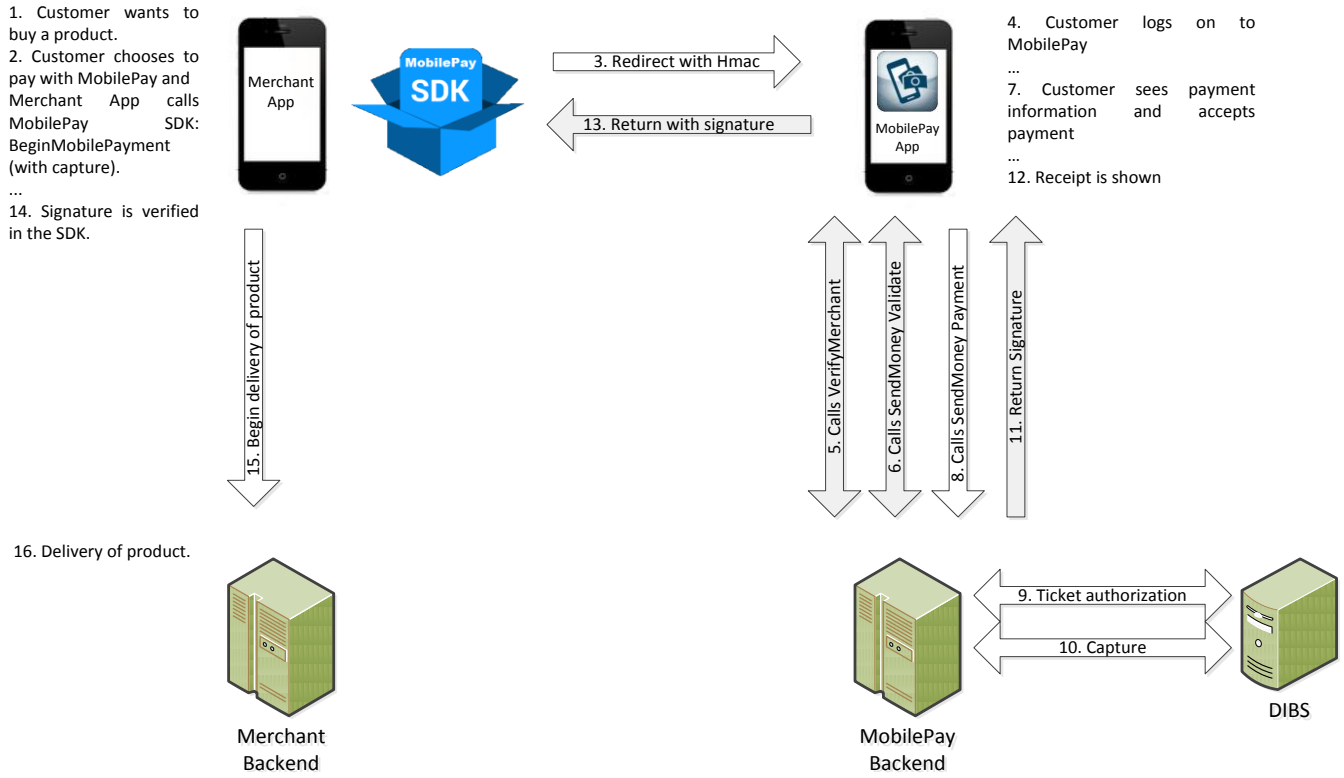
2.1 Target groups

The target group is project managers, system architects, and developers.

Implementation details such as coding details and platform specific details (Android, Windows Phone and iPhone) are not part of this guide. They can instead be found on GitHub under Danske Bank's AppSwitch SDK repository.

MobilePay AppSwitch Implementation Guide

3 MobilePay AppSwitch - App only



4 Communication

The merchant app communicates with MobilePay AppSwitch SDK and typically also with a merchant backend.

The MobilePay app communicates with MobilePay AppSwitch SDK and MobilePay Backend.

Selection of payment option MobilePay in the merchant app implies that the MobilePay AppSwitch SDK redirects to the MobilePay app. This redirection includes verification at the MobilePay backend of current merchantID (ID provided by Danske Bank) and specific secure data (HMAC calculation based upon an agreed key) used for validation of current message content sent from merchant app.

The customer follows the well-known MobilePay steps as logon, payment approval (swipe), and payment completion.

When the requested payment is completed in MobilePay, a signature is returned to the merchant app. The signed information contains the original merchant orderID and the MobilePay transactionID. The MobilePay AppSwitch SDK validates the signature and ensures that the orderID given to the MobilePay AppSwitch SDK by the merchant app matches the orderID received from MobilePay.

Verification of the digital signature and error handling are both handled in the MobilePay AppSwitch SDK.

There are a large number of services which the MobilePay app uses to communicate with the MobilePay backend at Danske Bank.

The services can be divided into two types of services; namely, authenticated and not authenticated services. Authenticated services require a PIN code, installed MobilePay appID and related MobilePay mobile phone number. Not authenticated services include services for registration of new users, getting app release information, help texts, contact information etc.

Examples of MobilePay app services are:

1. VerifyMerchant: Verifies that the HMAC and the merchantID are correct.
2. Logon: Enable MobilePay access (authenticated)
3. SendMoneyAppSwitch: Transfers the money from the customer to the merchant account. This step involves creation of a transactionID (paymentID) which is the signature of a payment. The SendMoneyAppSwitch service has two actions: VALIDATE and PAYMENT. The VALIDATE action verifies whether the customer can do the payment (ability in relation to the amount of payment and paycard status), but no payment is done. The PAYMENT action executes the actual payment. The VALIDATE action will also verify if current specified orderID

(provided from Merchant App) in current payment request has already been paid. If already paid, the payment details and signature will be returned (authenticated).

4. ShowList: Returns all the customers' transactions in MobilePay (authenticated).
5. ShowDetails: Returns the payment receipt and transactionID in MobilePay (authenticated).
6. AppVersion: Returns the minimum MobilePay app version required.

All security operations are executed in the MobilePay backend and not in the MobilePay app.

Furthermore, there is no need for any user administration or access control for the merchant to any servers at Danske Bank and there is no need for the merchant to host any servers that Danske Bank needs access to.

4.1 Communication between the merchant app and MobilePay app

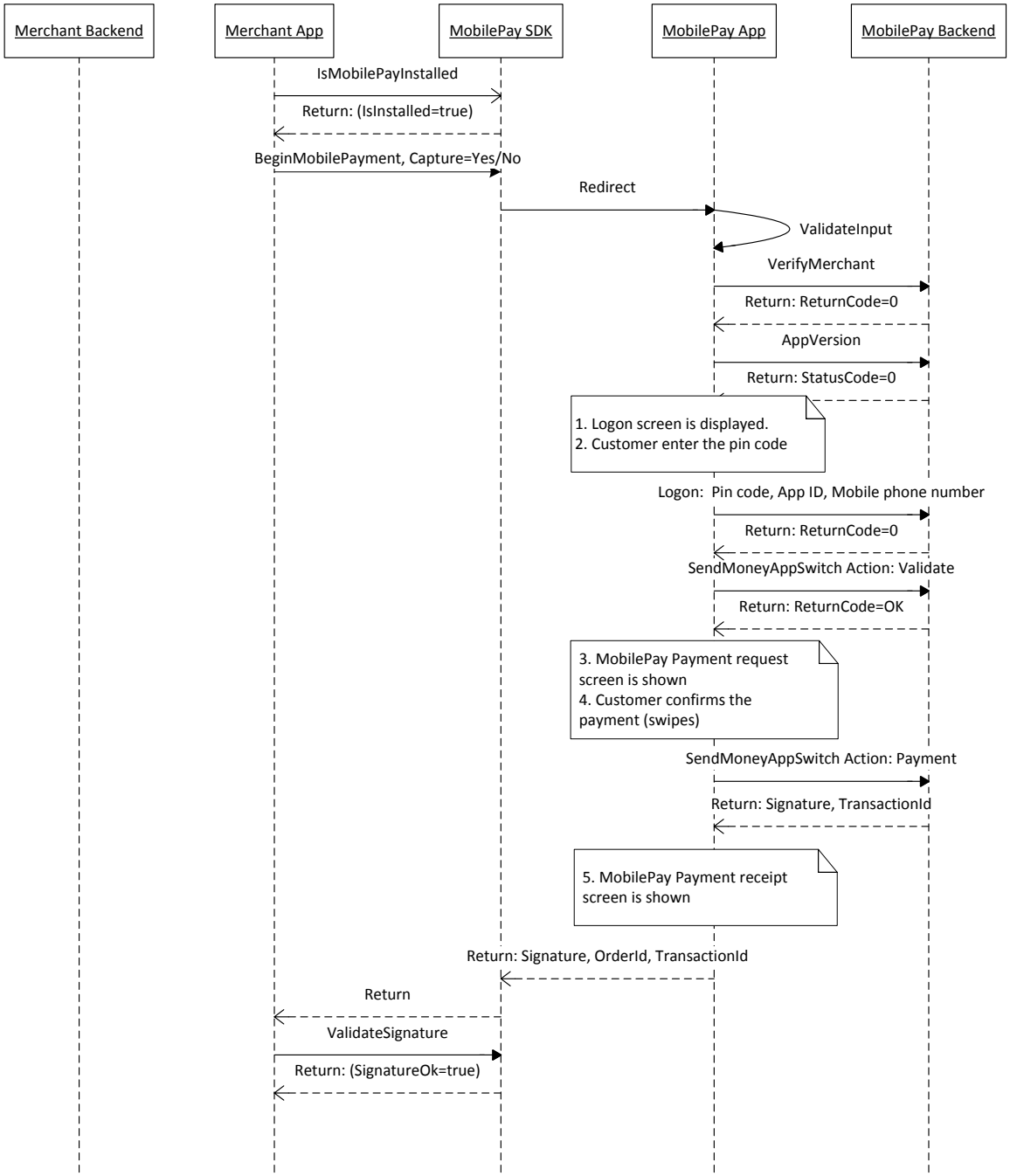
The data exchange between the merchant app and MobilePay app and vice versa is defined in sections 6.1 and 6.2, respectively. The merchant app utilizes the MobilePay AppSwitch SDK to start a payment that will redirect the payment inputs to the MobilePay app.

The MobilePay app validates the input and requests the user to login. After login the user will be presented to a payment request once a payment validation has taken place (user looked up in DIBS regarding ability to pay with current pay card in MobilePay).

When the user accepts (swipes), the payment is authorized at DIBS. The parameter "Capture" from the SDK can be either 'Y' (yes) or 'N' (no). By default, "Capture" is set to "Y". If set to "N", a reservation takes place – this is a non-supported feature of the current version of AppSwitch – App only.

The request is confirmed in MobilePay backend – a receipt in response is sent to the MobilePay app. The MobilePay app will then redirect back to the merchant app with a signature of the payment.

If the transaction is captured the signature will contain the status of the payment.



If these steps are successful, the order can be delivered.

4.2 How to avoid double payments

The MobilePay app has a built in validation which ensures that the customer will not accidently pay for the same service/product twice.

1. If the merchant app sends a new orderID to the MobilePay app, the customer will be able to pay for the order, and the money will be transferred to the merchant's account.

2. If the merchant app sends an orderID to the MobilePay app, which the customer has already paid for, the payment signature including the existing transactionID (paymentID) will be returned without a new payment taking place.

MobilePay will ensure a one-to-one relation between these artefacts:

- OrderID (from merchant app)
- Actual payment (money transfer)

Normally, there will also be a one-to-one relation between orderID and transactionID:

- TransactionID (paymentID from DIBS).

The customer is redirected from WIFI to phone network if any network issues occur. The network might resend the transaction, but double payments will not occur, because the second payment will be rejected due to the same orderID not being allowed for payment confirmation.

4.3 How to ensure authentication of the calling merchant app

The merchant is registered in the MobilePay backend with an AppSwitch ID (also called merchantID). This ID is attached to a Business Online agreement.

The MobilePay backend will verify the merchant in the following steps:

- Is the merchantID registered in the MobilePay backend database?
- Is it registered as being active?
- Is the HMAC correct (please refer to the section "Security and Certificates" in chapter 6)

4.4 How to ensure authentication of the payment

A MobilePay AppSwitch payment returns a digital signature to the calling merchant app. This signature ensures the authentication of the payment, because the Danske Bank private key has been used for

signing and therefore the Danske Bank public key can be used for verification at the merchant side. The public key is contained in a certificate and exchanged in a secure manner separate from the payment transaction.

MobilePay AppSwitch Implementation Guide

5 Error codes

The following sections describe the different error codes that can be returned from the MobilePay AppSwitch SDK.

The error codes from the MobilePay AppSwitch SDK are listed in the table below:

No.	Type
1	Invalid parameters to MobilePay app
2	Validate merchant request fails
3	MobilePay app version is out of date
4	Merchant Id is not valid
5	HMAC parameter is not valid
6	MobilePay timeout
7	MobilePay amount exceeded
8	Merchant app timeout
9	Invalid signature – Danske Bank has not signed it.
10	MobilePay AppSwitch SDK version is outdated
11	The order Id sent to MobilePay has already been used for a confirmed payment by the same merchant.

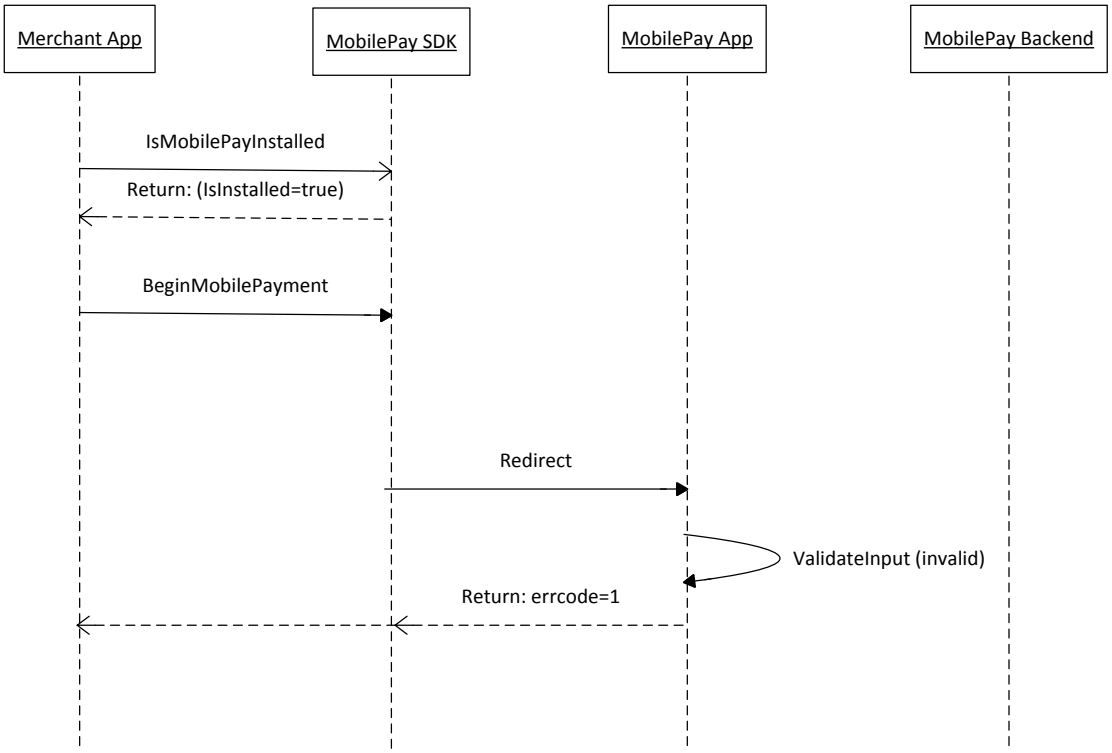
Most of these errors are technical errors (1,2,4,5,9,10,11) that must be presented as a generic error message without details to the user, where others ***should be handled by the merchant app*** (3,6,7,8) to guide the user in the right direction.

5.1 Invalid parameters to MobilePay app

The MobilePay AppSwitch SDK will return error code no. 1 to the merchant app if the input sent to the MobilePay app is invalid. The input can be invalid if e.g. the price is lower than 0 or if a required input is missing.

This error code should never be received in the merchant app in production. The error code should be handled in the merchant app by showing a message to the user and create a log in the merchant backend (if this is possible) that a problem with initiating a payment request has occured.

Example of return text: "MobilePay error 1: Payment parameters not valid."

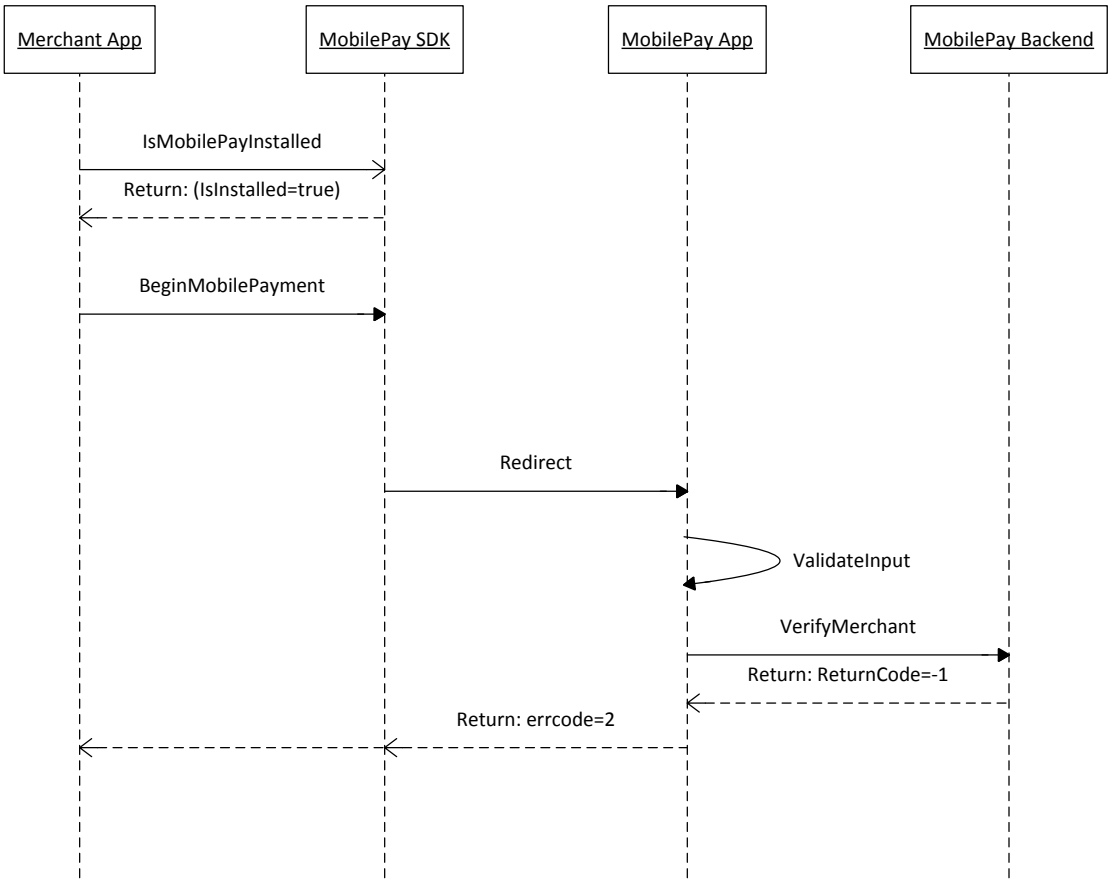


5.2 Validate merchant request fails

The MobilePay AppSwitch SDK will return error code no. 2 to the merchant app if the validation of the merchant failed due to network failure or timeout. The MobilePay app will validate the merchantID sent to it by making a validation request to the MobilePay backend.

This error code should be handled in the merchant app by showing a message asking the user to check network connectivity and try again.

Example of return text: "MobilePay error 2: Merchant validation failed. Check network connectivity and try again."

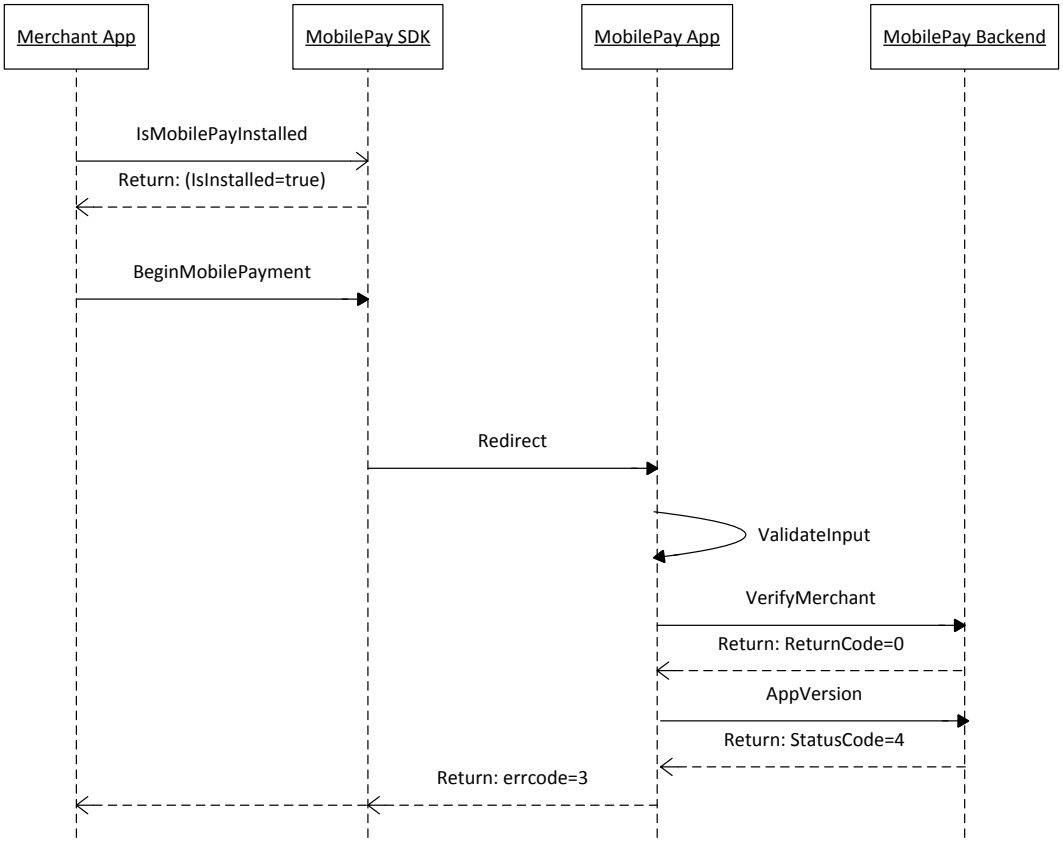


5.3 MobilePay app is out of date and must be updated

The MobilePay AppSwitch SDK will return error code no. 3 if the MobilePay app must be updated which can be due to security or legal requirements.

It is the responsibility of the merchant app to inform the user that the MobilePay app should be updated before trying again. The merchant app can show a message to the user and provide a link to an app store to download the latest version of MobilePay.

Example of return text: "MobilePay error 3: MobilePay app is not updated. Please do so and try again."

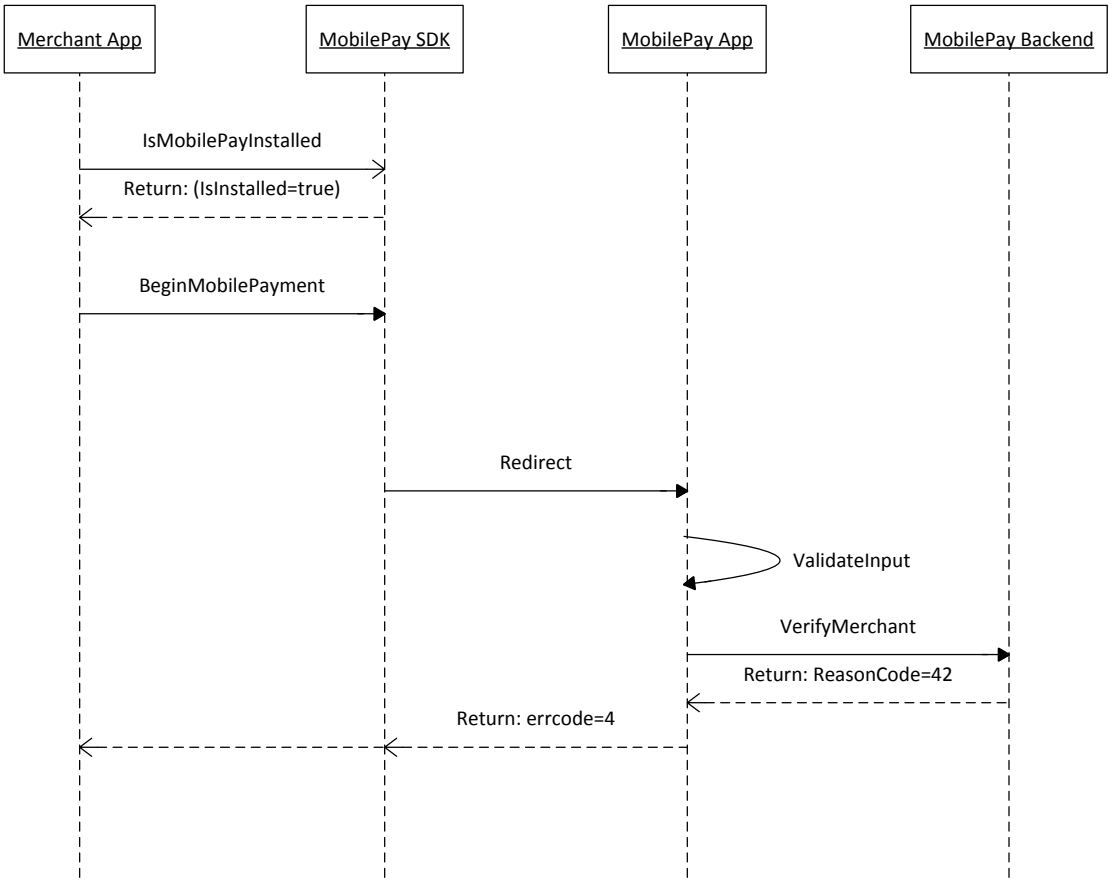


5.4 MerchantID is not valid

The MobilePay AppSwitch SDK will return error code no. 4 if the merchantID received by the MobilePay app is invalid. The MobilePay app validates the merchantID by making a validation request to the MobilePay backend.

This error code should never be received in the merchant app in production. The error code should be handled in the merchant app, by showing a message to the user and log it to the merchant backend (if possible).

Example of return text: "MobilePay error 4: The business has not been approved to carry out payments in MobilePay."

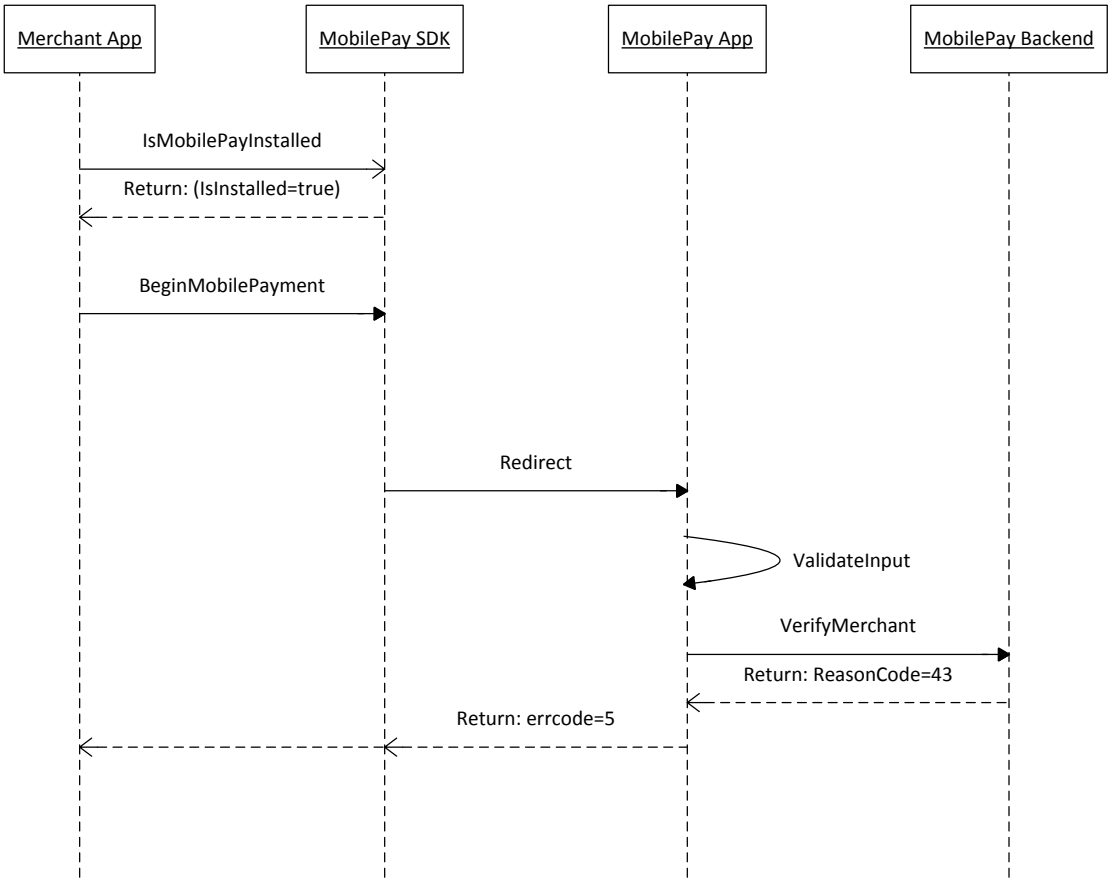


5.5 HMAC parameter is not valid

The MobilePay AppSwitch SDK will return error code no. 5 to the merchant app, if the HMAC parameter received by the MobilePay app is not valid due to value not matching – if not caused by corrupted data it might be caused by missing synchronisation in understanding of calculation – calculation method, content of message, and/or wrong key used.

This error code should never be received in the merchant app in production. The error code should be handled in the merchant app by showing a message to the user and log it to the merchant backend (if possible).

Example of return text: "MobilePay error 5: Error in payment data."

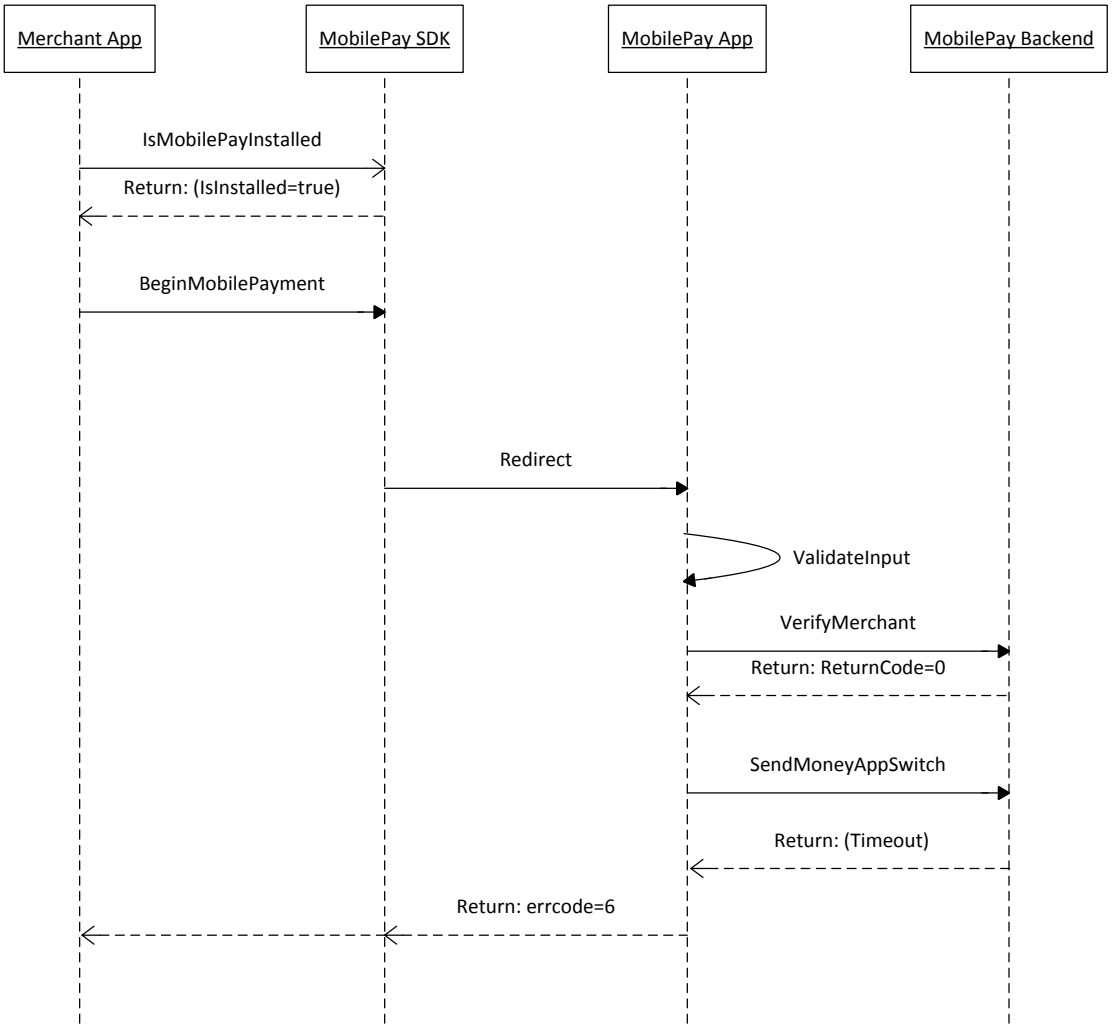


5.6 MobilePay timeout

The MobilePay AppSwitch SDK will return error code no. 6 to the merchant app if the call to the MobilePay backend (SendMoneyAppSwitch handles the payment) times out which is by default set to 5 minutes.

The merchant app should show a message to the user informing about the timeout and let the user try the same transaction again. The second attempt will then check if the previous request did succeed in the MobilePay backend and base its reply upon the result of this check.

Example of return text: "MobilePay error 6: MobilePay lost the connection to the server. Try again later."

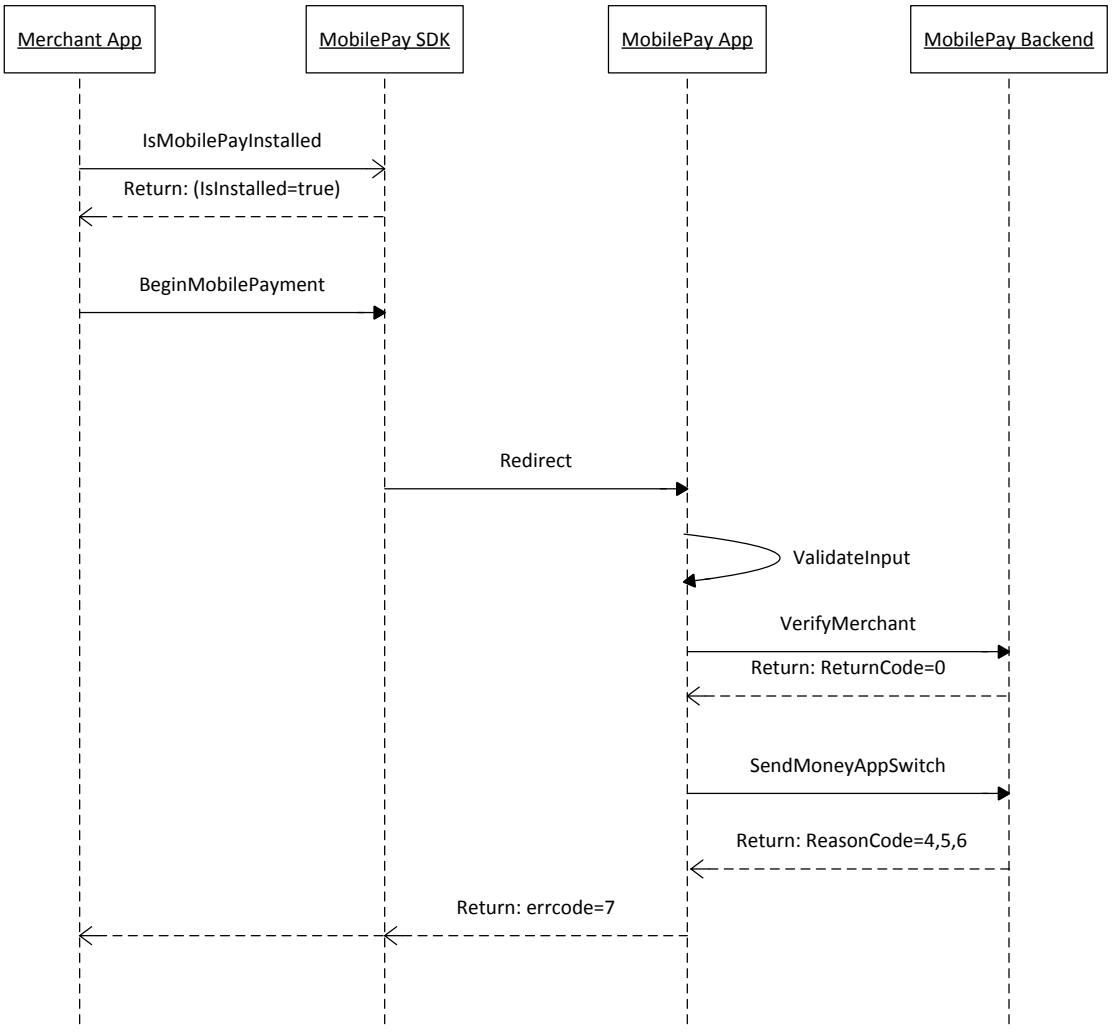


5.7 MobilePay amount exceeded

The MobilePay AppSwitch SDK will return error code no. 7 to the merchant app if the user's own daily or yearly limits are exceeded by the requested amount in the order.

The merchant app should show a message informing the user about exceeding the limit and that the user can view the limits under "Beløbsgrænser" in the MobilePay app.

Example of return text: "MobilePay error 7: MobilePay amount limit exceeded."

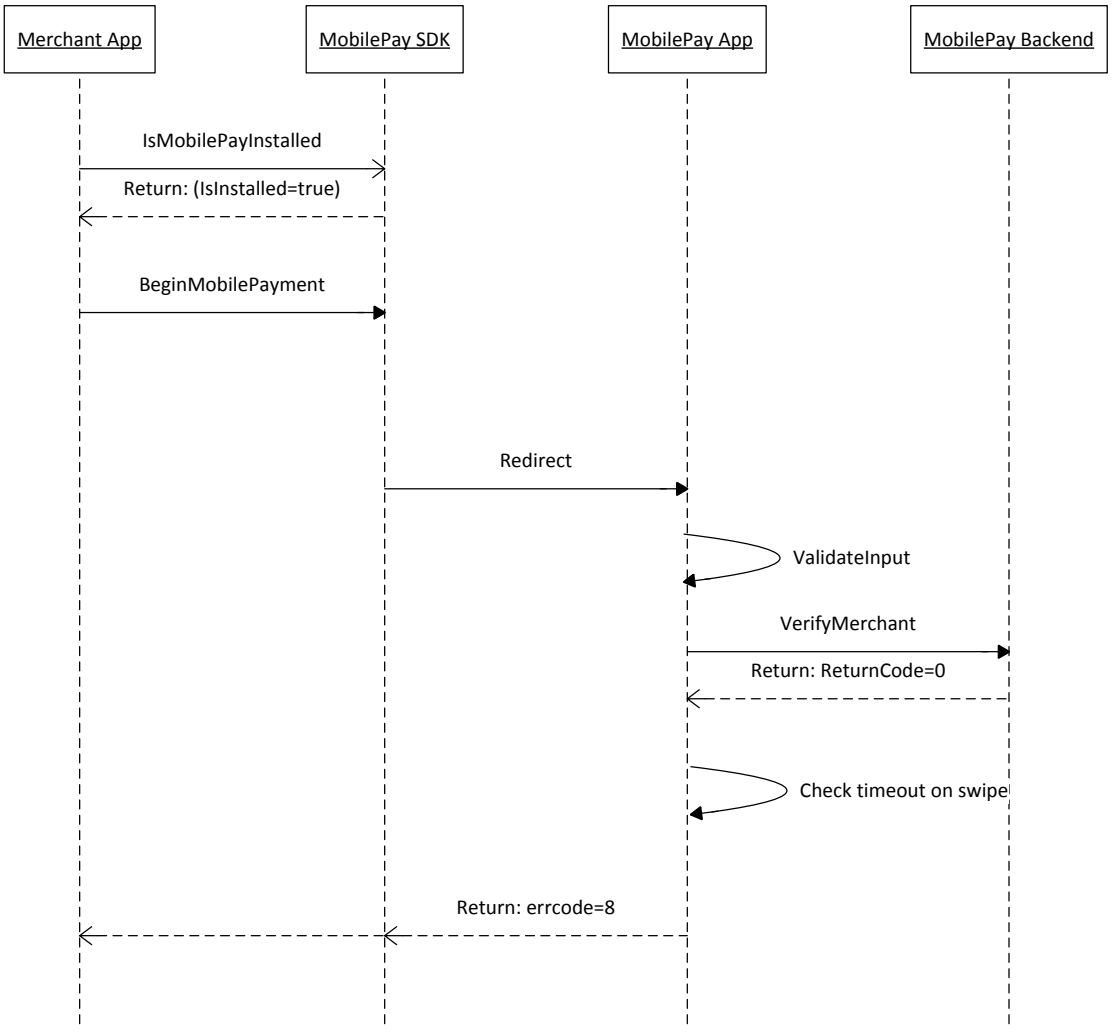


5.8 Timeout set in merchant app exceeded

The MobilePay AppSwitch SDK will return error code no. 8 to the merchant app if the purchase takes longer than defined by the merchant app. The default timeout is set to 5 minutes. The timeout will be checked in the MobilePay app when confirming the payment.

The merchant app should show a message informing that the user should try again before the timeout.

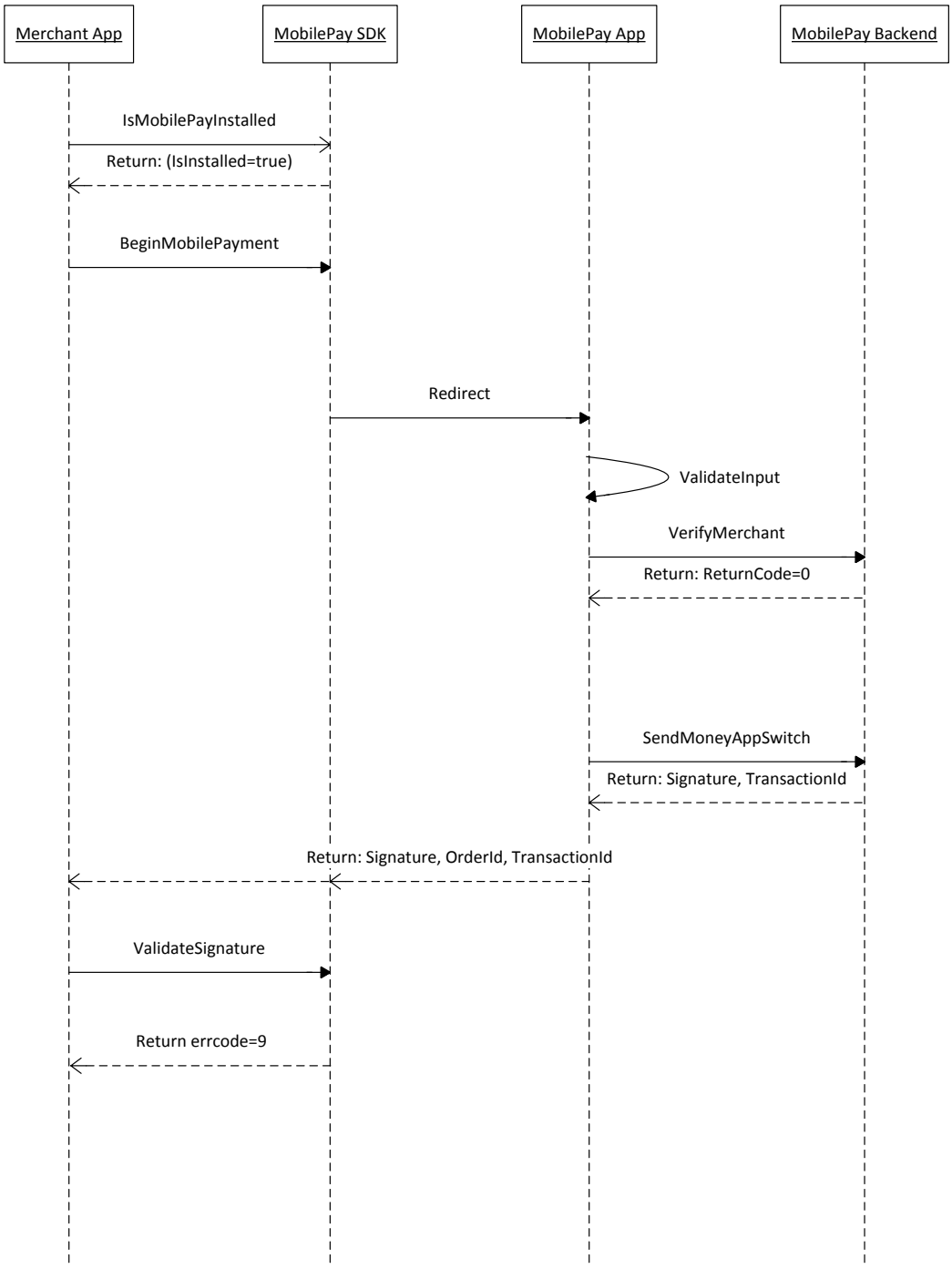
Example of return text: "MobilePay error 8: Timeout occurred. Please try again."



5.9 Invalid signature

The MobilePay AppSwitch SDK will return error code no. 9 to the merchant app if the SDK validation of the signature fails due to an invalid signature, or in the case that the same transaction id is sent to the merchant app twice after a single payment.

Example of return text: "MobilePay error 9: Invalid signature."



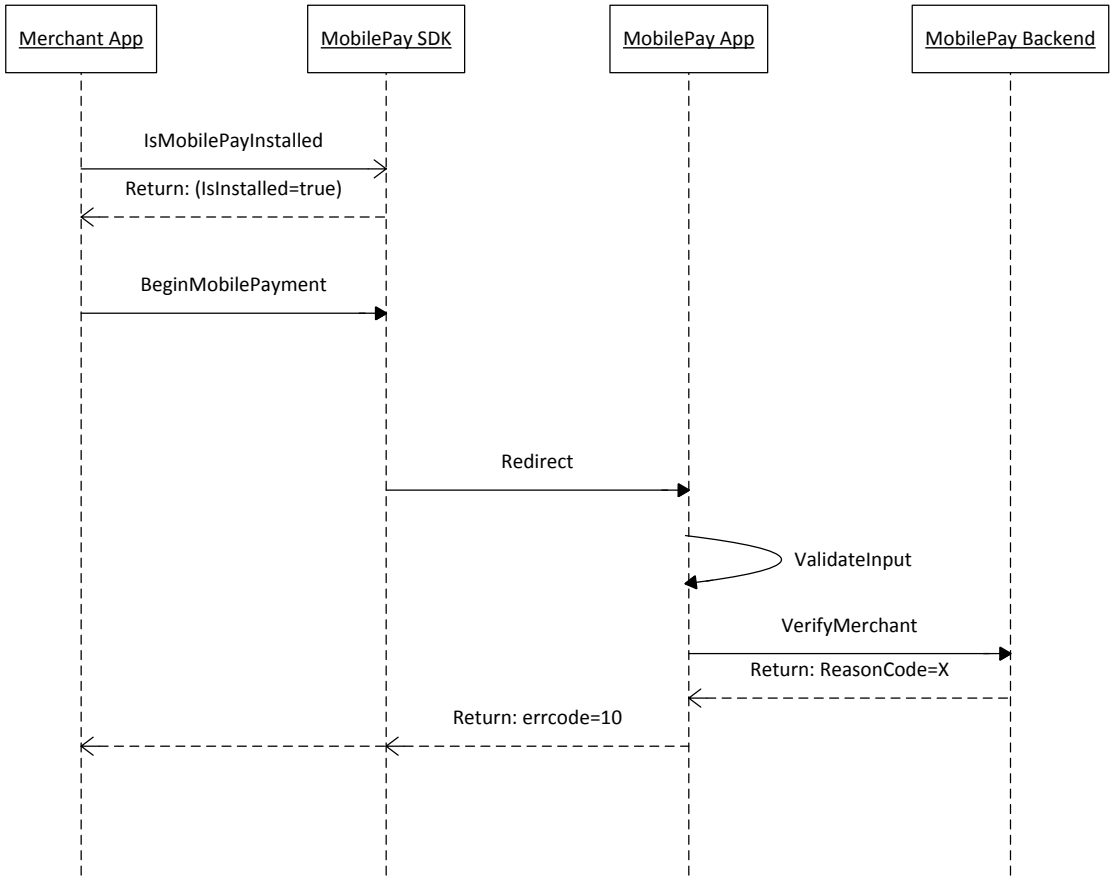
5.10 MobilePay AppSwitch SDK version is outdated

The MobilePay AppSwitch SDK will return error code no. 10 to the merchant app if the API version used by the SDK is declared obsolete by the MobilePay backend.

The merchant app receives this error code if the merchant app is not updated to the minimum required version of the MobilePay AppSwitch SDK.

The merchant app should show a message asking the user to update the merchant app.

Example of return text: "MobilePay error 10: Merchant app is not up to date. Please update and try again."

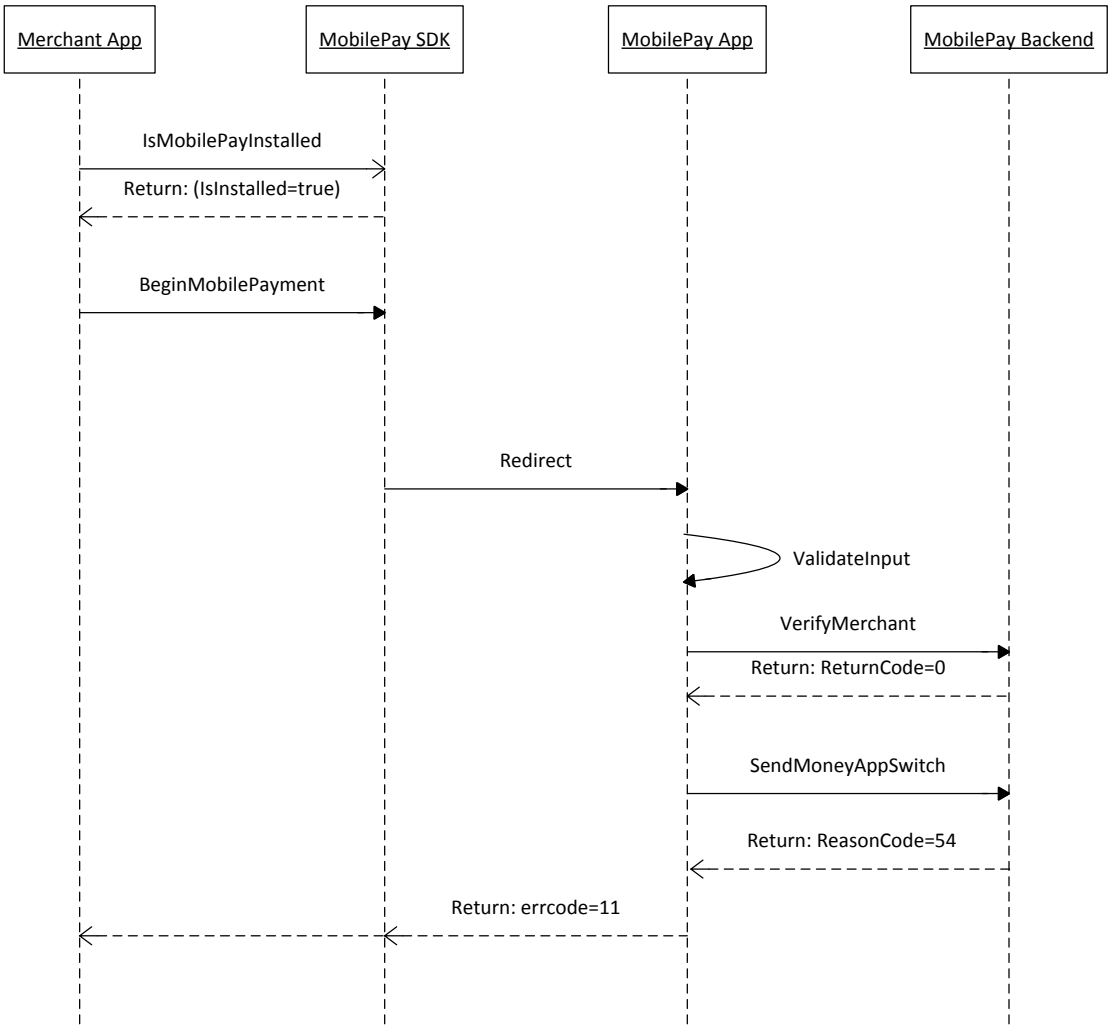


5.11 OrderID already used

Error code no. 11 will be returned to the merchant app if the orderID sent to MobilePay has already been used for a confirmed payment by the same merchant but not the current customer.

The merchant app should show a message to the user, and should create a new orderID attempting to make the payment again.

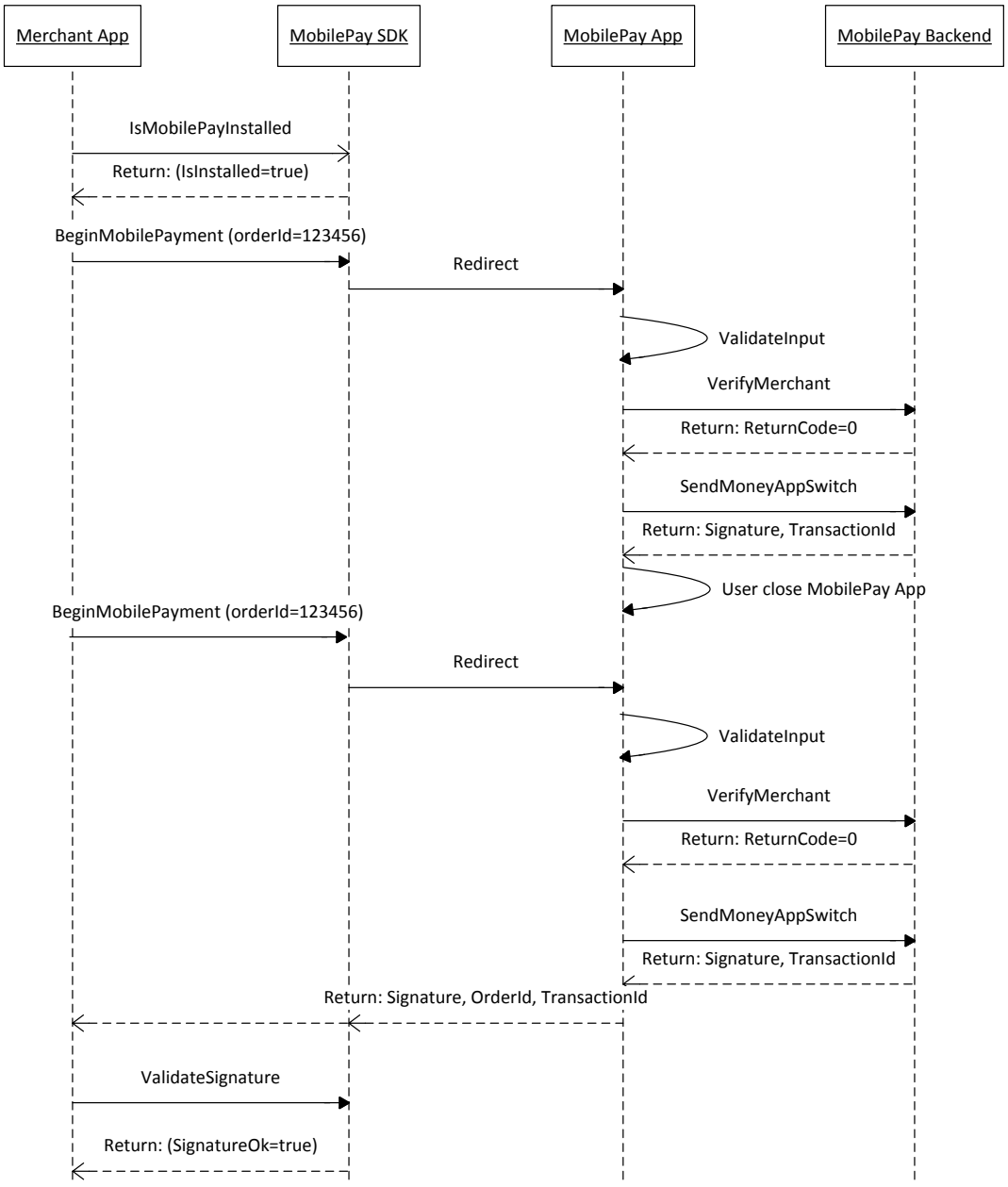
Example of return text: "MobilePay error 11: Error in payment data. Please try again."



5.12 Abandoned payment scenarios - MobilePay app is closed down while doing payment

In this case the merchant app will not receive a reply and this case can be handled as a normal timeout scenario.

The merchant app can choose to resend the same orderID to MobilePay, which will ensure that the customer will not pay twice for the same order and in all cases MobilePay will return the payment information, including the signature.

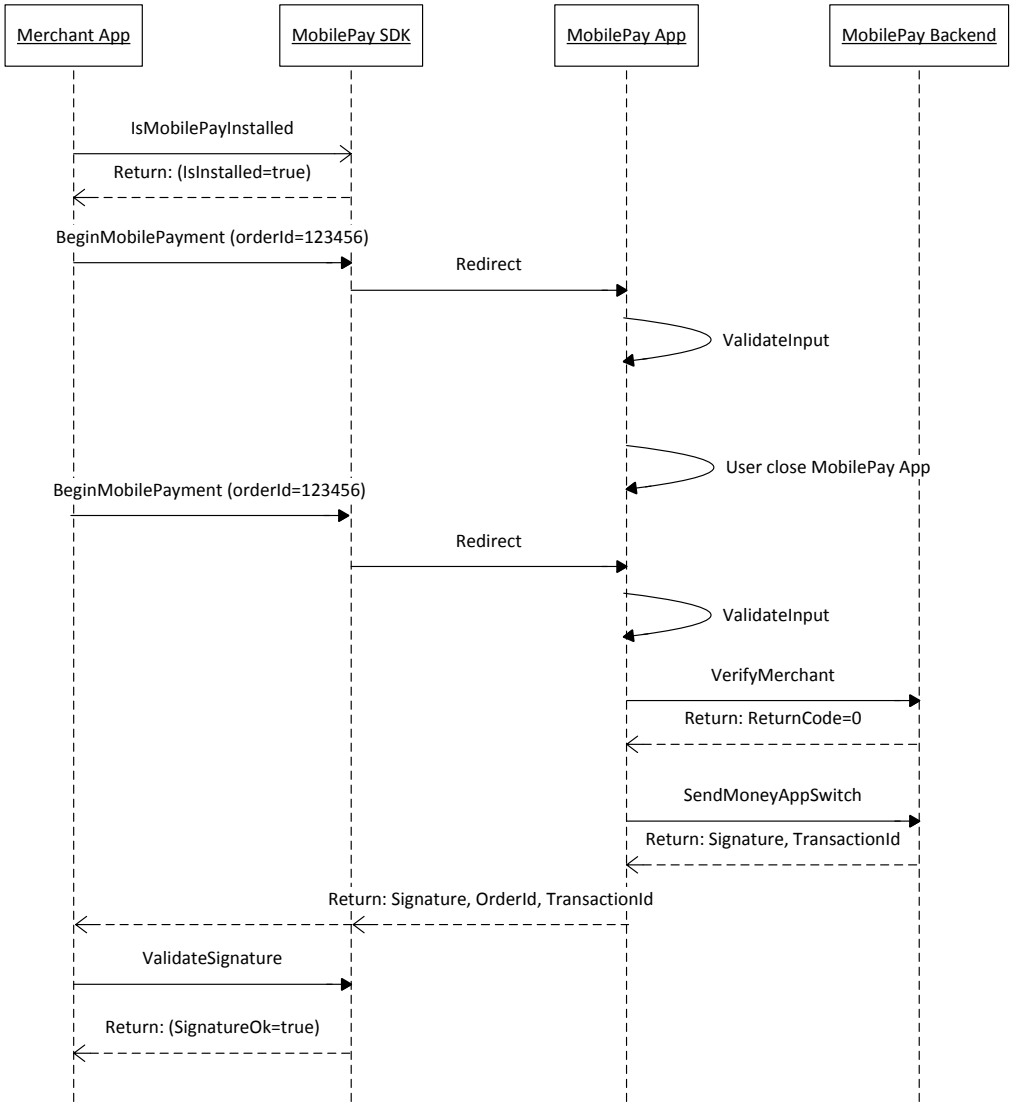


5.13 Abandoned payment scenarios - customer navigates away from MobilePay

The MobilePay app can be in two different modes, namely AppSwitch mode and normal payment mode. MobilePay is in AppSwitch mode when an AppSwitch payment is initiated and in normal payment mode when e.g. used in relation to P2P payments.

If the customer navigates away from MobilePay during an AppSwitch payment (e.g. the phone is answered) the payment flow is cancelled and the MobilePay app mode changes from AppSwitch mode to normal payment mode and the user is logged out.

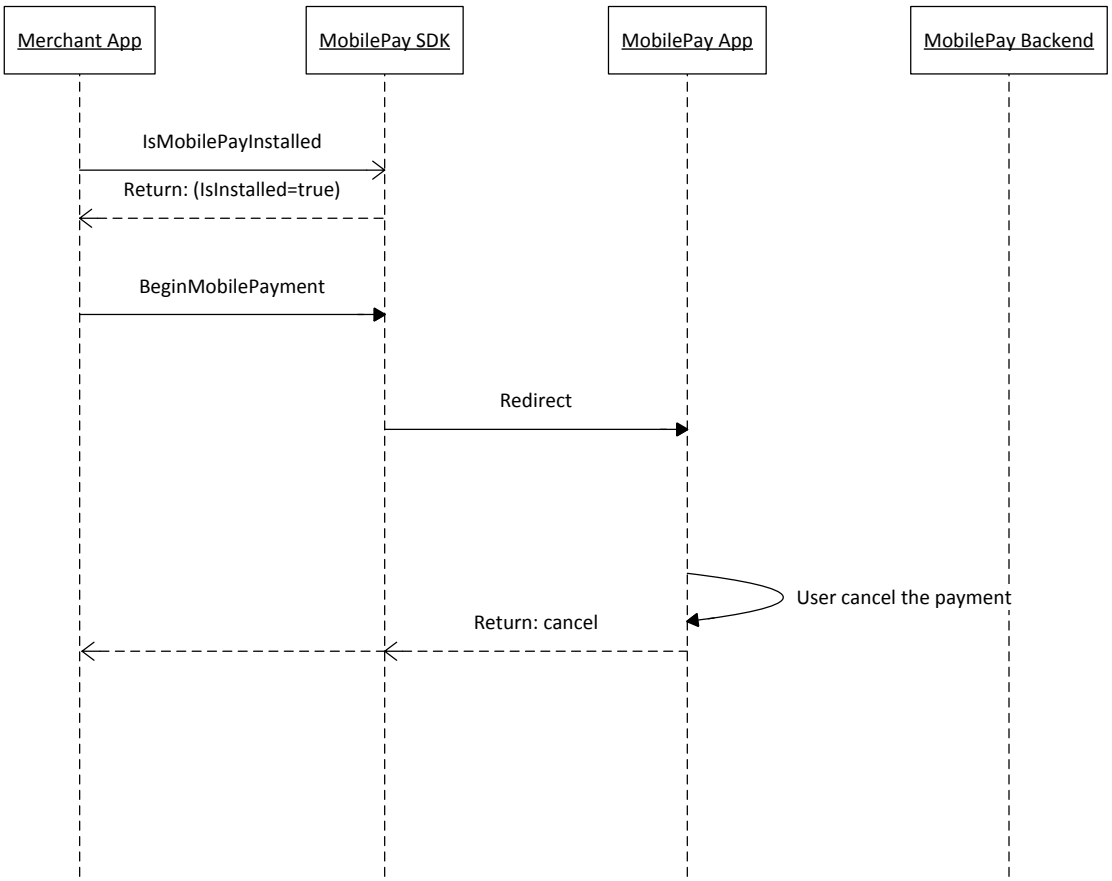
If the customer wants to pay after the phone call is completed the customer has to restart the payment from the merchant app.



5.14 Abandoned payment scenarios - payment is cancelled in MobilePay

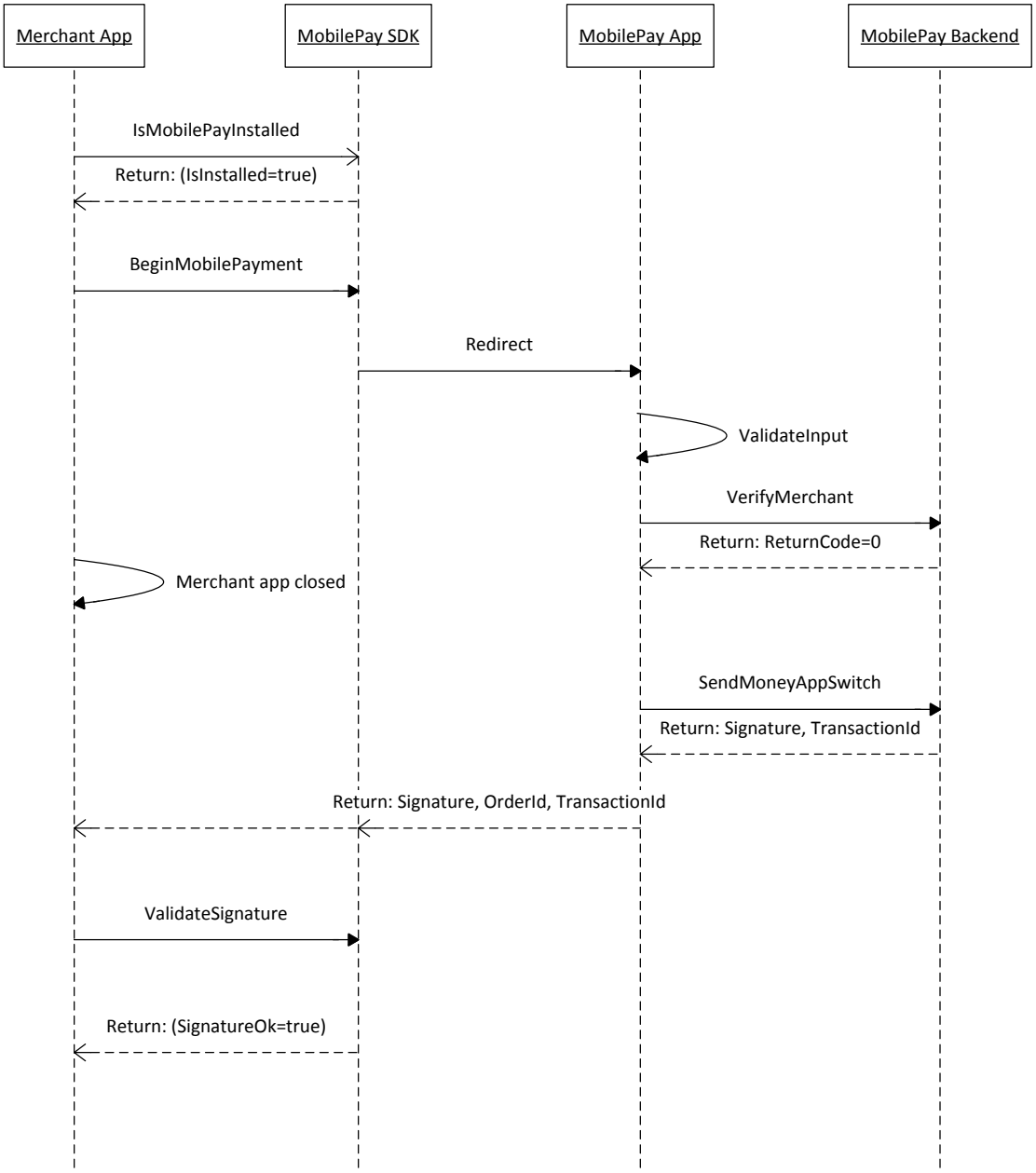
If the customer chooses to cancel the payment in the MobilePay app, the merchant app will be notified of the cancellation.

The merchant app can show a message to the user stating that the payment was cancelled and let the user continue shopping in the merchant app.



5.15 Abandoned payment scenarios - customer closes merchant app

If the customer closes the merchant app while doing a payment in the MobilePay app (e.g. using the multitask feature of the OS), the merchant app should also respond when receiving the call-back from MobilePay.



5.16 Abandoned payment scenarios - same orderID is sent to MobilePay twice

In case of network errors etc. it might happen that the merchant app resends the pending order (same orderID) to MobilePay, which the customer already has paid for.

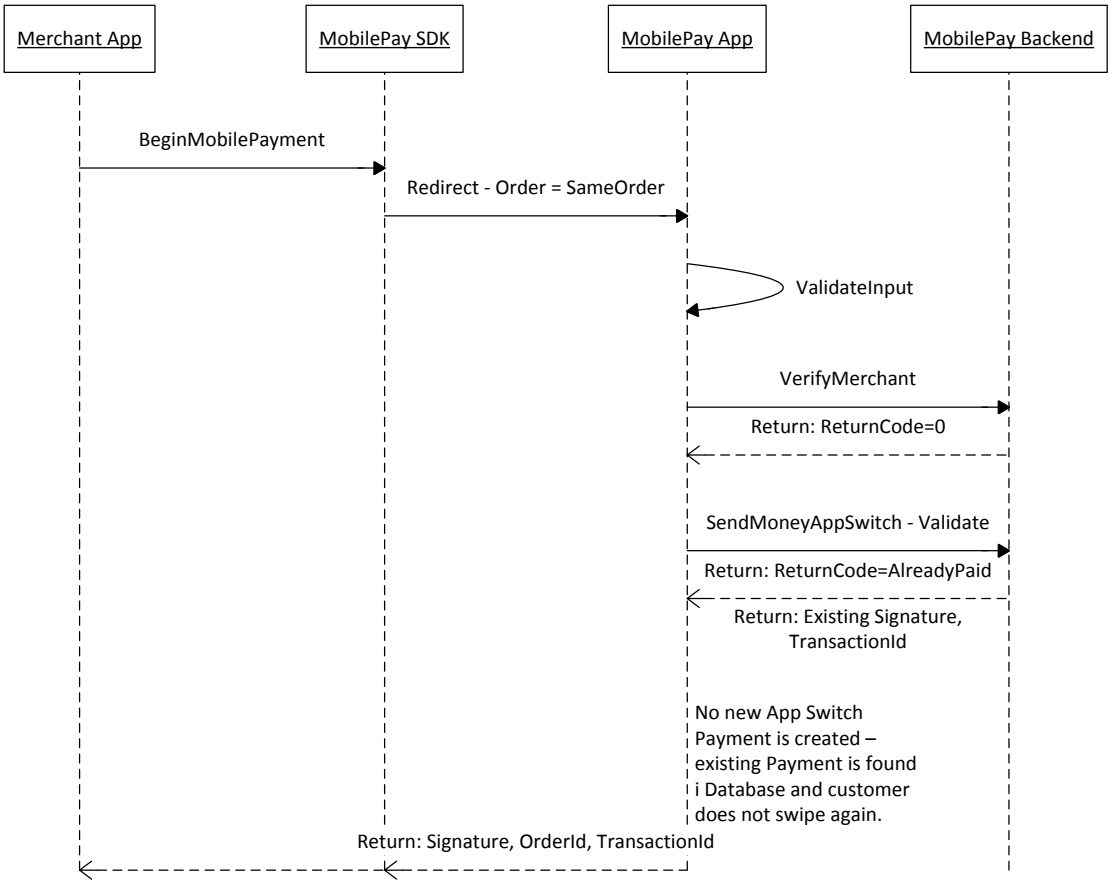
This error scenario is handled in MobilePay by returning the payment information to the merchant app letting the merchant app complete the order.

MobilePay uses a unique ID of the payment, which consists of the merchantID and the orderID. This ID is used to identify the payment transaction at DIBS and DIBS will reject the ticket authorisation the second time, if the same ID is used.

This means the customer will not be able to swipe for payment, but the MobilePay app will immediately show the receipt.

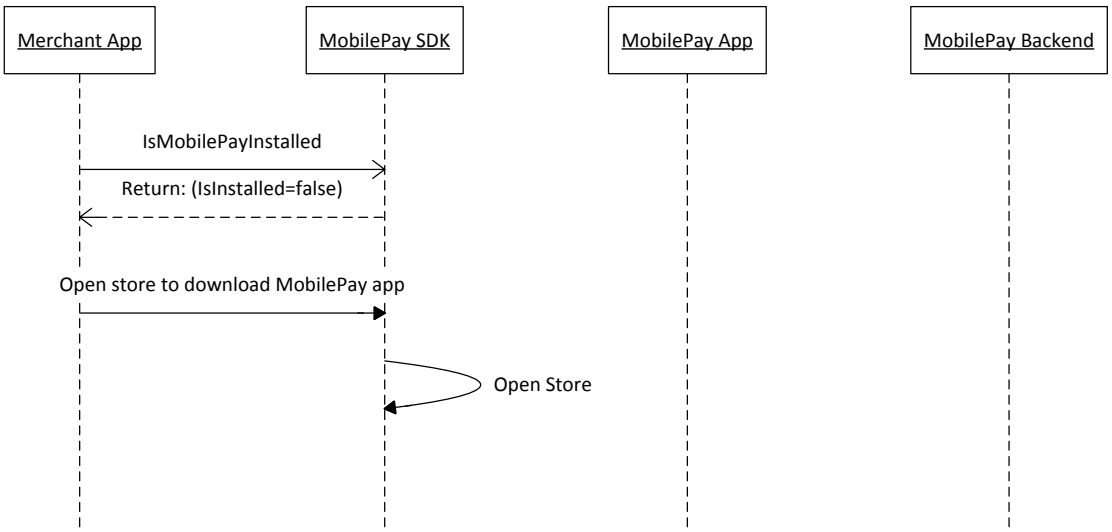
Please note that:

- Two redirects to MobilePay with same orderID returns signatures in both cases, but the customer has only paid once for this order. It must be ensured at merchant side, that it will not be possible to receive two services or products using the same orderID.


Merchant App
MobilePay SDK
MobilePay App
MobilePay Backend
BeginMobilePayment
Redirect - Order = SameOrder
ValidateInput
VerifyMerchant
Return: ReturnCode=0
SendMoneyAppSwitch - Validate
Return: ReturnCode=AlreadyPaid
Return: Existing Signature, TransactionId
No new App Switch Payment is created – existing Payment is found i Database and customer does not swipe again.
Return: Signature, OrderId, TransactionId

5.17 Installation issues - MobilePay is not downloaded

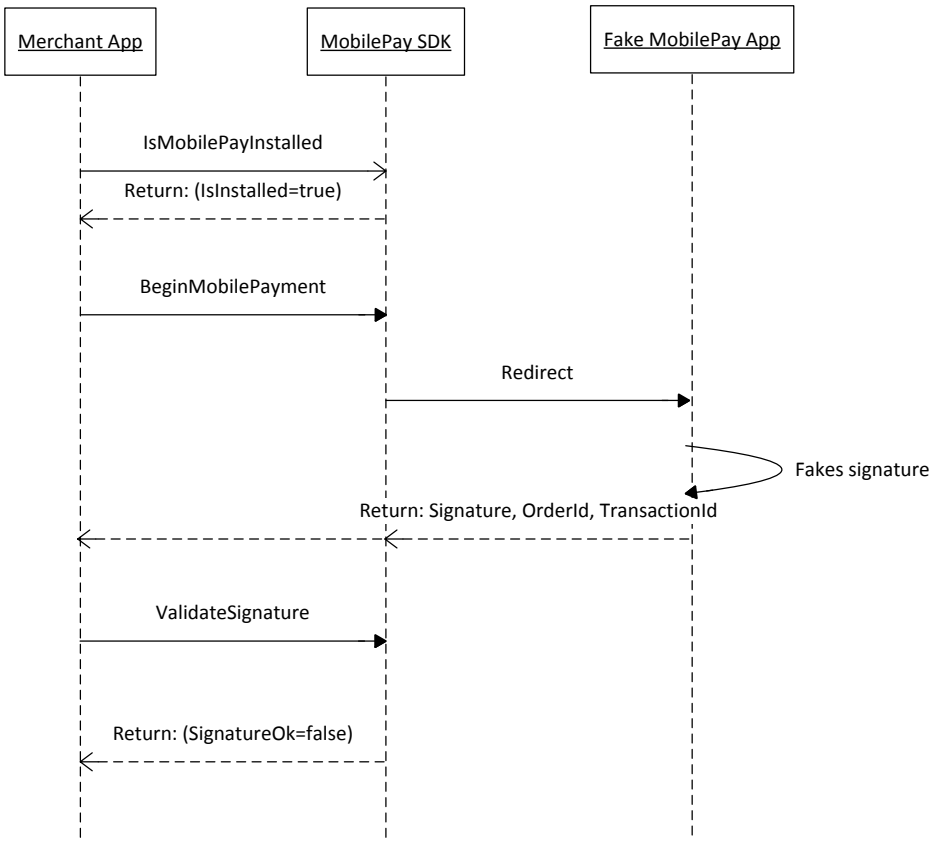
In cases where MobilePay is not installed on the customer's phone the merchant app can be setup to display an appropriate message. The SDK in GitHub has a method for checking if MobilePay is installed.



5.18 Installation issues - fake MobilePay app installed

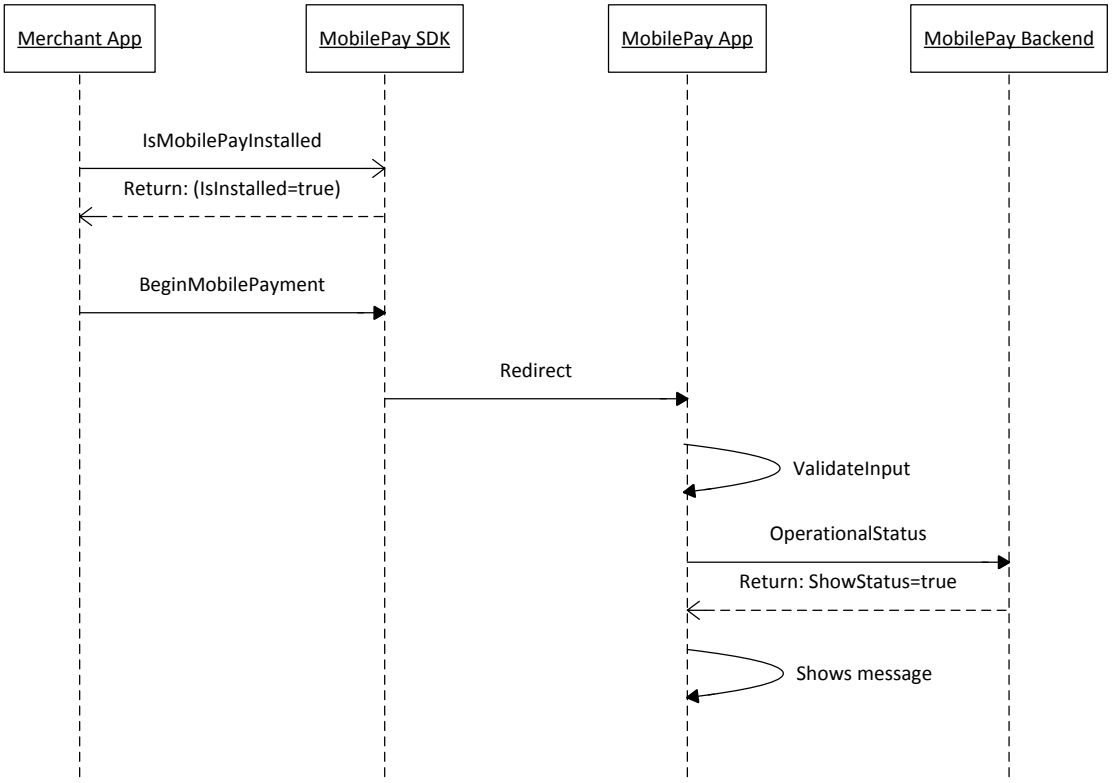
A fake MobilePay app will not be able to generate a valid signature.

Please also refer to section 4.4 “How to ensure authentication of payment”.



5.19 Installation issues - MobilePay is out of service

If MobilePay is out of service a notice will be displayed in the MobilePay app. In a case like this the customer can choose to cancel the transaction in MobilePay and the flow will then continue as explained in 5.14.



MobilePay AppSwitch Implementation Guide

6 Security and certificates

This section describes how the communication back and forth between the merchant app and the MobilePay app is secured and the usage of signatures and certificates.

The communication and security are both ensured by the MobilePay AppSwitch SDK on GitHub.

6.1 From merchant app to MobilePay app

The merchant app will deliver the following data to the MobilePay app. Current version of the SDK:

MerchantID	Char(60)
OrderID	Char(50)
Product Name	Char(40)
Product Price	Decimal
Receipt Message	Char(66)
SDK Version	Char(20)
Signature Version	Decimal
Success URL	Char(100)
Failure URL	Char(100)
Cancel URL	Char(100)
Capture (Y/N)	Char(1)
HMAC (data)	Char(64)

HMAC (data) is a SHA-256 hash value of all the preceding data. The MobilePay app will ask the MobilePay backend to verify the HMAC on this data. The key for the HMAC calculation is stored in the merchant app (MobilePay AppSwitch SDK part) and in Danske Bank Backend (HMAC key agreed upon between merchant and Danske Bank).

6.2 From MobilePay app to merchant app

If the parameter Capture specified in the input is 'Y', the MobilePay app will send the following data to the merchant app:

OrderID	Char(50)
MerchantID	Char(60)
TransactionID (PaymentID)	Char(20)
Signature	Char(2500)

And the signature will contain these data:

Signature version 2.0	
OrderID	Char(50)

MerchantID	Char(60)
TransactionID (PaymentID)	Char(20)
Amount	Dec(15,2)
Currency	Char(3)
Country	Char(2)

The MobilePay AppSwitch SDK will verify that the MSIGN certificate in the signature is issued by the Danske Bank root certificate. It is important that the root certificate is used for signature validation; otherwise this solution will stop working when a new MSIGN certificate is issued.

It is imperative that the Danske Bank root certificate is exchanged securely. Initially, the root certificate has to be exchanged manually, but at a later stage an automatic setup will be in place. NOTE: this automatic setup requires development from the providers of the merchant app.

The private key of MSIGN is securely stored at Danske Bank.

The signature is validated using the Danske Bank root certificate in the following way:

The signature is checked that it is signed by a certificate that has a valid certificate chain to the Danske Bank root certificate. It is also validated that the signature is comprised of the expected orderID, merchantID, transactionID, and amount.

6.3 Security from MobilePay app to MobilePay backend at Danske Bank

The data sent from the MobilePay app to the MobilePay backend is sent over HTTP with SSL.

When a user logs on the MobilePay app the user will have a session with the MobilePay backend. This session is created using the cryptographic encryption algorithms AES and RSA.

6.4 Data at Rest

The input that the MobilePay app receives from the merchant app is stored temporarily, i.e. no data is persistent in the MobilePay app.

All payment and customer data is stored in the Danske Bank MobilePay backend (DB2 database).

MobilePay AppSwitch Implementation Guide

7 MobilePay AppSwitch SDK updates

The SDK updates according to the following scheme: {MAJOR}.{MINOR}.{PATCH}. Within a given version number category, each number is assigned in an increasing order which means that version 1.4.3 is newer than 1.4.2.

The {MAJOR} number is increased when there are significant jumps in functionality or changes to the framework which could cause incompatibility with interfacing systems.

The {MINOR} number is increased when minor features or significant fixes have been added.

The {PATCH} number is increased when minor bugs are fixed.

8 Test setup

It is not possible for merchants to communicate with the Danske Bank test environment. Testing must therefore be done in the production environment.

The test merchantID "APPDK0000000000" can be used for test purposes. When the test merchantID is used it is possible to complete the payment flow without transferring any money. This means the merchant is able to test the security setup, SDK etc. without creating any payments.

When the merchant wants to test reconciliation files, the merchant must use the production merchantID in order to create real payments. However small amounts of e.g. 0,01 DKK can be used in this test setup.

A test version of the MobilePay app does exist at tpa.trifork.com. However, access to this site is controlled by Trifork alone, and interested parties must therefore direct enquiries to Support Erhverv in order to gain access.

9 Key Terms & Definitions

Terms	Definitions
AES	Advanced Encryption Standard (AES) is a specification for the encryption of electronic data established by the U.S
Alias	See Merchant Id
DIBS	Dansk Internet Betalings System
Data at Rest	Data at Rest is used as a complement to the terms Data in Use and Data in Motion[1] which together define the three states of digital Data.
GitHub	GitHub - Code sharing service. It is a Git repository web-based hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features.
HMAC	Hash Message Authentication Code is a specific construction for calculating a message authentication code (MAC) involving a cryptographic hash function in combination with a secret cryptographic key.
HMAC Message	The message the HMAC calculation is based upon
HMAC Key	Agreed upon (between sender and receiver) key used for HMAC calculation
HTTP	Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web
Merchant App	Synonym for an app that involves payment transactions - typically related to selling of goods or services.
Merchant Id	Merchant identification number - A unique merchant ID provided by Danske Bank.
MSIGN	Name of signing public key pair for MobilePay signatures. The MSIGN public key is included in a certificate issued by the Danske Bank root CA.
Order Id	Order identification number - Here referring to the unique provided Order ID (Shop's order ID) sent along with a payment request from a merchant (app) to MobilePay (app)
Payment Id	See Transaction Id
P2P	Peer-to-peer payment
REST	Representational state transfer (REST) is a simple stateless architecture that generally runs over HTTP. REST is used between the MobilePay app and the MobilePay backend.
RSA	RSA is one of the first practicable public-key cryptosystems and is widely used for secure data transmission. RSA involves a public key and a private key. The public key can be known by everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted in a reasonable amount of time using the private key.
SDK	Software Development Kit – MobilePay AppSwitch SDK is designed to be embedded in a merchant app.
SIM	Subscriber Identification Module
SSL	Secure Sockets Layer- a standard cryptographic protocol designed to provide communication security over the Internet.
Transaction Id	DIBS provided payment transaction ID
UI	User Interface