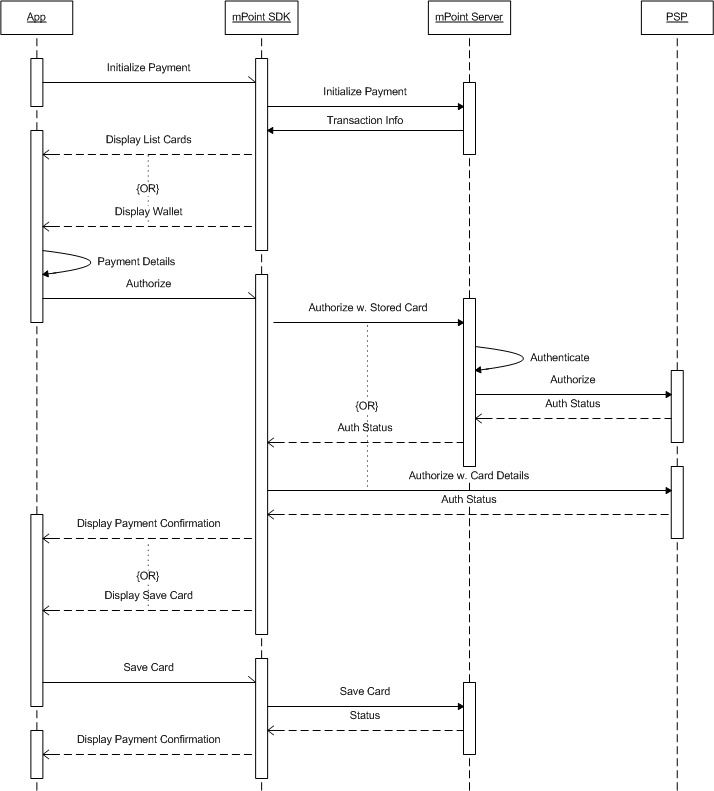
mPointTM – Native SDK

mPointTM’s native SDK has been designed using an asynchronous delegate architecture where the Mobile App implements and Interface / Protocol, which is used by the mPointTM SDK to asynchronously pass data back.

The sequence diagram below illustrates a standard flow for authorizing a payment using either a new card by passing the card details directly from the App to the Payment Service Provider (PSP) or a stored card by calling the mPointTM Server which in turn authorizes the payment with the Payment Service Provider.



The architecture illustrated by the sequence above has been designed to ensure that the merchant does not fall under PCI DSS.

# Project Configuration

To utilize the mPointTM SDK the binary files must be linked into the app. iOS developers will need to add the following frameworks to the project:

* CellPoint Mobile
* mPoint

Android developers will need to place the mpointsdk.jar file in the libs directory and add a reference to the JAR file to the Java build path.

# mPointTM SDK Methods

The mPointTM SDK includes a number of methods, which are used to complete a payment transaction. The available methods are briefly described in the table below.

|  |  |  |
| --- | --- | --- |
| Name | Overloaded | Description |
| initWithURL / mPoint | Yes | The constructor used to instantiate an mPoint object |
| initialize | No | Initializes a payment on the mPointTM server. The method may invoke one of the following callback methods depending on the transaction information in the response:  **displayCardList** - called when the app should display the list of available payment cards that the customer may choose from  **displayWallet** - called when the app should display the list of the customer’s stored cards. |
| authorize | Yes | Authorizes a payment for the transaction by providing the mPointTM server with a reference to the customer’s stored card or the Payment Service Provider with the entered card details. The method may invoke one of the following callback methods depending on the authorization status:  **handleStatus** - called when the authorization was declined by either the mPointTM server or the Payment Service Provider.  **displayPaymentConfirmation** - called when the authorization was approved.  **displayCreateAccount** - called when the authorization was approved and the customer has checked the “Store Card” checkbox.  **displaySaveCard** - called when the authorization was approved and the customer has checked the “Store Card” checkbox and the customer already has an account on the mPointTM server. |
| saveCard | No | Provides a name for the saved card. The method may invoke one of the following callback methods depending on the save status:  **handleStatus** - called when the operation was rejected by the mPointTM server.  **displayPaymentConfirmation** - called when the operation was accepted by the mPointTM server. |
| saveAccount | No | Provides a password for the customer’s mPointTM account and optionally a name for the saved card. The method may invoke one of the following callback methods depending on the save status:  **handleStatus** - called when the operation was rejected by the mPointTM server.  **displayPaymentConfirmation** - called when the operation was accepted by the mPointTM server. |