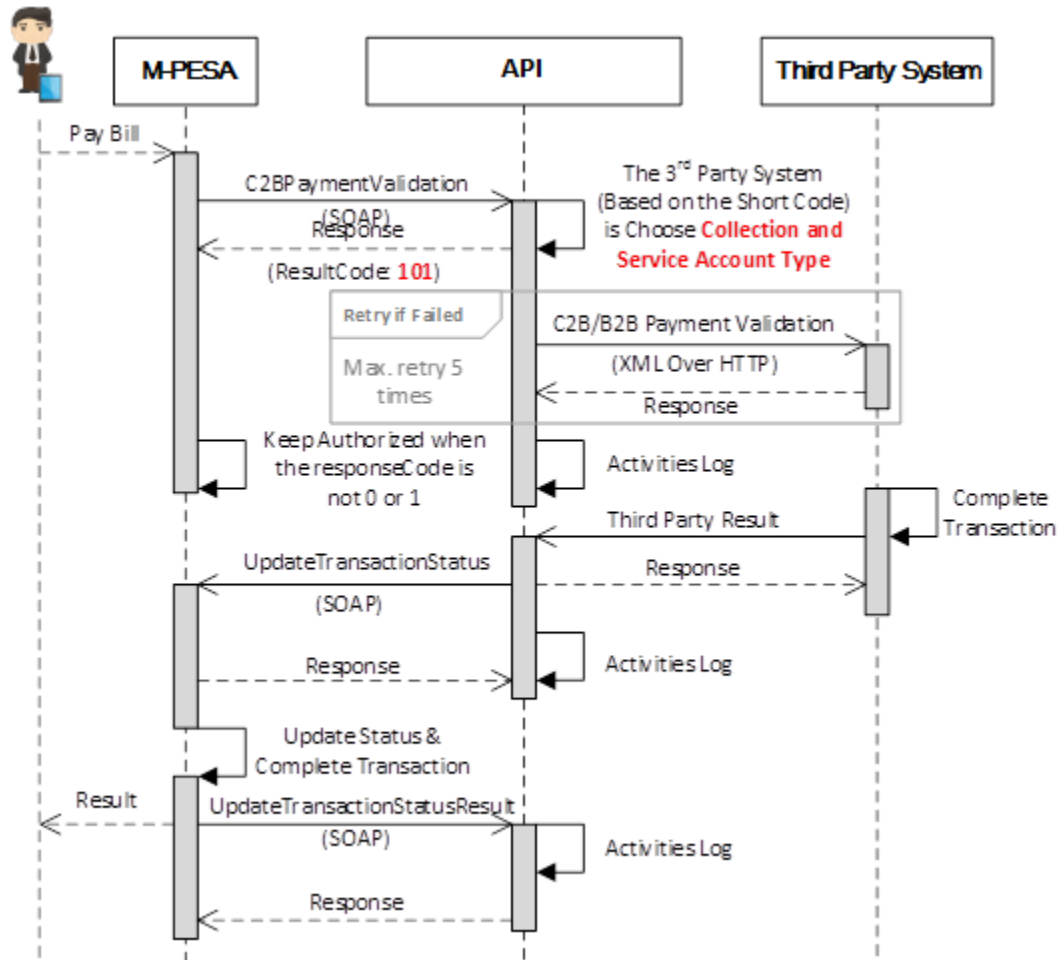


## C2B TRANSACTIONS CALL FLOW

This section describes how the M-Pesa API Platform communicates with M-Pesa core through SOAP interface, and communicates with third party system through XML-over-HTTP and FTP interface.



From the above diagram, the steps are broken down as follows

1. M-PESA will send the C2B transaction request to third party with the format below

```

<?xml version="1.0" encoding="UTF-8"?>
<mpesaBroker xmlns="http://infowise.co.tz/broker/" version="2.0">
  <request>
    <serviceProvider>
      <spId>400205</spId>    --this is your business number
      <spPassword>29N2NizUAIVCgSnsuaz+SSqjWgpYGFhnYtV4GfJrI8c</spPassword> -- this is password needs to be encrypted as explained below
      <timestamp>20180221124032</timestamp> --- this is time used to encrypt the sp password
    </serviceProvider>
    <transaction>
      <amount>1800.00</amount> -- this is the amount to sent by the customer
      <commandID>Pay Bill</commandID> -- this is command id to show that it is paybill
      <initiator>255742602260</initiator> -- this is Customer number made the request
      <originatorConversationID>025d7efd-58bc-4f15-b06b-2aab91cde3b1</originatorConversationID>
      <recipient>400205</recipient>
      <mpesaReceipt>5BL716QNJB</mpesaReceipt> -- this is M-PESA receipt
      <transactionDate>2018-02-21 12:40:27</transactionDate> -- date of the request
      <accountReference>255742602260</accountReference> -- The reference number of the payment
      <transactionID>125189974</transactionID>
      <conversationID>025d7efd-58bc-4f15-b06b-2aab91cde3b1</conversationID>
    </transaction>
  </request>
</mpesaBroker>

```

## 2. Third party will respond with synchronous to M-PESA

Upon the reception of the request, third party needs to respond immediately before performing any request check. Below is the format of the response

```

<?xml version="1.0" encoding="UTF-8"?>
<mpesaBroker xmlns="http://infowise.co.tz/broker/" version="2.0">
  <response>
    <conversationID>025d7efd-58bc-4f15-b06b-2aab91cde3b1</conversationID>
    <originatorConversationID>025d7efd-58bc-4f15-b06b-2aab91cde3b1</originatorConversationID>
    <transactionID>125189974</transactionID> -- this is M-PESA transaction ID from the request received
    <responseCode>0</responseCode>
    <responseDesc>Received</responseDesc>
    <serviceStatus>Success</serviceStatus>
  </response>
</mpesaBroker>

```

Responsecode needs to be 0 always

ResponseDesc needs to be received always

Servicestatus needs to be success always

### 3. Transaction processing

Third party will process the transaction based on his/her rules but the following need to be party of those rules

- Transaction duplicate checking- Every request that third party receives must be checked if it is duplicate or not, this can be achieved by checking M-PESA receipt since it is unique. If transaction is duplicate third party needs to initiate the callback with the status of the transaction in third party system, if it is not third needs to accept it as new transaction and do the rest of processing as usual.
- Below is the format for the callback

```
<?xml version="1.0" encoding="UTF-8"?>
<mpesaBroker xmlns="http://infowise.co.tz/broker/" version="2.0">
  <result>
    <serviceProvider>
      <spId>888999</spId>
      <spPassword>ynVEEr+d1I9yxQZuXaGI48azUeYA6KgWeCw/TwFzoQQ=</spPassword>
      <timestamp>20180252150805</timestamp>
    </serviceProvider>
    <transaction>
      <resultType>Completed</resultType>
      <resultCode>0</resultCode> ---- 999 for failure or 0 for successfully
      <resultDesc>Successful</resultDesc> failure
      <serviceReceipt>5BL216R1ELM</serviceReceipt>
      <serviceDate>2018-02-52 15 08 05+0300</serviceDate>
      <serviceID>6787</serviceID> --bet transaction ID
      <originatorConversationID>2c7566b8-166d-4c96-9389-fb4880573151</originatorConversationID>
      <conversationID>2c7566b8-166d-4c96-9389-fb4880573151</conversationID>
      <transactionID>125209081</transactionID> --THI IS OUR ID
      <initiator>ibm_in</initiator>
      <initiatorPassword>23qhi33akBYIgyBA==</initiatorPassword>
    </transaction>
  </result>
</mpesaBroker>
```

Resulttype- this will be completed if the transaction is successful and failed if the transaction is unsuccessful.

Resultcode – this will 0 if transaction is successful or 999 if transaction is unsuccessful

#### 4. Synchronous response from M-PESA.

Upon reception of callback M-PESA will respond to third party to mark the reception of the callback with the format below

```
<?xml version="1.0" encoding="UTF-8"?>
<mpesaBroker xmlns="http://infowise.co.tz/broker/" version="2.0">
  <response>
    <conversationID>9af2888b-0d71-4629-8967-2ffec646018e</conversationID>
    <originatorConversationID>9af2888b-0d71-4629-8967-2ffec646018e</originatorConversationID>
    <responseCode>0</responseCode>
    <responseDesc>Attempted successfully</responseDesc>
    <serviceStatus>Confirming</serviceStatus>
    <transactionID>125190393</transactionID>
  </response>
</mpesaBroker>
```

Response code will always be 0

Response desc will always be Attempted successfully

serviceStatus – this will depend on the result code in the callback if the result code was 0 it will be Confirming if the result code was 999 it will be cancelling.

#### SP PASSWORD ENCRYPTION

The sp password hashing is done by using the algorithm below

##### PHP

```
$encryptedPasswordProvider = base64_encode(hash("sha256", $spid.$spPassword.$timestamp, True));
```

Where \$spid=spid from the request received from M-PESA.

\$spPassword=the plain sp password, you will be given this during integration

\$timestamp=timestamp you used to encrypt the password in this format YYYYMMDDHHMISS .

##### JAVA

```
public String getSPPassword(String spId, String password, String timestamp) throws
NoSuchAlgorithmException {
```

```
    String dgStr = spId + password + timestamp;
    byte[] enc = MessageDigest.getInstance("SHA-256").digest(dgStr.getBytes());
```

```
    return new String(Base64.encodeBase64(enc));
```

```
}
```

