# 

[Use Cases](#_r4yvmqk5awl5)

[Real-time Disburse Use Case](#_h5ps78etjw6v)

[Batch Disburse Use Case](#_2ct6r55oxsng)

[Implementation](#_uyh3u2ssjpg6)

[Real time disbursement](#_mqp1nurv8vez)

[Batch disbursement](#_r0njbbm6gqfq)

[Customer initiated payments](#_jvoggrpst0w8)

[Security](#_9p4wlur04ma5)

[Database Design](#_bxc2ajbnih39)

[Meeting Notes](#_fnnbzks0yrt7)

[Sat Jun 9, 2017 -- Meeting notes](#_drajz7fzuxaa)

[Fri Jun 2, 2017 -- Meeting notes](#_zb865da3vok4)

[Wednesday, 17 May, 2017 11:34 follow-up notes](#_pqzr80wiopum)

[Wednesday, 17 May, 2017 11:01 Ed’s notes](#_ow4xqp6aabo7)

[Wednesday, 17 May, 2017 11:01 notes](#_l4l0hr318kqj)

# 

# Use Cases

### Real-time Disburse Use Case

1. In Mifos, as the loans are disbursed there will be an event fired in the background to queue up these transactions.
2. The transactions will be further forwarded to MMP for processing.
3. MMP will return status of the transactions via callback URL. MMP will notify customers of this transaction.
4. If successful, the disbursal is made and nothing needs to be done in Mifos.
5. Unsuccessful events will be rollbacked (undo disbursals) via automatic job or via MFI signing into mobile money portal, reviewing the failed transactions and manually hitting a ‘process’ button to reverse the disbursals.

### Batch Disburse Use Case

1. Mifos - At the end of the day, branch manager or operations manager will export all the disbursements in an excel.
2. The report will have parameters by date, by office, **by payment type**, etc.
3. If there are more than one Mobile Money Provider(MMP)/payment type, MFI will need to run the report multiple times – one excel per MMP.
4. Mobile Money Portal – MFI will sign in to upload the excel files. Or they will have a process button in the report to call MMP Bulk process.

It should have authentication for MFI users based on MFI identifier – use Mifos api.

MFI user login will have

* Upload File Screen. He/she will choose Payment type in a drop-down and upload the file. The file should be saved in corresponding destination folder for MMP to pick up.
* View File History – parameters (date, user, payment type, etc)
* View Raw Status Files from MMP – file name, date, who uploaded, which ip
* View Detail Status File. Failed transactions will be grouped at the bottom and MFI can click to a button to reverse disbursement to Mifos for failed transactions.
* The status of Mifos api reverse transactions will be saved and displayed.
* Report for each day requests (total amount)
* Report for each day receipts (total amount by status)
* Report for totals by date (inbound and outbound)
* Audit – user (and IP) who uploaded the file, user (and IP) who process the reverse disbursements

Mobile Money Portal – MMP will sign in and process the files.

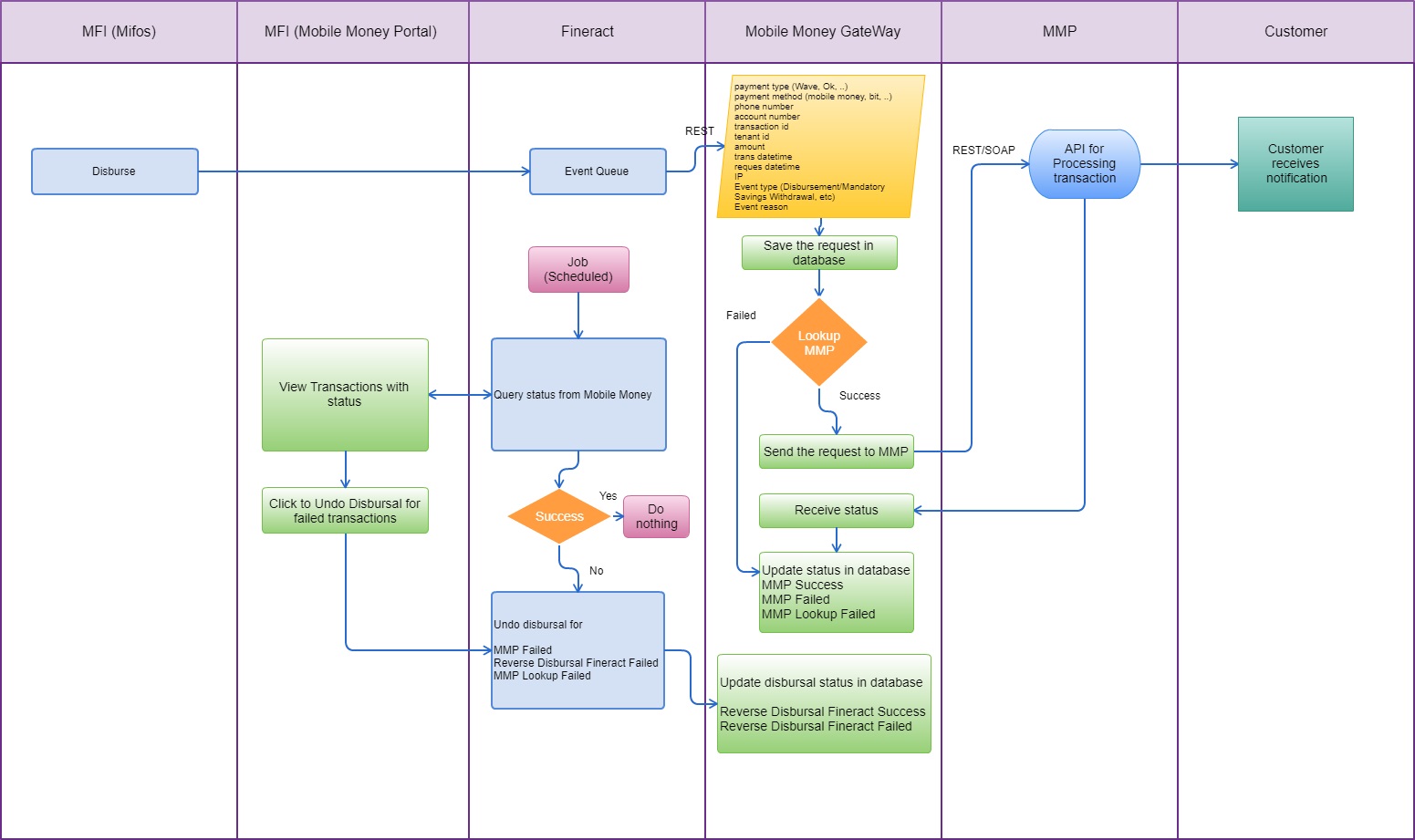
It should have authentication module.

MMP will have

* Upload File Screen – for uploading the processed files with status. They will navigate to the MFI they have access to.
* View Status File History
* View Raw Submitted Files.
* They get to see the status of the file whether it’s processed or not.
* Report for each day requests (total)
* Report for each day receipts (total)
* Report for totals by date (inbound and outbound)
* Audit – user (and IP) who uploaded the file, user (and IP) who process the reverse disbursements

# Implementation

### Real time disbursement



### 

### Batch disbursement

### Customer initiated payments

# Security

# Database Design

PK - Primary key

FK - foreign key

# Outbound\_transactions table is used to store all the request we get during a disbursement.

Incoming\_client\_request table stores the request a client makes to mifos through the mobile money gateway when he wants to save or repay a loan

Inbound\_transactions table records information about a client’s request after the MMP has processed the request on its end and send it to the mobile money gateway.

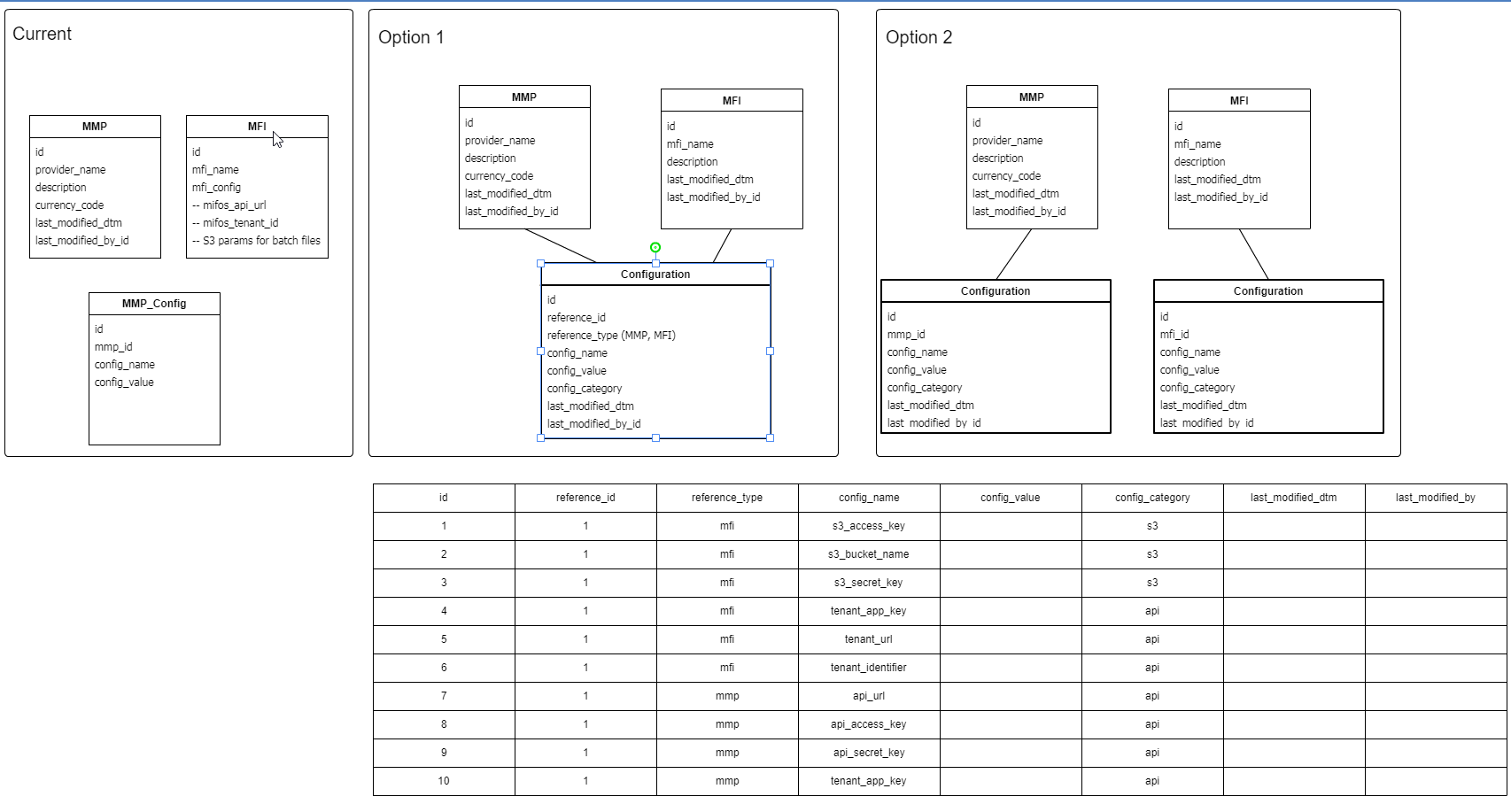
Mobile\_money\_providers table stores the list of mobile money providers registered in the gateway.

Provider\_reports stores the reports from the mobile money provider.

Alternative Design:



# 

**Normalizing of Config**

# Meeting Notes

### Sat Jun 9, 2017 -- Meeting notes

**Disbursement**

* Real Time mode -- through a queue via event
* Schedule mode -- through a scheduled job like other jobs we have in Mifos.
* Batch mode -- through a screen/place where MFI will upload a file or see a list of all transactions and hit process to send the transactions to Mobile money gateway.

All these events will be sent to Mobile Money Gateway which in turn will send them to MMP.   
Nazeer to provide parameters to set in the job so that the same job won’t be invoked at the same time on the application servers (load balancing scenario).

Mifos will have a job to query the status column in Mobile Money Gateway to decide whether or rollback the disbursal (only roll back for failed transaction).

**Funds from customers (Loan repayment or saving deposits)**

MMP will invoke two Mobile Money Gateway apis for this transaction.  
1) To query the balance by account

2) To make the actual payment. When the payment is made, the gateway will queue this transaction and return to MMP api that the request was good and received. There should be a call back URL from MMP so that we can reply the actual success/failure of posting in Mifos. Asynchronous process.

Thynn to find out if MMP can support such call back URL.

**Security**

How will the security be handled?

Oauth or Basic auth?

Anthony will do more research on this. Whether or not to follow SMS design.

If Oauth, how to distribute security key.

**Database Design**

Make it so it can handle million of requests. Partition by tenant?

**API**

Ionic framework

### Fri Jun 2, 2017 -- Meeting notes

**Funds initiated by MFIs (such as disbursement or withdrawal from savings, etc.), the transactions will take place in Mifos.**

Methods:

**Option 1: Real-time (preferred method)**

* use notifier in Fineract to listen in the events and make api calls to Mobile Money Gateway which will further process the requests. Moblie Money Gateway will have logic for invoking MMP specific apis (based on payment type) to perform these transactions.
* a detailed log of what was sent to MMP gateway from which server ip/api, etc should be saved in Mobile Money Gateway.
* MMP api will return success or failed.
* In Mobile Money Gateway, the status and reason for MMP API failure will be saved.
* MFIs will have a chance to review these transactions and decide whether to reprocess or reverse. This screen (for reconciliation) belongs to Mobile Money Portal along with all other Mobile Money Functions/Reports.
* In Fineract, there will be a job to pull these transactions by status (such as #3 below) periodically and process them through queue (not API calls).
* -- The status of these reversal transactions should be post back to Mobile Money Gateway.

Suggestions for status – we can rephrase them.

1. MMP success à the fund is at destination.

2. MMP failed –> The fund has not been reflected back in Mifos yet.

3. Ok to reverse à MFI okay’d to reverse them (\*\* this Is to give MFIs a chance to see what happened)

4. Set to retry à MFI wants to retry the failed ones.

5. Fineract reverse disbursement success –> the money is back in Mifos. The fund is back at source.

6. Fineract reverse disbursement failed –> The fund has not been reflected back in Mifos yet. Resort to manual reverse at this point?? This can potentially create the books to be unbalanced – funds sitting in limbo (neither at source or destination).

7. Fineract reverse no action taken as it’s already reversed à no need to reverse because it was somehow already reversed (could be someone did it in Mifos manually)

8. Manually reversed à Resort to manual reversing (only for those that keep failing through automated process – status #5 above). They should be able to reverse it manually in Mifos and the status gets sent to Mobile Money Gateway to ‘Manually reversed’.

The goal is to be able to see where the funds are at any moment. How much is reflected back in Mifos and what’s pending to be reflected (2nd status above).

It’s our understanding that MFIs close their balances each day and reconcile them actively. The next day, they should be able to see the details of why the figures changed (from reversals).

If they can run a report by status, by date, etc., they can expect how much is in pending mode – either way (inbound/outbound).

Option 2: Via Scheduler (not recommended)

* At the end of the day or periodically, a scheduler job will pick up all the new disbursements, chunk it and make the api calls to Mobile Money Gateway which will further process the requests.
* If this method is chosen, there is a risk of schedulers being invoked at the same time from each app server (in case of load balancers – multiple app servers). The schedule is saved in one db but app servers are using that same time to trigger these.
* It will require additional logic of putting through queue and manage the transactions there to ensure we don’t process the same transaction more than once.

Option 3: Webhook (not recommended)

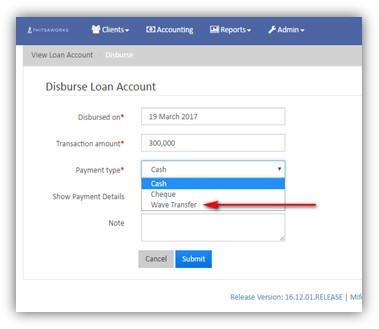
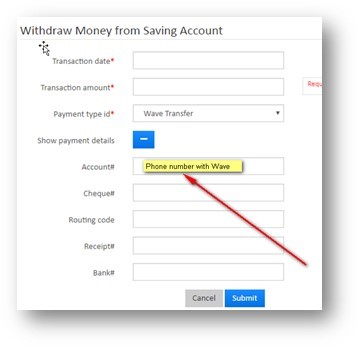
-- similar to #1 real time approach but it has limitations and not recommended.

**Option 4: Excel batch mode (temp solution but there is a chance this solution will stay for some MMPs with no API capabilities)**

* This approach will require document management type solution in Mobile Money Portal.
* We didn’t discuss this option in great length but at the end of this email, please see our initial stab at this workflow.

**Multiple numbers per client**

1. We should have a new table on the client level to store the mobile numbers, MMP and payment method. Payment method could be mobile money or other types (for future integration).
2. During recording of these transactions, the numbers should automatically pop up in the payment details – account section. (This will reduce typing errors)



**Funds initiated by Clients (\*\* This workflow has a revised version later in the meeting notes)**

1. Customers use a smart phone and open the MMP app on their phones.

2. MMP api will call Mobile Money API with source reference, destination reference.

3. Mobile Money Gateway will log the event including date/time, IP that’s requesting.

4. Mobile Money Gateway will query Mifos API and return the account numbers, amount due, due date. (after verifying that the client belongs to this MMP via the new table discussed above).

5. They will see a list of accounts with amount due/date.

Three lines if they have three accounts – loans, mandatory savings, voluntary savings

• For loan and mandatory savings, they will click on each line to make exact payment amount in the smart phone app. They will make two separate transactions (one for loan and one for savings).

• For voluntary savings, they get to enter an amount and hit send. (There is no fixed amount for voluntary savings as they can put in whatever they are comfortable with)

6. MMP api will check whether the customers have the balance in their MMP account and if yes, they will send the amount and account to Mobile Money Gateway.

7. Mobile Money Gateway will log the event including date/time, IP that’s requesting.

8. Mobile Money Gateway will post the transaction in Mifos and return the transaction status to MMP and save the status.

9. If the status is failed, MMP api will put the money back to client’s account.

Both MFI and MMP users can log in to Mobile Money Portal and view the transaction logs with status and reason.

### Wednesday, 17 May, 2017 11:34 follow-up notes

Thanks for your time on the call today. I'll send a follow-up to the public lists too but I wanted to shared with you a link to the demo server of Musoni System so you could see how they handle mobile money configuration. Most of their integrations are still not in real-time via APIs but this will help to understand a bit more:

<https://demo.musonisystem.com/demo/index.php>

user: demo

pass: demo123

Simply navigate to Configuration >> MMT Services and then you can see the configuration settings and status of transfers for various mobile money providers.

### Wednesday, 17 May, 2017 11:01 Ed’s notes

**Information Needed**

* Need partners and users in the community to share the high-level transaction flows that need to be supported
  + What are the inbound transactions (client to MFI)?
  + What are the outbound transactions (MFI to client)?
  + Explain from the perspective of the user (client or MFI staff) how a transaction is initiated and effected via whatever device (USSD interface, smartphone interface, web interface (staff))
  + Identify what information is or needs to be passed on during the transaction
* Understanding this will help us know what information must flow from/to Mifos/Fineract and the mobile money system via the mobile money gateway
* Partners need to provide access to the mobile money APIs for their country/region.

* + Does mobile money provider allow additional information to be passed on, if so what and how?
* What happens if a client has multiple mobile money accounts (i.e. one with MTN and o

**Outstanding Questions**

* What happens when clients have multiple financial accounts with the MFI (i.e. loan account(s), savings accounts)?
* If using the client mobile # as unique identifier, how do you specify which account transfer should go to?ne with Orange Money)
  + How do you distinguish their different mobile #s in Mifos/Fineract?
* Do clients pay for a group loan via a single mobile money account?
  + How do you handle allocation of such payments?

**Phasing**

* Phase 0
* GSOC 2016 (complete)
  + - Initial development of gateway/mobile money integration module & stand-alone UI with focus on MTN Cameroon
* Phase 1 - GSOC 2017
  + Complete Mobile Money Gateway
    - Make generic to support integration with mobile money APIs from different providers
      * Currently testing with Beyonic (multiple countries), Myanmar, Cameroon, Kenya
    - Support the transaction flows outlined above.
  + UI
    - Staff - Users of Mifos/Fineract to initiate disbursements should be able to do that from the current Web UI - no separate UI should be needed
    - Clients - clients should be able to initiate mobile money transactions from the mobile money providers’ UI - USSD or Smartphone
    - UI for checking status of payments, etc.
* Phase 2 - GSOC 2017
  + UI - Clients should be able to initiate mobile money transactions via a Mifos self-service Android app
    - Could extend off of the UI that Daniel built in 2016.

### Wednesday, 17 May, 2017 11:01 notes

**High level info flow**

Funds from customer

1. Loans - Undo Disbursals -- can be triggered by MFI from Mifos (Need to handle failed transactions) or Mobile Money GUI
2. Loans – Make Repayments – from be triggered from MFI client via money provider app or self-service app
3. Loans – Prepay Loans – from be triggered from MFI client via money provider app or self-service app
4. Savings – Deposits – from be triggered from MFI client via money provider app or self-service app
5. Loans – charges, etc. – from be triggered from MFI client via money provider app or self-service app

For #2-5, MMP will initiate a request to the gateway with MFI client phone number or specific account number (loan 1#, loan 2#, savings #,etc).

A list of active account numbers and amount owe will be returned to MMP and MFI client has an option to either choose the exact amount or enter an amount.

Funds to customer

1. Loan disbursement – triggered by MFI from Mifos (Need to handle failed transactions) or Mobile Money GUI.
2. Withdrawal from Savings – triggered by MFI from Mifos (need to handle failed transactions) or Mobile Money GUI.

As for database design (mobile money), we’d like to request a few more tables.

A table for saving transaction-status.

-Tenant-transactions-id

-date/time of transaction (when it hit mobile money gateway)

-status

-API return code (from either side provider or Fineract)

-Server ip of the request

-Comments or error details/codes

-External id (transaction id of external system)

In tenant-transactions table, we should save all the information that’d be transmitted either direction.

1. The json or messages itself **or**
2. critical information in the message such as client account number with provider (at the time of the transaction), what amount was returned/received via the API, etc.
3. \*we’d need all the details to troubleshoot and reconcile on each end.

In tenants table, should we add tenant identifier, MFI api url, etc to let this platform know which tenant/database/server.

For item #2-5 above (where 3rd party retrieving info), we should check whether the MFI client is associated with this MMP. Otherwise, we should NOT return any result for security reasons. Only allow legitimate requests for lookup info. For this, we will need to store a relationship between MMP and MFI Client and a workflow for maintaining this info.

Feature request

1. Mobile money gateway UI to display transactions for reconciling (reports, etc)
2. Mobile money gateway UI to show error report for monitoring purposes
3. In Mifos, allow saving of different phone number for different MMP payment type

