Settlement administrator stories

The following stories define the use cases that a hub administrator will require to be implemented so that they can control and use the settlements functionality currently under development.

# A note on status

The new settlement functionality will assign transfers to settlement batches based on a deterministic rule specified as part of the settlement model to which the transfer belongs. Each batch will have a status associated with it, and an administrator should be able to change the status of one or more settlement batches, subject to rules which are described in Section 0 below. The status of a settlement batch may be set to one of the following values:

* OPEN: transfers may be added to the settlement batch. The batch may be settled.
* CLOSED: transfers may not be added to the settlement batch. The batch may be settled.
* IN\_DISPUTE: transfers may not be added to the settlement batch. The batch may not be settled.
* AWAITING\_SETTLEMENT: transfers may not be added to the settlement batch. The batch may be settled, but only via confirmation of the settlement request which changed its status to AWAITING\_SETTLEMENT.
* SETTLED: transfers may not be added to the settlement batch. The batch may not be settled.

The thinking around the AWAITING\_SETTLEMENT status is as follows. When an administrator requests a settlement (see Section 2.4 below), they are requesting that the content of that settlement should be fixed and that the batches selected for settlement as part of that request should be excluded from settlement via any other route: the settlement request in effect reserves the batches for settlement. The requirements do not prevent the same settlement batch being selected for more than one report; and therefore the underlying functionality of settlement definition needs to be able to distinguish the context in which a batch is marked for settlement.

An example of this is: let’s assume that we have a settlement report which covers all transfers from 12:00 to 18:00 on 1st March 2023, and a second settlement report which covers all transfers from 15:00 to 16:00 on the same date. If an administrator decides to settle the transfers defined in the second of those reports, then all of the batches which contain transfers between 15:00 and 16:00 should have their status set to AWAITING\_SETTLEMENT. When an administrator views the first report (as described in Section 2.3 below) they should see that the batches which cover the period between 15:00 and 16:00 have been marked for settlement; and they should not appear as part of the first report’s settleable total, because they were reserved by another settlement execution request. From here, the following outcomes are possible:

* The settlement request for the second report is confirmed. Now the batches from 15:00 to 16:00 are marked as SETTLED in the first report.
* The settlement request for the second report is cancelled. Now the batches from 15:00 to 16:00 are marked as CLOSED in the first and second reports, and may be settled via either report.

If an administrator requests settlement of the first report before either of the outcomes described above, then the settlement batches covering the periods 12:00 to 15:00 and 16:00 to 18:00 should be marked as AWAITING\_SETTLEMENT. If an administrator subsequently cancels the second settlement request, then the batches from 15:00 to 16:00 will be marked as CLOSED. If an administrator now wants to include them in the settlement associated with the first report, then they should cancel the settlement request for the first report and re-apply it. Now the CLOSED batches from the second report can be included in the settlement request for the first report.

# Stories

## View settlement models

As a hub administrator, I want to be able to see the settlement models currently defined for my scheme, so that I can specify settlement reports using the characteristics of each settlement model.

## Create a settlement report

As a hub administrator, I want to be able to specify a group of transfers by their characteristics and view their contents, so that I can understand the state of settlement obligations in the scheme.

The action of creating a settlement report should return a statement of the current report, as defined in Section 2.3 below.

### Additional requirements

The characteristics of the transfers to be included should be specifiable along the following axes:

1. A status value for the batches to which the transfers belong. For instance, it should be possible to request a report which contains all of the batches whose status is IN\_DISPUTE.
2. A settlement model to which the transfers belong
3. The time at which the transfer completed. I want to be able to specify times down to the maximum granularity defined in the settlement model’s batch definition, but not below that. Specifying a time granularity which is below that supported by the settlement model should result in an error.
   1. Potential segmentation levels are:
      1. Year
      2. Month
      3. Day of month
      4. Hour
      5. Minute
      6. Second
   2. At any level of time segmentation, I want to be able to specify a start and end point.
   3. If a start and end point is specified at a particular level (for instance, day of month,) then all higher levels (for instance, year and month) should have a single specification (for instance, February 2023).
4. Any other characteristics which form part of the settlement model selected. Examples might be currency channel or sub-scenario (where, for instance, remittance transfers are identified by their sub-scenario.)

## View a settlement report

As a hub administrator, I want to see the current content of a previously created settlement report, so that I can understand the obligations of the participants in my scheme.

I want to specify the report using *either* the identifier that was returned when the report was created in Section 2.2 above, or the identifier that was returned from a settlement execution request (see Section 2.4 below.)

A settlement report should contain:

* The unique identifier of the settlement report, if there is one.
* The settlement identifier associated with the settlement report (see Section 2.4 below), if there is one.
* The criteria which were used to specify the inclusion of settlement batches in the settlement report.
* A record for each settlement batch, containing:
  + The name of the batch
  + The current status of the batch
  + For each participant in the scheme:
    - Their net position in the batch.
* For each batch status except AWAITING\_SETTLEMENT, for each participant in the scheme:
  + The total position for all batches in the report which have that status.
* Batches with the AWAITING\_SETTLEMENT status should be divided into multiple rows, one for each settlement execution request to which they belong. The first row should show the batches awaiting settlement because they belong to the settlement execution request for this report. Subsequent rows should show batches awaiting settlement as a consequence of other settlement execution requests.
* For each participant in the scheme:
  + The total settleable position for the report. The settleable position is the sum of all batches whose status is one of:
    - OPEN
    - CLOSED
    - AWAITING\_SETTLEMENT, where this status was acquired by the settlement execution request whose identifier is associated with this settlement report.

## Request execution of a settlement

As a hub administrator, I want to be able to request that a settlement should be executed based on the content of a settlement report. It should be possible *either* to execute a settlement on a previously defined report *or* to specify a set of criteria (in the same form as defined in Section 2.2 above) for the settlement to be executed.

A settlement request should result in the following actions:

* All settlement batches covered by the criteria for settlement which have a status of OPEN or CLOSED should have their status changed to AWAITING\_SETTLEMENT.
* The proposed settlement should be assigned a unique identifier.
* A settlement report (as defined in Section 2.3 above) showing the amounts to be settled.

## Cancel execution of a settlement

As a hub administrator, I want to be able to cancel a previously requested settlement. The settlement to be cancelled should be identified by its unique identifier (see Section 2.4 above).

The consequence of this request should be that all the batches whose status was marked as AWAITING\_SETTLEMENT by the settlement request should have that status changed to CLOSED.

## Confirm execution of a settlement

As a hub administrator, I want to be able to confirm that a settlement which has previously been requested according to the use case described in Section 2.4 above, so that I can implement maker/checker functionality on settlement execution if required, or so that I can review the contents of the settlement and confirm action.

The consequence of this request should be that all the batches whose status was marked as AWAITING\_SETTLEMENT by the settlement request should have that status changed to SETTLED.

## Change the status of a settlement batch

As a hub administrator, I want to be able to change the status of one or more settlement batches so that I can manage a set of transfers.

The following ways of identifying the settlement batches whose status is to be changed should be supported:

* The name of the settlement batch
* All settlement batches in a pre-defined report to which the proposed change can be applied, specified by the identifier of the report.
* All settlement batches in a pre-defined settlement request to which the proposed change can be applied, specified by the identifier of the settlement request.

The following status changes should be allowed:

* Change OPEN to CLOSED
* Change OPEN to IN\_DISPUTE
* Change CLOSED to IN\_DISPUTE
* Change IN\_DISPUTE to CLOSED

## Identify the settlement batch to which a transfer belongs

As a hub administrator, I want to be able to identify the settlement batch to which a transfer, specified by its transfer ID, belongs, so that I can take action on the settlement batch in case of dispute.

## Unreported batches

As a hub administrator, I want to be able to see any settlement batches which have not been included in at least one settlement report, so that I can ensure that there are no settlement batches which have escaped the settlement process.

I should be able to select one or more batch status values to constrain the settlement batches selected for inclusion in the report.

For each entry, I would like to see the settlement batch name and its current status..

## Unsettled settlement reports

As a hub administrator, I want to be able to identify any settlement reports which have either not been settled, or have only been partially settled, so that I can take action where appropriate to ensure that all transfers are settled.

I should be able to specify a time period for the report. Selecting a time period means selecting all the settlement reports which include settlement batches whose time stamp falls within the time range selected.

# Further review comments

## 202s and PUTs

The **POST /matrices/{id}/recalculate** endpoint returns a 202. In the asynchronous Mojaloop syntax, this would lead one to expect that a corresponding **PUT** would be created by the switch. Ordinarily, the admin interface endpoints are synchronous – though this needn’t be the case if asynchronous is more appropriate. But in this case there seems to be neither a matching **PUT** nor a response body. Does any information get returned to the caller?

## Using objects for repeated enumerations

There are cases where the same enumeration is defined twice: for instance, **batchStatuses** in **GET /batches** and **state** in **SettlementMatrixBatch**. Better practice is to replace the two definitions with references to the same object, to ensure that changes to the enumeration are immediately reflected across all instances of it.

## Notification of stale batches

I think I recall from the console demonstrations we’ve seen that updates happen to the client, such as saying when recalculation is complete and when a matrix is out of date. Is that correct? If so, how does that process work?

## SettlementConfig

The definition of a settlement describes the length of a batch in seconds (*batchCreateInterval*). But defining a batch timing as (for instance) 1 minute is semantically different from defining it as 60 seconds, since the former means *measured from when the minute changes*. How does the current implementation set the start period for a batch?

## matrices/{id}/recalculate

I think I’ve said this before. In my opinion, the matrix should be recalculated (for those elements that require recalculation) as part of a **GET**. Asking people to manually request a recalculation (particularly if this is as a consequence of getting a response to a **GET /matrices/{id}** where the state is **OUT\_OF\_SYNC**) irritates them and, as far as I can see, offers no gain in efficiency. Can we talk about this?