mojaloop

Settlements: how we're doing, where we're going

Michael Richards
Rapporteur, ModusBox

What is a rapporteur?

colporteur

noun

A peddler of devotional literature.

Origin of colporteur

French *alteration* (*influenced by col neck, from the idea that peddlers carry their wares on trays suspended from straps around their necks*) *of* Old French *comporteur from comporter to conduct, peddle* ; see **comport** .

What is a rapporteur?

colporteur

noun

A peddler of devotional literature.

Origin of colporteur

French *alteration* (*influenced by col neck, from the idea that peddlers carry their wares on trays suspended from straps around their necks*) *of* Old French *comporteur from comporter to conduct, peddle* ; see **comport** .

What is a rapporteur?

colporteur

noun

A peddler of devotional literature.

Origin of colporteur

French *alteration (influenced by col neck, from the idea that*

Hence the well-known phrase:

“Rapporteur ain’t nuthing ta ** wit”**

Settlements: plan from the last convening

- Support existing settlement requirements
  - Deferred net settlement by currency (Mowali)
  - Continuous gross settlement (TIPS)

- Additional requirements:
  - Provide upgrade path for existing implementations

How did we do?

- Deferred net settlement
  - Can be specified by currency
  - Administration API endpoints have been modified to support multi-currency
- Continuous gross settlement
  - Design is complete
  - Coding is under way

Settlement windows: how do they work?

- Every transfer is assigned to a single live settlement window.
- No matter what:
  - Settlement model it belongs to;
  - Currency it is denominated in
- Why a single window?

Settlement windows: how do they work?

- Every transfer is assigned to a single live settlement window.
- No matter what:
  - Settlement model it belongs to;
  - Currency it is denominated in
- Why a single window?
  - To ensure that every transfer always belongs to some settlement window
  - To ensure that all the ledger entries for a transfer belong to the same settlement window
  - To simplify the assignment process

Settlement windows: how do they work?

- Every transfer is assigned to a single live settlement window.
- No matter what:
  - Settlement model it belongs to;
  - Currency it is denominated in
- Why a single window?
  - To ensure that every transfer always belongs to some settlement window
  - To ensure that all the ledger entries for a transfer belong to the same settlement window
  - To simplify the assignment process
- You can then assign multiple settlement windows to a settlement

Net settlement

- Transfers are aggregated together and the net of all the transfer is settled
- Which is either:
  - A single bill for each participants, covering all of their transactions; or
  - A bill for each pair pf participants, covering their transactions with each other

Gross settlement

- Every transfer is individually settled when it is completed
- So there's either:
  - No settlement process (in the settlement window sense); or
  - Lots of settlement processes

Settlement status

- Every element of a settlement has a status

Settlement status

- Every element of a settlement has a status
- What's an element? Depends on the settlement model:
  - For net settlement, it's the net amount that a participant owes or is owed
  - For gross settlement, it's the individual transfer

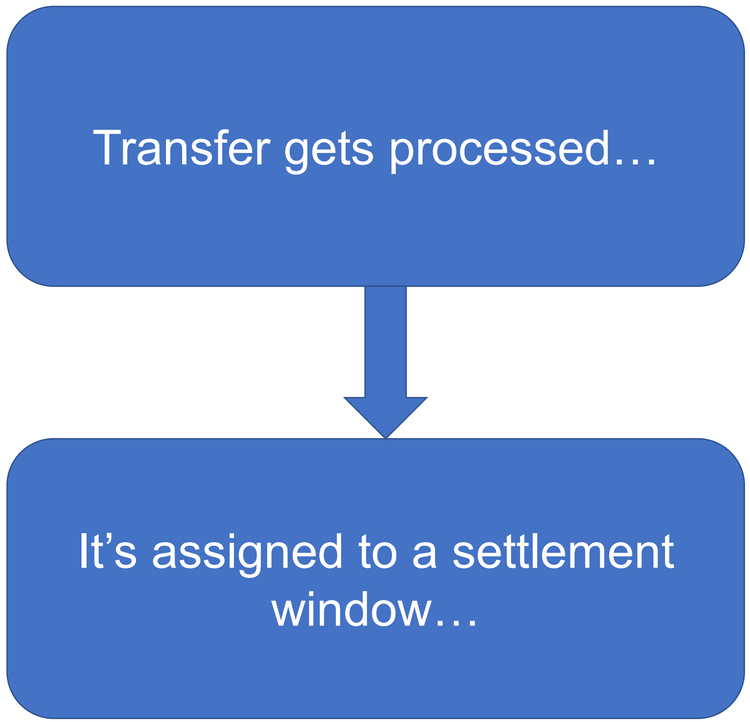
Settlement status

- Every element of a settlement has a status
- What's an element? Depends on the settlement model
  - For net settlement, it's the net amount that a participant owes or is owed
  - For gross settlement, it's the individual transfer
- What's a status? For instance:
  - Ready for settling
  - In process of settling
  - Settled
  - Ooops, aborted

So now we have our concepts:

- Settlement windows
- Settlement types
- Settlement status

How things used to be:



How things used to be:

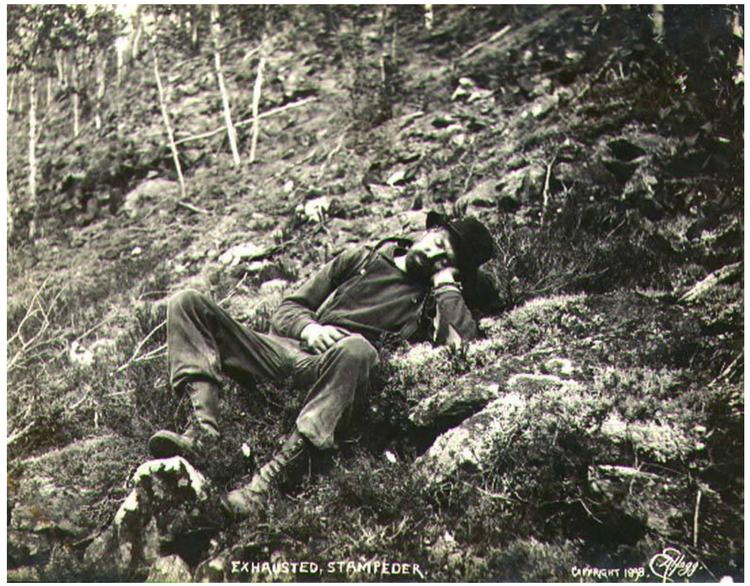
Transfer gets processed…

It’s assigned to a settlement window…

How things used to be:

Transfer gets processed…

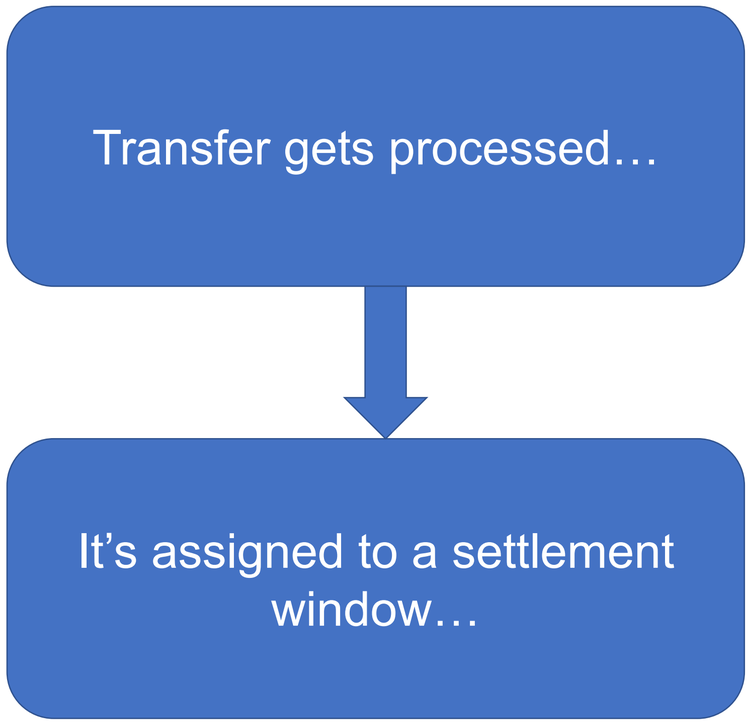
It’s assigned to a settlement window…



Settlement is requested…

Aggregations are produced and state is recorded at the aggregation level.

How things are going to be:



How things are going to be:

Transfer gets processed…

It’s assigned to a settlement window…

We’re settling gross!
This transfer gets settled individually!!

How things are going to be:

Transfer gets processed…

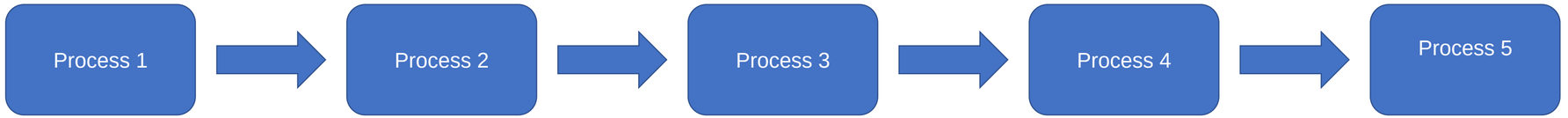
It’s assigned to a settlement window…

We’re settling gross!
This transfer gets settled individually!!

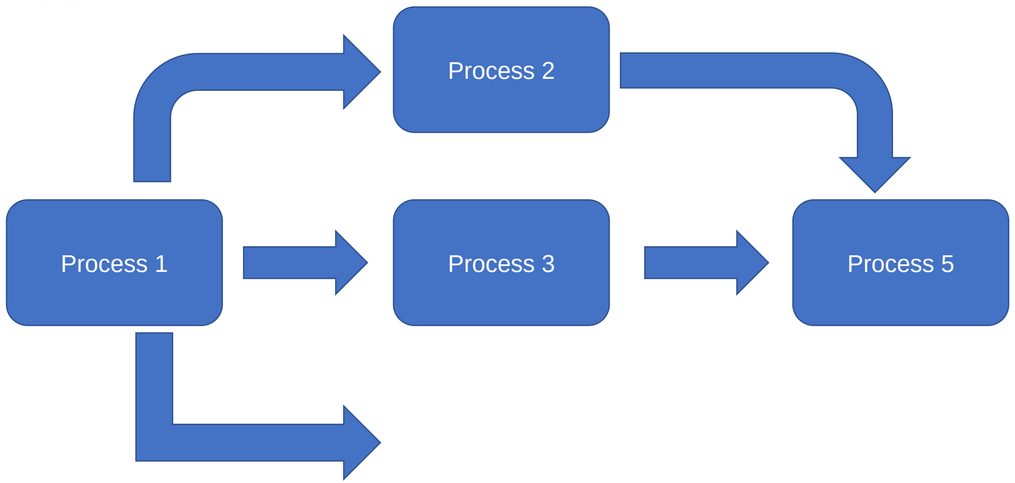
We need to record status against each ledger entry in each transfer…

Ensuring performance

- Replace synchronous operations with asynchronous operations (Performance 101)
- Before:

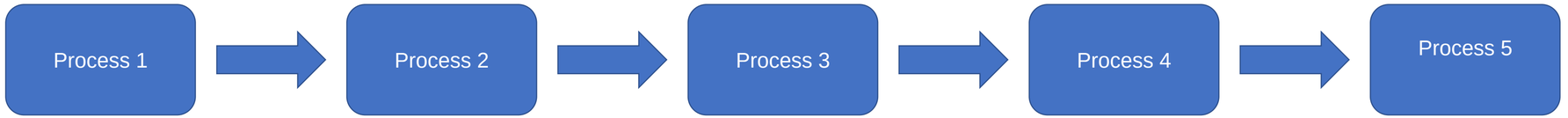


- After:

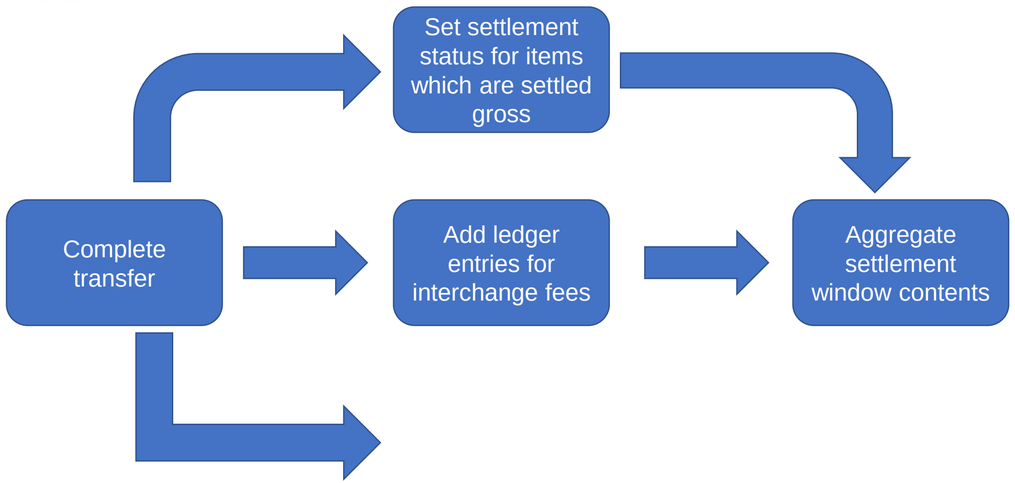


Ensuring performance

- Replace synchronous operations with asynchronous operations (Performance 101)
- Before:

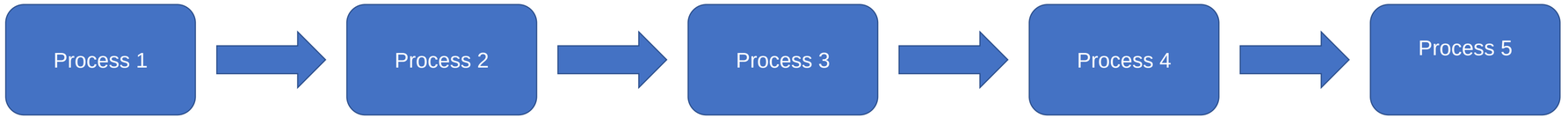


- After:

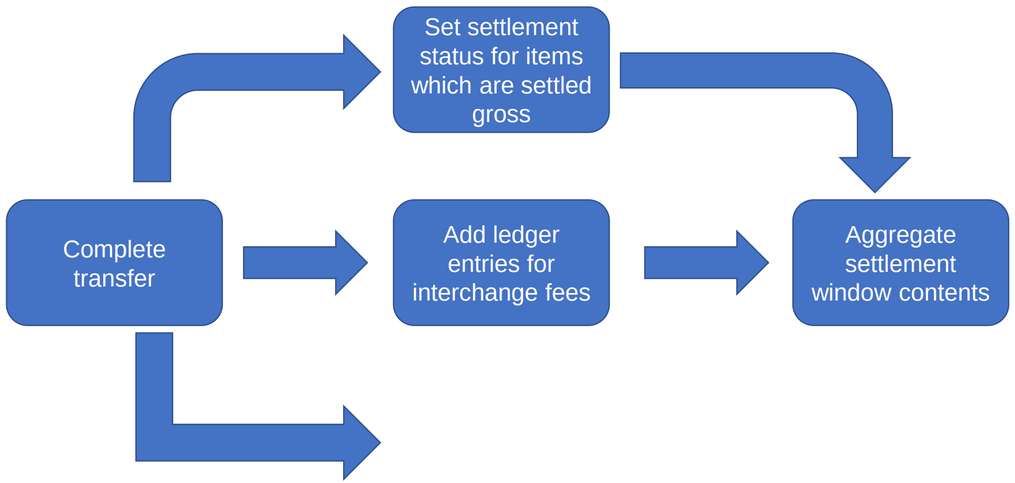


Ensuring performance

- Replace synchronous operations with asynchronous operations (Performance 101)
- Before:



- After:



Which only leaves the question:
How does aggregation know when everything's ready?

Ensuring performance

- Replace synchronous operations with asynchronous operations (Performance 101)
- This replaces the immediate problem with a synchronisation problem.

Synchronising operations for gross settlement

- What do we need to synchronise?
  - For any transfers which are settled gross, they need to have a status
  - For any transfers which have additional ledger entries added, the entries need to have been, er, added

Synchronising operations for gross settlement

- What do we need to synchronise?
  - For any transfers which are settled gross, they need to have a status
  - For any transfers which have additional ledger entries added, the entries need to have been, er, added
- How do we synchronise?
  - By looking for some evidence of work having been done
  - i.e. all ledger entries have status records associated with them

So what we do is:

1. Put the transfer on a stream when it completes
2. Have a process which adds the additional ledger entries for interchange fees where required reads from the stream
3. Have a process which sets the status for individual ledger entries reads from the stream
4. Settlement window aggregation can only start when there are no ledger entries which belong to the settlement window which do not have a status set

Settling transfers gross

The process which sets the status of a transfer which is settled gross can:

- Create status records for all the appropriate status values
- Log that it performed the settlement operation for that transfer
- Optionally send settlement information to an external listener

So what remains?

- The aggregation level question

So what remains?

- The aggregation level question
- Er, which is?

Proposed solution:

- We use a rule to assign a category to a transfer

Proposed solution:

- We use a rule to assign a category to a transfer
- We store the category against the transfer

Proposed solution:

- We use a rule to assign a category to a transfer
- We store the category against the transfer
- We aggregate by category, whatever that is

Any questions?