MODUSBOX

Settling liabilities in Mojaloop systems

Defining an API

Indispensible cultural motto

A spending hand that alway poureth out,
Had need to have a bringer-in as fast;
And on the stone that still doth turn about,
There groweth no moss: these proverbs yet do last;
Reason hath set them in so sure a place,
That length of years their force can never waste.

Sir Thomas Wyatt

Objectives

1. To discuss (and, hopefully, define) the things that we need to include.
2. To define the settlement models that we intend to support
3. To define the objects that we propose to work with
4. To define the actors we expect to want to use a Settlement API
5. To model the ways in which we expect each actor to use the API

Objectives

1. To discuss (and, hopefully, define) the things that we need to include.
2. To define the settlement models that we intend to support
3. To define the objects that we propose to work with
4. To define the actors we expect to use a Settlement API
5. To model the ways in which we expect each actor to use the API
6. To dip our toe into the dark waters of definition…

What sorts of things do we need to include?

Defining terms

Definition of terms: settlement accounts

- A *traditional settlement account* is external to the system. It holds actual funds for a DFSP in a bank account at the settlement bank. This account is debited or credited with settlement entries, made by the settlement bank under arrangement with the payment scheme.
  - DFSPs manage their accounts by transferring funds into or out of other accounts they hold. Balances in these accounts represent liability cover for the participant(s) in those accounts.
- A pooled settlement account is a settlement account which holds actual funds in a bank account at the settlement bank representing liability cover for more than one DFSP.
- Either traditional or pooled settlement accounts *may* be used with either net or gross settlement.

Definition of terms: gross and net, traditional and pooled

- A settlement account may also be a pooled account.
  - In a gross settlement model, there will normally only be a pooled account.
    - Since gross settlement settles after each transfer, the use of multiple settlement accounts requires the capacity to debit and credit these accounts at very high throughput.
  - In a net settlement model, there will normally only be settlement accounts.
    - It’s not clear how net settlement can be managed within a pooled account.
  - In a gross/net hybrid settlement model, there will be traditional settlement accounts and pooled accounts.
    - A gross/net settlement model could settle net between traditional settlement accounts and gross within pooled accounts.

Definition of terms: scheme accounts

- A scheme account is an object within the scheme. It represents a specific type of account in a given currency.
- A DFSP can only belong to one scheme account of a given type in a given currency.
- Only scheme accounts can participate in a net settlement.
- Settlements are carried out at the *scheme account* level, not the DFSP level.
- Settlements are carried out between scheme accounts of the same type

Definition of terms: settlement types

- Net settlements and gross settlements
  - In a *net* settlement model:
    - Multiple individual transfers are summed together *over a period*, and the net of those transfers is settled by participants.
  - In an *gross* settlement model:
    - Each transfer is settled *immediately* it is committed.
- Bilateral and multilateral settlements
  - In *bilateral* settlements, each participant settles *with the scheme* for the net of all the transfers in which it has participated.
  - In *multilateral* settlement models, each participant settles with every other participant for the net of the transfers between those two parties.

Settlement models in the market today

Net Settlement Systems

- Almost universally used for retail open-loop systems
- Can be either bilateral (each participant receives an overall statement) or multilateral (each participant settles with every other participant)

Gross Settlement Systems

- Universally used for wholesale open-loop systems (RTGS)
- Being used by some RTRP systems (e.g. Mexico)

Gross Settlement with Pooled Account

- Universally used for Mobile Money systems
- Being used in US banking RTP system.
- Under consideration for Tanzania TIPS.

Hybrid Gross/Net Systems

- Proposed in this presentation

The gross/net hybrid settlement model

- This is a hypothetical proposal, since no existing systems implement this model
- Are there circumstances that might require it?
  - If so, what are they?
  - If not, are there any implications of not supporting it in the API?
- How would it work?

Definition of terms: ledger groups and ledgers

- A *ledger group* defines a group of ledgers in a scheme, one per participant, which are used to store movements of one or more types.
  - Only one ledger group in a scheme may be used to store funds transfers.
  - Ledger groups belong to an account
  - In addition to storing actual transfer amounts, ledger groups can be used to implement scheme-specific charges such as interchange fees.
- A *ledger* stores a record of the liabilities and assets for a particular DFSP of the types of movement defined for a particular ledger group.
- A *position* is the net of liabilities and assets accumulated by a DFSP in a ledger.

Definition of terms: NDC

- A *Net Debit Cap (NDC)* for a participant represents the net of deposits and withdrawals made to or from a settlement account by a DFSP.
  - For pooled accounts, the NDC also includes the net of credits and debits in the pooled account for the DFSP
- An NDC represents the maximum negative position which a DFSP may accumulate. Transactions which would take the DFSP's position beyond the NDC are not permitted.
  - One or more ledger groups may be defined as contributing to the maximum negative position which is checked against a Net Debit Cap
- The scheme may vary an NDC *ad lib*, either for individual participants or for the scheme as a whole
  - For example, a scheme might reduce an NDC to encourage participants, or because it trusts them.
  - For example, a scheme might increase an NDC to provide cover for periods when a settlement account is unavailable.
- An NDC is associated with a ledger, because it is DFSP-specific; but not all ledgers need to have NDCs.
  - For instance: interchange fees may be recorded in ledgers and settled via scheme accounts without the scheme wanting to require them to be backed by real funds

Definition of terms: settlements

- A *transaction window* groups a number of transactions together across all ledgers in the scheme.
  - The transactions in a transaction window may be cleared for settlement or not.
- A *settlement* transfers funds between settlement accounts and resets the Net Debit Cap for each participant in the settlement
  - A settlement must be aligned on a transaction window boundary.
  - If a settlement does not align on a transaction window boundary, then the transaction window which crosses the settlement boundary should be closed at the point of the settlement boundary and a new transaction window created to hold the remainder of the original transaction window.

So, what objects are we proposing?

And, er, why?

Scheme accounts

- Define points between which (net) settlements occur

Why?

The settlement point is decoupled from the ledgers which record charges so that we can implement settlements which include more than one type of charge.

Ledger groups

- Implement different ledger types (= different types of charge) for individual participants
- Allow the rules for the calculation of charges to be defined by schemes and modified as required

Why?

We want to be able to define multiple types of charge (for instance, processing fees.)

It should be possible for scheme administrators to define these charge types and the rules by which they are calculated

Ledgers

- Define different ledger types (= different types of charge) to allow flexible definition of charge types

Why?

The scheme needs to be able to store the credits and debits for each type of charge independently, so that it can:

- implement different settlement patterns and liquidity coverage strategies for different types of charge;
- modify those settlement patterns and liquidity coverage strategies as may be required.

Net Debit Caps

- Define the external liquidity provision required to cover one or more types of charge (=ledgers)

Why?

- The scheme needs to be able to vary the amount of liquidity provision required for different types of charge.
- The scheme needs to be able to encourage participation and manage risk by varying the amount of liquidity cover required for individual participants in the scheme.
- The scheme should be able to change both of these policies *ad hoc*.

Hybrid gross/net settlement models

What are they and do we need to worry about them?

How does a hybrid gross/net model work in practice? An example

- DFSP A1 and A2 share a pooled account: account A
- DFSP B has its own settlement account, account B
- Each DFSP funds its settlement account with $1000
- First, we do an intra-account transfer:

	Transfers			Deposits			Account Balances		Account Positions		DFSP Positions			DFSP share of pooled account A	
Operation	**DFSP A1**	**DFSP A2**	**DFSP B**	**DFSP A1**	**DFSP A2**	**DFSP B**	**Account A**	**Account B**	**Account A**	**Account B**	**DFSP A1**	**DFSP A2**	**DFSP B**	**DFSP A1**	**DFSP A2**
DFSP A1 funds account A with $1000				1000			1000	0	0	0	0	0	0	100%	0%
DFSP A2 funds Account A with $1000					1000		2000	0	0	0	0	0	0	50%	50%
DFSP B funds account B with $1000						1000	2000	1000	0	0	0	0	0	50%	50%
DFSP A1 sends $100 to DFSP A2	**-100**	**100**					**2000**	**1000**	**0**	**0**	**-100**	**100**	**0**	**45%**	**55%**

How does a hybrid gross/net model work in practice? An example

- DFSP A1 and A2 share a pooled account: account A
- DFSP B has its own settlement account, account B
- Each DFSP funds its settlement account with $1000
- First, we do an intra-account transfer.
- **Next, we do two transfers between accounts:**

This is equivalent to DFSP A2 withdrawing $75 from the pooled account

This is equivalent to DFSP A1 depositing $150 into the pooled account

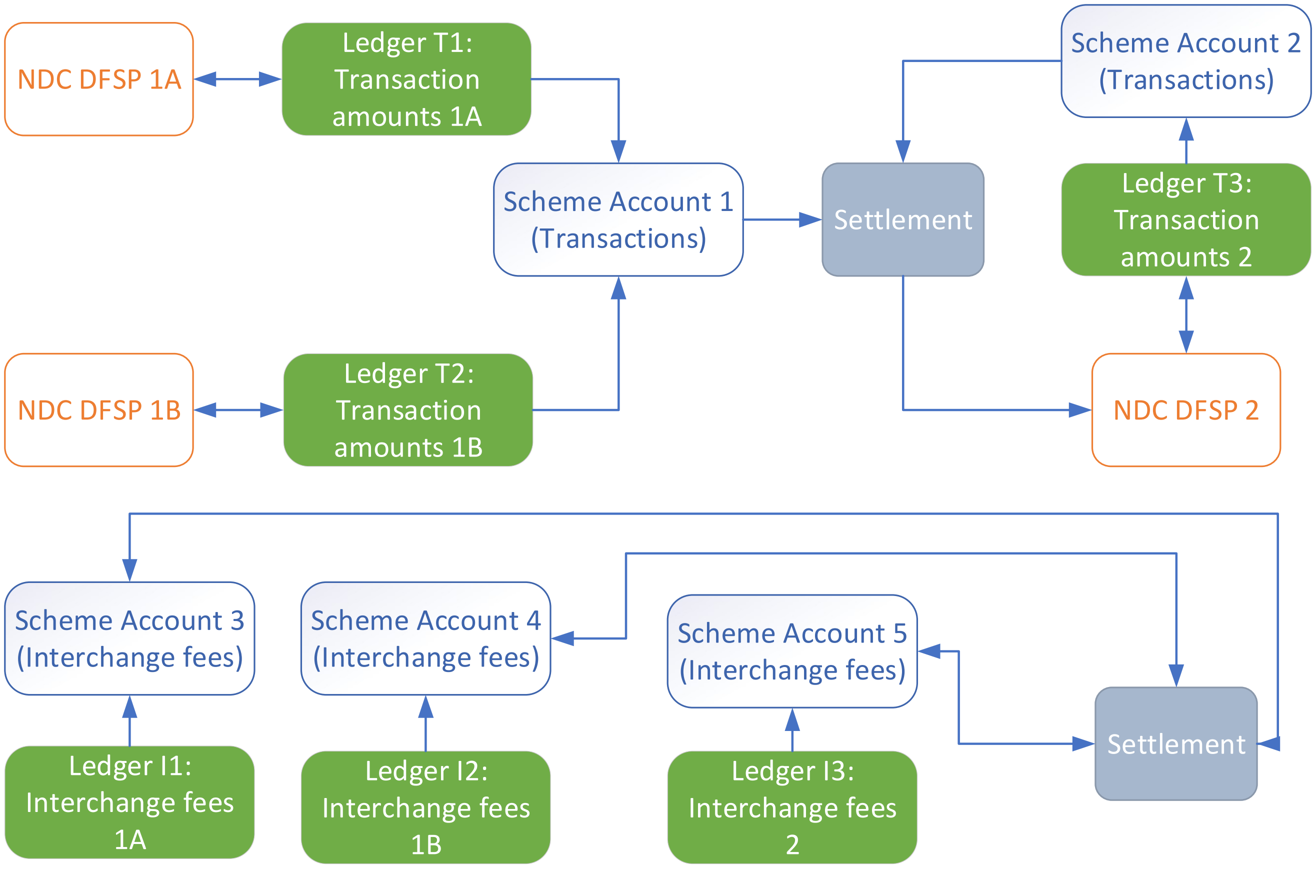
	Transfers			Deposits			Account Balances		Account Positions		DFSP Positions			DFSP share of pooled account A	
Operation	**DFSP A1**	**DFSP A2**	**DFSP B**	**DFSP A1**	**DFSP A2**	**DFSP B**	**Account A**	**Account B**	**Account A**	**Account B**	**DFSP A1**	**DFSP A2**	**DFSP B**	**DFSP A1**	**DFSP A2**
DFSP A1 funds account A with $1000				1000			1000	0	0	0	0	0	0	100%	0%
DFSP A2 funds Account A with $1000					1000		2000	0	0	0	0	0	0	50%	50%
DFSP B funds account B with $1000						1000	2000	1000	0	0	0	0	0	50%	50%
DFSP A1 sends $100 to DFSP A2	-100	100					2000	1000	0	0	-100	100	0	45%	55%
DFSP A2 sends $75 to DFSP B		**-75**	**75**				**2000**	**1000**	**-75**	**75**	**-100**	**25**	**75**	**47%**	**53%**
DFSP B sends $150 to DFSP A1	**150**		**-150**				**2000**	**1000**	**75**	**-75**	**50**	**25**	**-75**	**51%**	**49%**

How does a hybrid gross/net model work in practice? An example

- DFSP A1 and A2 share a pooled account: account A
- DFSP B has its own settlement account, account B
- Each DFSP funds its settlement account with $1000
- First, we do an intra-account transfer.
- Next, we do two transfers between accounts.
- **Now, we settle:**

	Transfers			Deposits			Account Balances		Account Positions		DFSP Positions			DFSP share of pooled account A	
Operation	**DFSP A1**	**DFSP A2**	**DFSP B**	**DFSP A1**	**DFSP A2**	**DFSP B**	**Account A**	**Account B**	**Account A**	**Account B**	**DFSP A1**	**DFSP A2**	**DFSP B**	**DFSP A1**	**DFSP A2**
DFSP A1 funds account A with $1000				1000			1000	0	0	0	0	0	0	100%	0%
DFSP A2 funds Account A with $1000					1000		2000	0	0	0	0	0	0	50%	50%
DFSP B funds account B with $1000						1000	2000	1000	0	0	0	0	0	50%	50%
DFSP A1 sends $100 to DFSP A2	-100	100					2000	1000	0	0	-100	100	0	45%	55%
DFSP A2 sends $75 to DFSP B		-75	75				2000	1000	-75	75	-100	25	75	47%	53%
DFSP B sends $150 to DFSP A1	150		-150				2000	1000	75	-75	50	25	-75	51%	49%
Settlement							**2075**	**925**	**0**	**0**	**50**	**25**	**-75**	**51%**	**49%**

Example: proposed settlement architecture


NDC DFSP 1A
Ledger T1: Transaction amounts 1A
Scheme Account 1 (Transactions)
Settlement
Scheme Account 2 (Transactions)
Ledger T3: Transaction amounts 2
NDC DFSP 1B
Ledger T2: Transaction amounts 1B
NDC DFSP 2
Scheme Account 3 (Interchange fees)
Scheme Account 4 (Interchange fees)
Scheme Account 5 (Interchange fees)
Settlement
Ledger I1: Interchange fees 1A
Ledger I2: Interchange fees 1B
Ledger I3: Interchange fees 2

What types of actors do we need to support?

… and how do they need to interact with the scheme?

Actors

- Scheme policy-makers – do not interact directly with the Settlement API
- Scheme administrators
- Scheme support staff
- Participant administrators
- The scheme itself

What do scheme policy-makers need to do?

- Decide on the settlement model
- Decide on settlement timetables for net settlements
- Decide on the types of charges to be supported and the rules for applying them
- Decide the NDC policy

What do scheme administrators need to do?

- Tell the scheme what types of account are required and, for each type, what settlement model is to be followed
- Tell the scheme what scheme accounts to set up
    - Tell the scheme what documentation to produce for each account for each net settlement and where to send it
- Tell the scheme what ledger groups to set up
- Tell the scheme what charges to apply in each ledger group
- For schemes which contain net settlement:
    - Tell the scheme what settlement timetable to apply
    - Request ad hoc settlements
- Tell the scheme what NDCs to set up and what margins to apply to them
    - Define default alert limits for participants
- Tell the scheme what deposits and withdrawals have been made to and from external accounts
- Approve participant requests to withdraw funds from settlement accounts *(outside API)*
- Review data relating to settlements

What do support staff need to do?

- Review data relating to settlements
- Identify groups of transactions where settlement needs to be delayed (e.g. because of questions about the legitimacy of a set of transactions.)
- Mark transaction groups as settleable or not settleable

What do participant administrators need to do?

- View historic, current and predicted liquidity requirements
- Define NDC alert levels
- Inform scheme administrators of deposits into settlement accounts *(outside API)*
- Request scheme administrators to withdraw funds from settlement accounts *(outside API)*

What does the scheme need to do?

(Deep breath…)

What does the scheme need to do?

- Schema account typesSettlement models:
  - Store the required account types and, for each type, the settlement model to be followed
- Net Debit Caps:
  - Create an NDC
  - Apply defined modifications to an NDC
  - Alter an NDC amount
  - Check that a proposed transfer does not violate an NDC and reject it if it does.
  - Alert participant administrators if their ledger position moves within a defined margin of their NDC
- Scheme accounts
  - Create a scheme account
  - Attach ledgers to a scheme account
- Ledger groups
  - Register a ledger group and the rules for creating entries in that group
  - Create a ledger for each participant in a ledger group
  - Create entries in ledgers:
    - Based on transfers
    - Based on the rules defined for each ledger group

What does the scheme need to do?

- Transaction groups:
  - Set up a timetable to generate events to create transaction groups
  - Create *ad hoc* transaction groups
  - Mark transaction groups as settleable or non-settleable based on user input
- Settlements (net (and hybrid) models only)
  - Set up a timetable to generate events to create settlements between accounts of a given type
  - Trigger *ad hoc* settlements
  - Process a settlement
    - Calculate the transaction windows to be included in the settlement
    - Calculate the number of settlement instructions to be generated
    - Generate the content of each settlement instruction
    - Send each settlement instruction to the URI specified
    - For each settlement instruction, adjust the appropriate NDC based on the settlement amount

The floor is yours…

Appendix

Settlement API

What areas does an API need to support?

- Settlement model definition
- Ledger group definitions
- Settlement accounts
- Transaction windows
- Settlement definition
- Settlement instructions
- Settlement reporting
- NDC changes

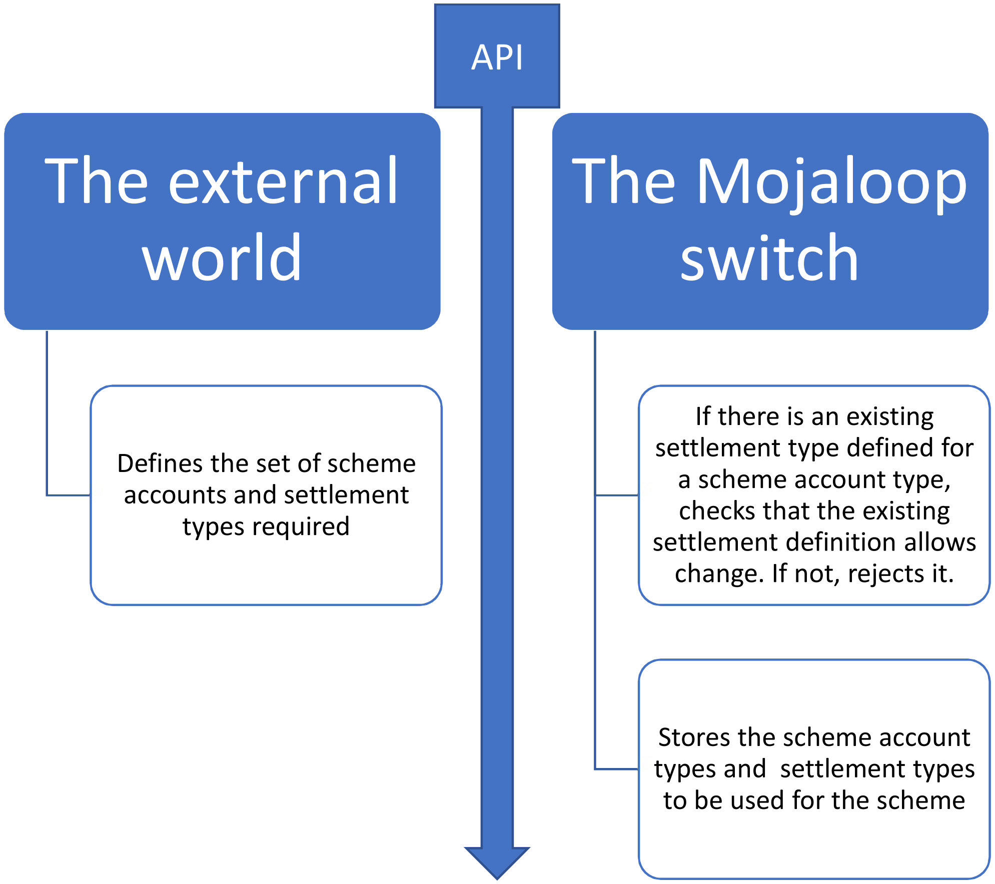
Settlement model: characteristics

- Scheme administrators can set up scheme account types.
- For each scheme account type, scheme administrators can select:
  - Net settlement via multiple accounts
  - Gross settlement via a single pooled account
  - Gross settlement via multiple accounts
  - … and (perhaps) hybrid settlement…
- If net settlement or gross settlement via multiple accounts is selected, the scheme should not allow pooled scheme accounts.
- If gross settlement via a single pooled account is selected, the scheme should not allow multiple scheme accounts

Settlement model: open questions

- Should the API support changing settlement models in flight?
  - Initial assumption: no
  - Which means: a call to this resource should fail with an error unless
    - No more than one scheme account is defined
    - If a scheme account is defined, no more than one DFSP is attached to that scheme account

Scheme account types: who does what?



API

The external world

Defines the set of scheme accounts and settlement types required

The Mojaloop switch

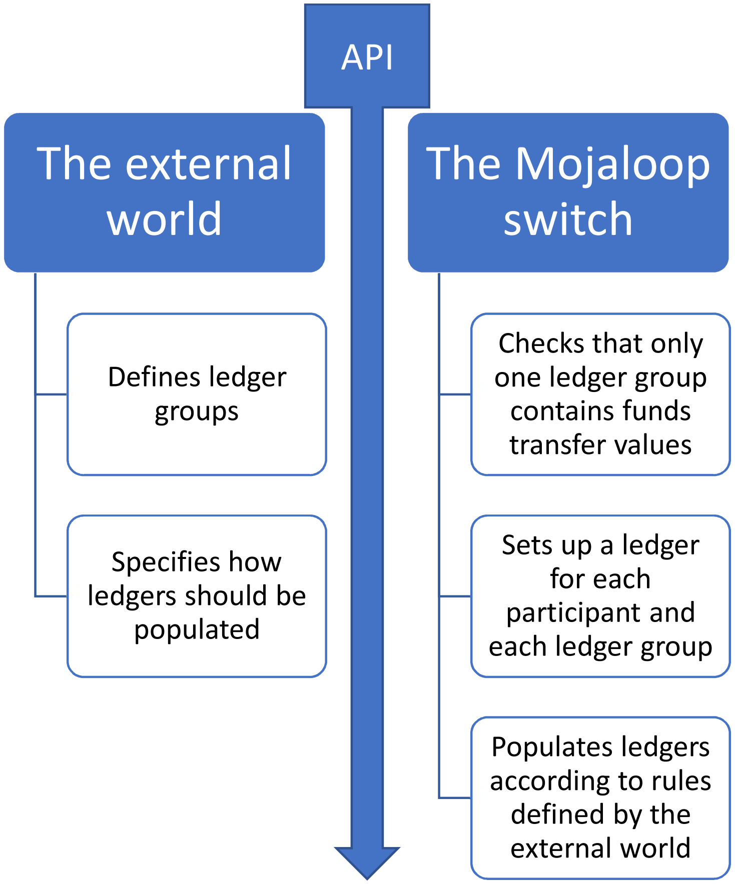
If there is an existing settlement type defined for a scheme account type, checks that the existing settlement definition allows change. If not, rejects it.

Stores the scheme account types and settlement types to be used for the scheme

Ledger groups: characteristics

- Scheme administrators can define ledger groups via the API
- A participant should have one ledger for each ledger group defined by the administrator.
- Once they have been created, ledger groups may not be deleted. They may be deactivated.
- For each ledger group, the administrator can specify whether or not its contents should be included in NDC calculations.
- A ledger of a given type can be populated *either* by funds transfer values *or* by values derived from funds transfer values by a deterministic process.
- Only one ledger group in a scheme can be populated directly by funds transfer values.
- ledger groups can be defined *either* for gross *or* for net settlements

Ledger groups: who does what?



API

The external world

Defines ledger groups

Specifies how ledgers should be populated

The Mojaloop switch

Checks that only one ledger group contains funds transfer values

Sets up a ledger for each participant and each ledger group

Populates ledgers according to rules defined by the external world

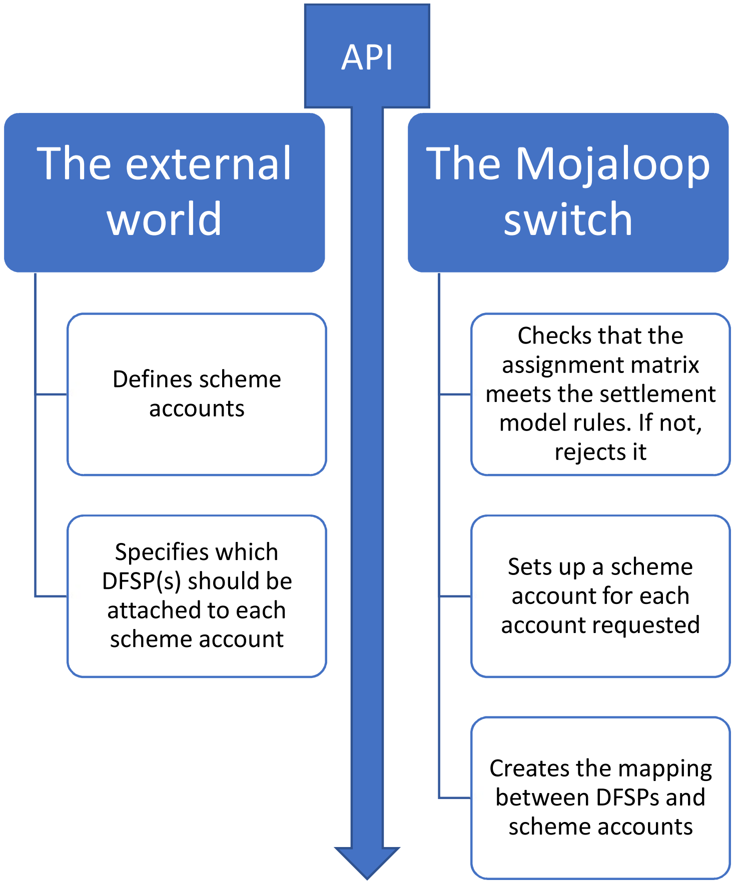
Scheme accounts: characteristics

- Scheme administrators can define scheme accounts via the API
- Each scheme account must belong to a defined scheme account type
- Scheme administrators can assign DFSPs to scheme accounts via the API.
- The scheme should apply rules to the assignment of DFSPs to scheme accounts as follows:
  - If net settlement or gross settlement via multiple accounts has been selected, then only one DFSP may be attached to each scheme account
  - If gross settlement via a single pooled account has been selected, then only one scheme account may be created.
- Scheme administrators can view the mapping between scheme accounts and DFSPs
- It should not be possible for a settlement (or a settlement timetable) to be specified if any DFSP in the scheme is not assigned to a scheme account.

Settlement accounts: open questions

- If a DFSP assigned to a settlement account is already assigned to another settlement account, should the assignment for that DFSP be changed by the switch, or should this be an error?

Scheme accounts: who does what?



API

The external world

- Defines scheme accounts
- Specifies which DFSP(s) should be attached to each scheme account

The Mojaloop switch

- Checks that the assignment matrix meets the settlement model rules. If not, rejects it
- Sets up a scheme account for each account requested
- Creates the mapping between DFSPs and scheme accounts

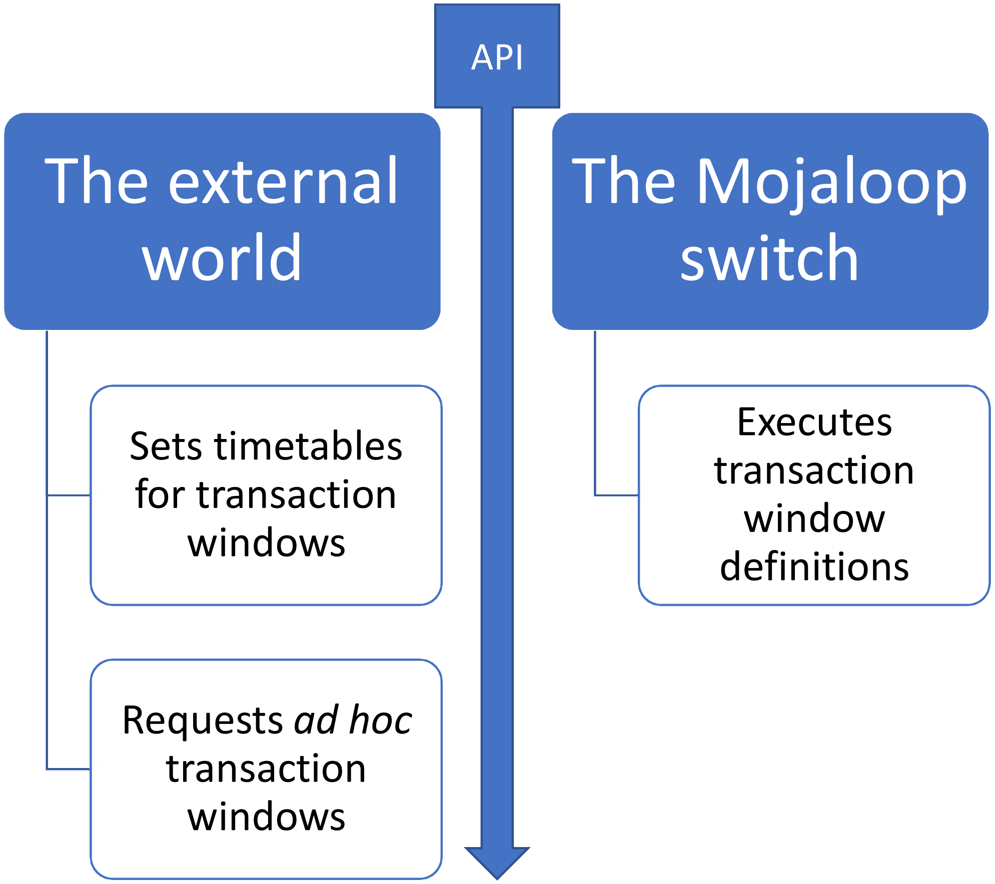
Transaction windows: characteristics

- Created by technical administrators
- Used to group transfers together
- Can be created on a timetable or *ad hoc*
- Only one transaction window can be active at a given time
- A transaction window applies to all ledgers in a scheme
- It should be possible to define a transaction window as settleable or not settleable: if a transaction window is not settleable, the transactions it contains should be excluded from any settlements that are made while it is in that state.

Transaction windows: open questions

- Should it be possible to assign a defined series of transactions to a transaction window (e.g. to isolate a group of transactions where fraud is suspected,) or should it only be possible to define a transaction window which ends *now*?
  - Working assumption: should be able to specify a start and end
- Should transaction windows be defined as settleable or not settleable by default?
  - Working assumption: Yes

Transaction windows: who does what?


API

The external world

Sets timetables for transaction windows

Requests *ad hoc* transaction windows

The Mojaloop switch

Executes transaction window definitions

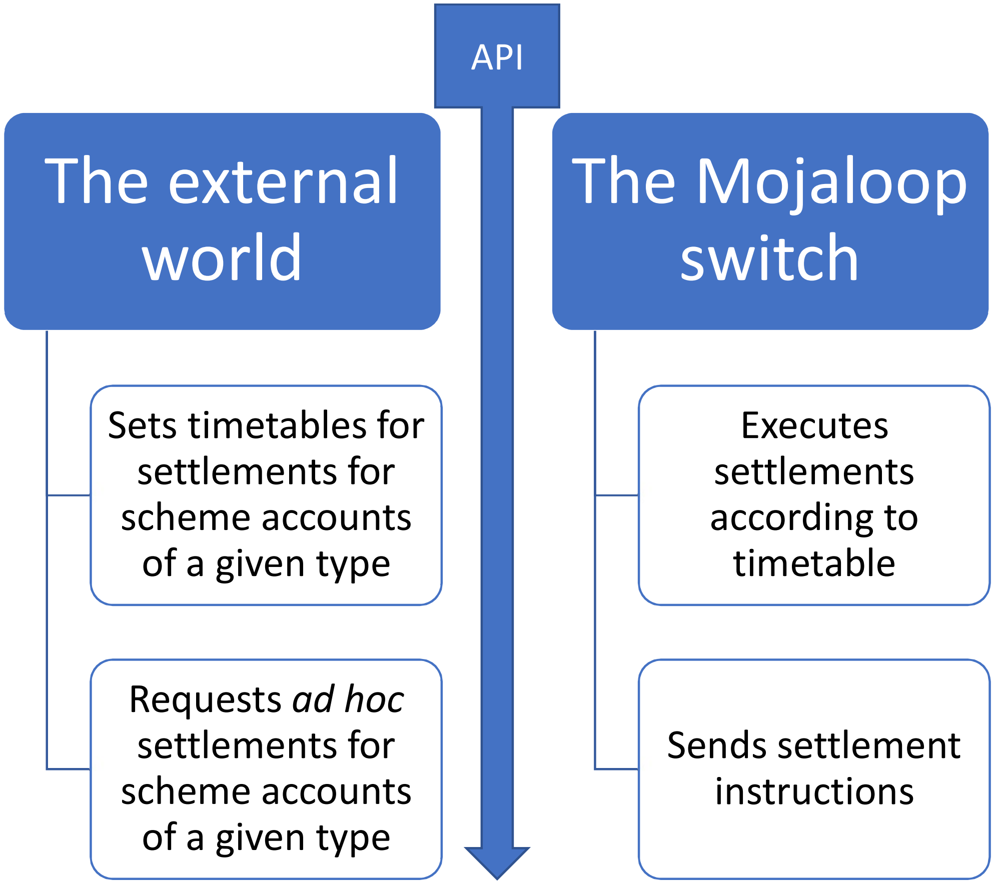
Settlement definition: characteristics

- Only applies to net settlements: by definition, gross settlements are transaction by transaction.
- The timetable for settlements is defined by the scheme
- Can be on a timetable or *ad hoc*
- Settlements can be either point-to-point (between two defined parties) or net

Settlement definition: open questions

- On point-to-point and net settlements, should the API support mixed models, or is this an either/or?
- If the API supports mixed models:
  - Assume that a point-to-point settlement overrides a net settlement
- Should the definition of a settlement imply closure of any currently open transaction window at the point of settlement?
- Should a settlement apply to all ledger groups, or should the administrator specify the ledger groups to which a settlement should apply?

Settlement definitions: who does what?



API

The external world

- Sets timetables for settlements for scheme accounts of a given type
- Requests *ad hoc* settlements for scheme accounts of a given type

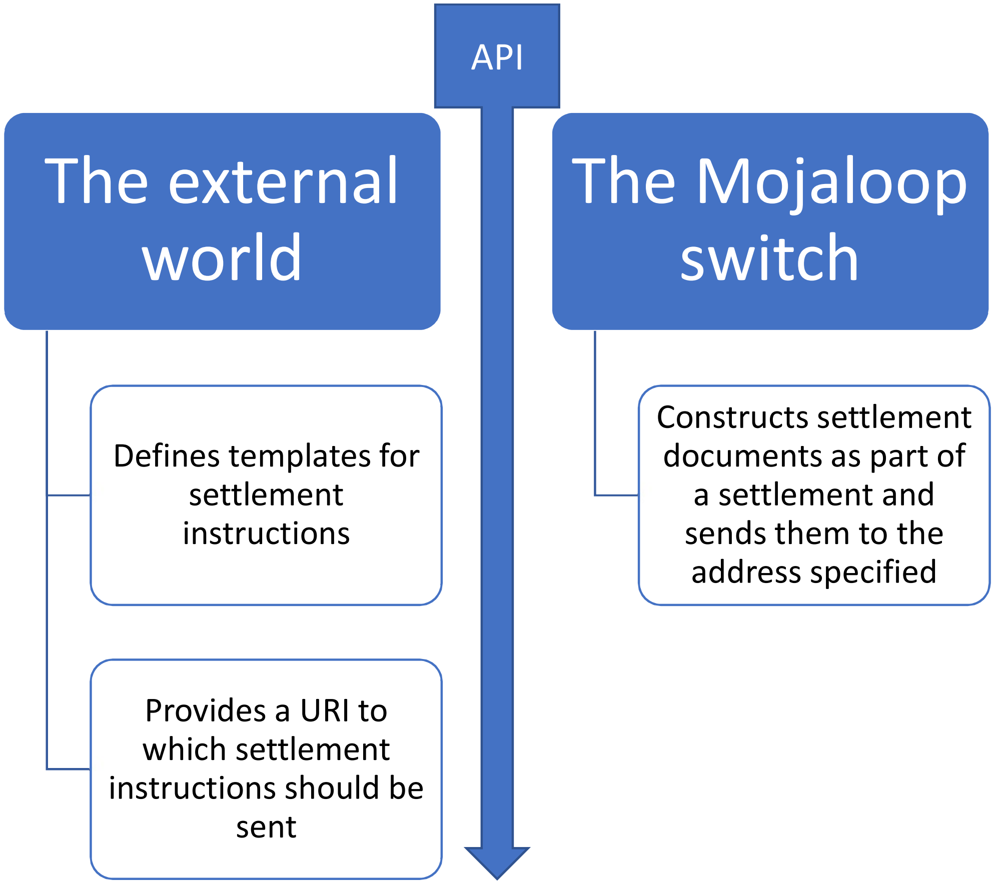
The Mojaloop switch

- Executes settlements according to timetable
- Sends settlement instructions

Settlement instructions: characteristics

- Apply to settlement accounts
- Need to define:
  - The form of an instruction to transfer funds from a net debtor or to a net creditor.
  - A URL to which that instruction should be sent

Settlement instructions: who does what?



API

The external world

Defines templates for settlement instructions

Provides a URI to which settlement instructions should be sent

The Mojaloop switch

Constructs settlement documents as part of a settlement and sends them to the address specified

Settlement reporting: characteristics

- Reports are produced by making GET requests on the following resources, which are described below:
  - accountMatrix
  - ledgergroupMatrix
  - transactionWindowContent
  - settlementContent
  - ndcMatrix

Net Debit Cap changes: characteristics

- Applies to gross and net settlement models
- The scheme can specify
  - An NDC margin by amount or percentage (positive or negative)
  - NDC applicability
    - Overall
    - For one or more individual DFSPs

What objects should the API support?

- schemeAccountType
- schemeAccount
- schemeAccountMatrix
- ledgerGroup
- ledgerGroupMatrix
- transactionWindowTimetable
- transactionWindowContent
- settlementTimetable
- settlementContent
- ndcParameters
- ndcMatrix

schemeAccountType

- Members:
  - Name: the name of the account type
  - SettlementModel: the settlement model to be followed for scheme accounts of this type
- Actions:
  - GET
  - POST
  - PUT
  - DELETE

schemeAccount

- Members:
  - Name: the name of the account
  - Type: the schema account type that this scheme account belongs to
  - Instruction: information required to issue an instruction as part of a settlement
    - Instruction: a template for instructing a system (e.g. RTGS) to transfer funds as part of a settlement
    - Destination: a URI specifying where the instruction should be sent as part of the settlement
  - Participants: a list of the DFSPs who participate in this account
    - fspId: for each participating DFSP, the ID and name of the participant
- Actions:
  - GET
  - POST
  - PUT
  - DELETE

schemeAccountMatrix

- Members:
  - An array of *schemeAccount* objects, one for each account in the scheme
  - A list of all DFSPs who are not associated with an account
- Actions:
  - GET

ledgerGroup

- Members:
  - Name
  - Id – an internal identifier set by the switch
  - settlementType – gross or net

ledgerGroupMatrix

- Members:
  - An array of ledgerGroupDetail objects:
    - A *ledgerGroup*
    - An array of *ledgerInformation* objects, one for each ledger of that type:
      - dfspName
      - Currency
      - An array of position objects:
        - Time
        - Position
- Actions:
  - GET

transactionWindowTimeTable

- Members:
  - An array of *transactionWindowEntry* objects:
    - One of:
      - A regular timetable in cron format
      - A start time and an end time
      - A start time only (= transaction window runs from start time until the start of the next transaction window)
      - An end time only (= transaction window runs from the end of the last transaction window before the end time defined until the end time defined)
      - Nothing (= transaction window runs from now until superseded by another transaction window)
      - A start transfer ID and an end transfer ID
      - A start transfer ID only (= transaction window runs from start transfer ID until the start of the next transaction window)
      - An end transfer ID only (= transaction window runs from the end of the last transaction window before the end transfer ID until the end transfer ID)
    - Settleable: a true/false indicator showing whether the transaction window(s) is/are able to be settled or not.
    - ID: if the transactionWindowEntry object represents a single transaction window, this field contains a unique identifier for the transaction window
- Actions:
  - GET
  - POST
  - PUT
  - DELETE

transactionWindowContent

- Members:
  - An array of *transactionWindow* objects:
    - ID: the unique identifier of the transaction window
    - startTime: the completion time of the earliest transfer contained in the transaction window
    - endTime: the completion time of the latest transfer contained in the transaction window
    - startId: the transfer ID of the earliest transfer contained in the transaction window
    - endId: the transfer ID of the latest transfer contained in the transaction window
    - Settleable: a Boolean value indicating whether the transaction window is settleable or not.
    - Settled: a Boolean value indicating whether the transaction window has settled or not
    - Optionally, an array of transfer objects representing the transfers which are included in the transaction window
- Actions:
  - GET
  - PUT

settlementTimeTable

- Members:
  - An array of *settlementEntry* objects:
    - One of:
      - A regular timetable in cron format
      - A start time and an end time
      - A start time only (= settlement window runs from start time until the start of the next settlement window)
      - An end time only (= settlement window runs from the end of the last settlement window before the end time defined until the end time defined)
      - Nothing (= settlement window runs from now until superseded by another settlement window)
      - A start transfer ID and an end transfer ID
      - A start transfer ID only (= settlement window runs from start transfer ID until the start of the next settlement window)
      - An end transfer ID only (= settlement window runs from the end of the last settlement window before the end transfer ID until the end transfer ID)
      - An array of transaction window IDs: the settlement should settle the transactions contained in these transaction windows.
    - ID: if the settlementEntry object represents a single settlement, this field contains a unique identifier for the settlement
- Actions:
  - GET
  - POST
  - PUT
  - DELETE

settlementContent

- Members:
  - An array of *Settlement* objects:
    - An array of *TransactionWindow* objects, representing the transaction windows contained in the settlements
    - An array of *SettlementParticipant* objects, representing the consequence of the settlement for each of its participants:
      - Participant name
      - Participant currency
      - Net credit or debit as a consequence of the transfers covered by this settlement
      - Optionally, an array of transfer objects representing the transfers which are included in the settlement
- Actions:
  - GET

Ndc

- Members:
  - An array of *NdcParameter* objects:
    - FspId: the ID of a participant to whom these parameters apply. If not present, this is a default which applies to all participants who do not have individual NDC parameters set.
    - Currency: the currency to which this NDC information applies. If not present, this is a default which applies to all currencies.
    - ndcAmount. This MAY NOT be present if the FspId is blank. If it is present, it represents the amount to be added to (if positive) or subtracted from(if negative) the participant's NDC value
    - ndcMargin. If this value is blank on a POST or PUT, it should be ignored. Otherwise, one of:
      - A positive or negative absolute value which is to be applied to the participant's balance in a settlement account (or to their share of a pooled account) to calculate their NDC. A positive amount means that the participant is allowed to transact above their current settlement amount; a negative amount means that the participant is required to hold settlement guarantees above their ability to transact. Any value above 1 or below -1 should be interpreted as a value of this type.
      - A positive or negative percentage value which is to be applied to the participant's balance in a settlement account (or to their share of a pooled account) to calculate their NDC. A positive amount means that the participant is allowed to transact above their current settlement amount; a negative amount means that the participant is required to hold settlement guarantees above their ability to transact. Any value greater than or equal to -1 and less than or equal to 1 should be interpreted as a value of this type.
    - alertLevel. One of:
      - A positive absolute value which generates an alert if the participant's position moves within the defined amount of the current level of their NDC. Any value above 1 should be interpreted as a value of this type.
      - A positive percentage value which generates an alert if the participant's position moves within the defined percentage of the current level of their NDC. Any value less than 1 should be interpreted as a value of this type.
      - Zero. A value of zero means that no alerts are required.
- Actions:
  - GET
  - POST
  - PUT
  - DELETE

NdcMatrix

- Members:
  - An array of *ndcPosition* objects:
  - FspId
  - Currency
  - An array of *NdcPositionPoint* objects:
    - Time
    - NDC absolute
    - NDC adjusted
    - Exposure
- Actions:
  - GET