mojaloop

ssnapp™

Split Payment and Settlement API

What is the SSNAPP Mojaloop Goal?

- **Split Payment and Settlement API**
  - Split payments between multiple DFSPs
  - Settle the payments on an
    - “Any or All” or
    - “All or Nothing” basis

The 5 “W”s

Who

- Customers
- Merchants
- Governments

What

A way for a single payment from a single PAYER to be automatically split and settled among multiple PAYEES

When

A PAYER makes a single purchase of serveral related items or services from multiple vendors, including VAT-type taxes

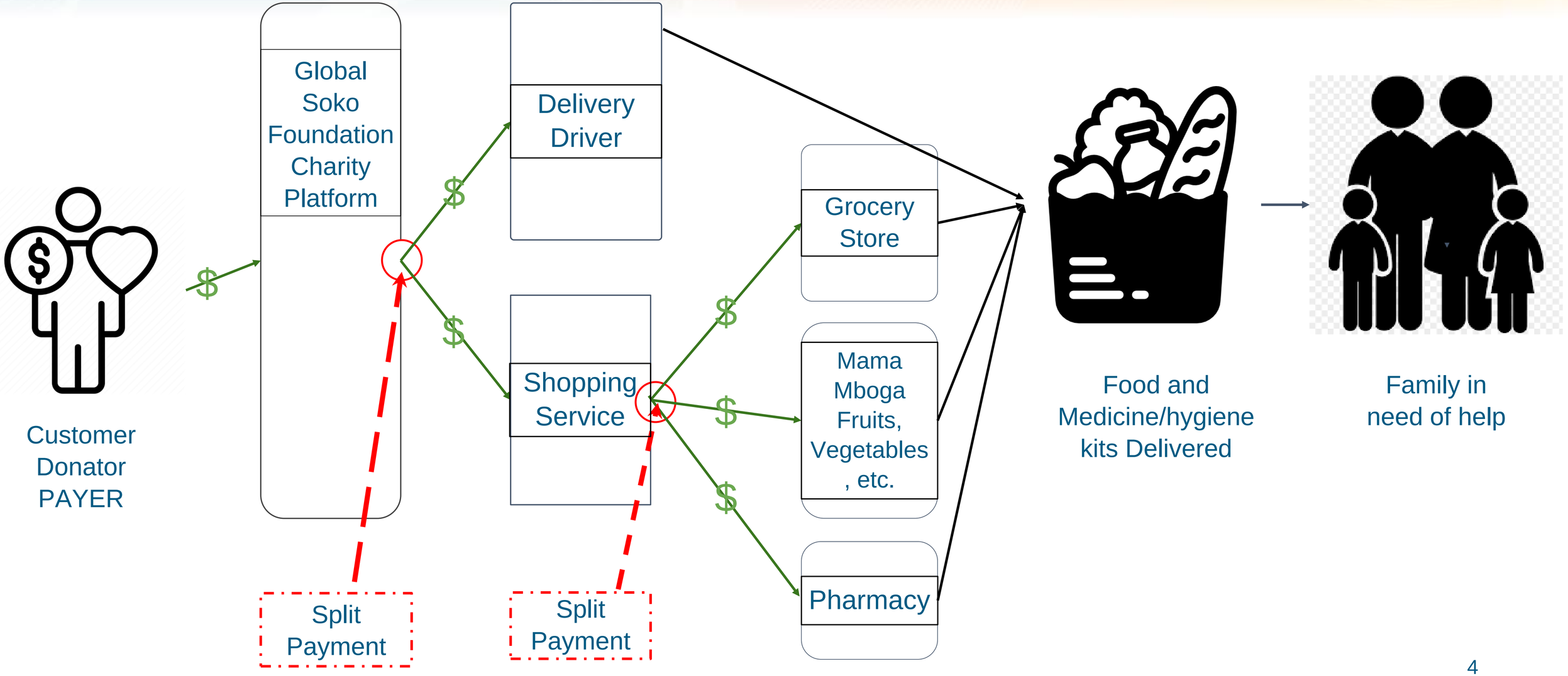
Where

- E-commerce Stores
- POS Systems
- Charitable Organizations
- Share Economy Platforms

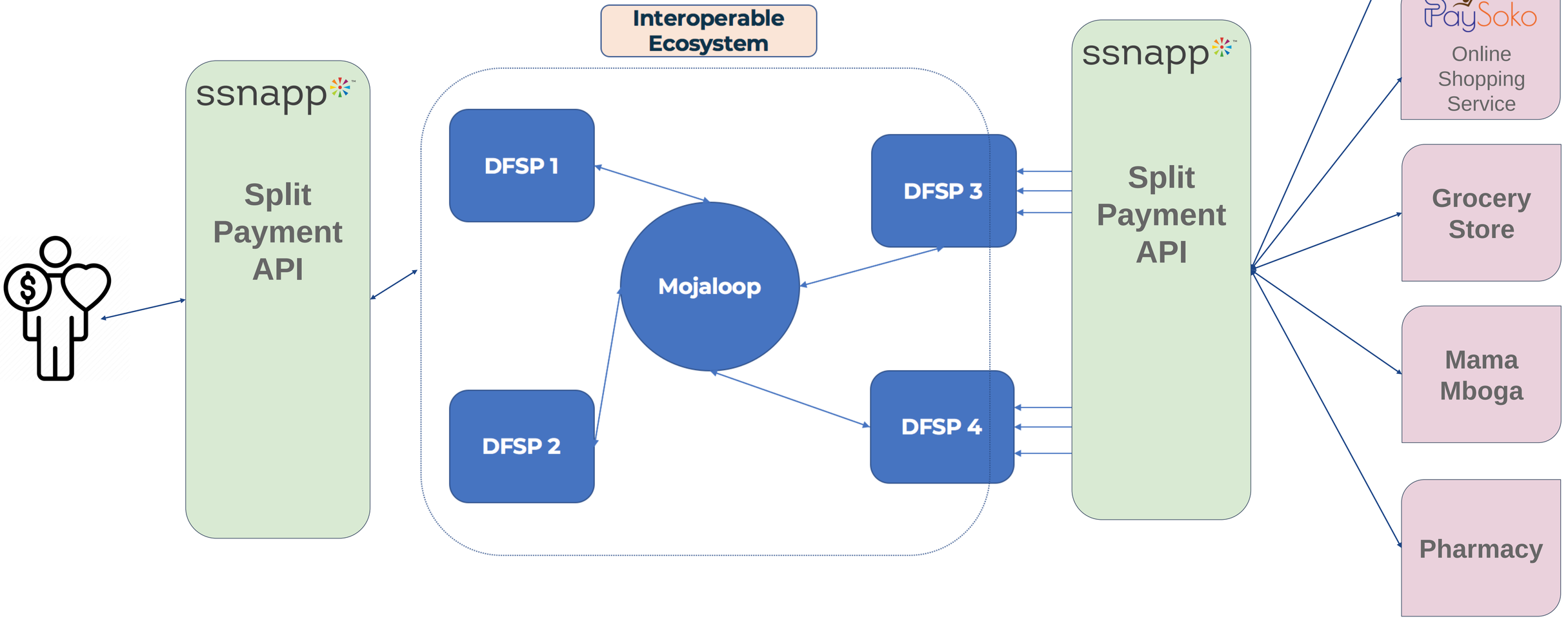
Why

- Trust
- Transparency
- Traceability
- Efficiency
- Ease of use
- Ease of deployment
- Cost Effective

Example 1
Charity Platform – Any or All



Split Payment Gateway API



Example 2

Split Payment to Multiple Vendors – All or Nothing Compound Transaction

- A Customer buys a Ticket to a football game
- The Customer's single PAYER payment should be resolved into a number of separate payments to individual PAYEES; for example:
  - *The Football Club receives the price of the Ticket*
  - *The Ticketing Agency receives a commission for their services*
  - *The Government receives the tax that is due on the Transaction*
- Formally, this is a single Compound Transaction: if any part of it fails, the entire Transaction fails
- The PAYEE accounts that receive payment may belong to different DFSPs

Example 2

Split Payment to Multiple Vendors – All or Nothing

An outline of the desired process

- The Customer makes a (single) payment
- The funds are deducted from the Customer’s account as a lump sum
- The funds are then split into the Transaction’s component parts
- Each PAYEE DFSP quotes independently to effect its part of the overall Transfer
- Each PAYEE DFSP postpones commitment of its part of the funds until it is notified by the Switch that it may commit
- The PAYER DFSP is notified of the success of the overall Payment Transaction

Example 2
Scenario

- A Customer buys a Ticket for City Football Club’s match
- The Ticket is purchased through an Agent
- The Ticket costs 100 ZAR
- The Agent knows (and has informed their DFSP of this) that:
  - 80% goes to City Football Club
  - 10% is a commission fee payable to the Agent
  - 10% is tax payable to the Government

The Sequence of Mojaloop Actions
Identification

1. The Customer asks to pay the Agent
2. The Customer's DFSP asks Mojaloop:
    - *Who should I route the payment to?*
3. Mojaloop contacts the Agent's DFSP:
    - *Can I route payments with this identifier to you?*
4. The Agent's DFSP responds:
    - *Yes*

- All this happens in exactly the same way as now…

The Sequence of Mojaloop Actions
Agreement

1. The Customer's DFSP asks the Agent's DFSP (via Mojaloop) for a Quotation to perform the Transfer
2. The Agent's DFSP recognises that this payment needs to be split
   - *We assume that this is done by the identifier …*
3. The Agent's DFSP responds with an error code indicating that this is a Split Payment Compound Transaction
   - *We need a new error code to convey this information …*
4. The Customer's DFSP recognises that this will be a Compound Transaction, so it doesn't cancel quite yet

- All this requires a new error code …

The Sequence of Mojaloop Actions

Request for Payment

1. Now, the Agent's DFSP asks Mojaloop for the DFSPs that represent the people who are going to be credited as part of this Transfer
   - *This happens in the same way as now*
2. The Agent's DFSP sends a three part Request to Pay to the Customer's DFSP
   - *Send 80 ZAR to the Football Club's account, 10 ZAR to the Agent's account and 10 ZAR to the Government's tax account …*
   - *This is an extension of the call: at present, you can only make a single Request for Transfer per call …*
3. The Customer's DFSP receives the Request to Pay

The Sequence of Mojaloop Actions

Multiple Requests for Quotation

1. The Customer's DFSP sends out a Bulk Request for Quotation with three parts, one for each party who needs to be credited
   - *This assumes that the extensions to Bulk Transfers required to support bundling of Requests to multiple DFSPs have been implemented*
2. The Switch breaks out the Transfer Requests and sends one to each PAYEE DFSP
3. Each PAYEE DFSP responds with its own Quotation, and signs the Quotation with its own private key
4. The Switch assembles the Responses and sends them back to the Customer's DFSP

The Sequence of Mojaloop Actions
Multiple Requests for Transfer

1. The Customer's DFSP sends out a Bulk Transfer Request with three parts, one for each party who needs to be credited
    - *This assumes that the extensions to Bulk Transfers required to support bundling of Requests to multiple DFSPs have been implemented*
    - *There will be a new flag on the Bulk Transfer Request. If it is set, it means that an error in any member of this set of Transfers means that all the members should be cancelled*
2. The Switch breaks out the Transfer Requests and sends one to each creditor party
3. Each Request contains the flag that says to the PAYEE DFSP:
    - *Don't commit your funds to the Customer until you receive final confirmation*

The Sequence of Mojaloop Actions

Multiple Requests for Confirming the Transfer

1. Each PAYEE DFSP responds to the Switch using the proposed syntax for: Please inform me when you have agreed the Transfer
    - *This assumes that the proposed extension to the Transfer Response syntax has been implemented*
2. If any PAYEE DFSP responds negatively, then:
    1. The Switch sends a Cancellation Request to all the PAYEE DFSPs, and they cancel the commit
    2. The Switch sends an Error Response to the PAYER DFSP
3. If all PAYEE DFSPs respond positively, then:
    1. The Switch informs the PAYER DFSP that the Transfer was successful
    2. The Switch informs each PAYEE DFSP that the Transfer was successful and it can clear the funds to the PAYEE account

Thank You