Merchant Scheme Extensions + QR Support

Lewis Daly, Justus Ortlepp, Paul Makin

PI18 - April 2022

Agenda

- Progress Report
- Demo

Learnings from a National Deployment

Initially focussed on interoperable merchant payments in a complex customer environment

For now, customer-initiated; merchant-initiated (MRTP) and fintech/app initiated (PISP/3PPI) are further down the roadmap

Customer uses either static QR or USSD: Merchant receives payment “instantly”, releases the goods

Interoperable across all connected DFSPs (banks, MFIs, MMOs, SACCOs etc)

What We've Been Doing

Merchant Payments

Developing customer-initiated merchant payments infrastructure as a modular extension around a Mojaloop Hub

Underlying Concepts

Using a merchant registry and merchant IDs to support fraud/ML detection

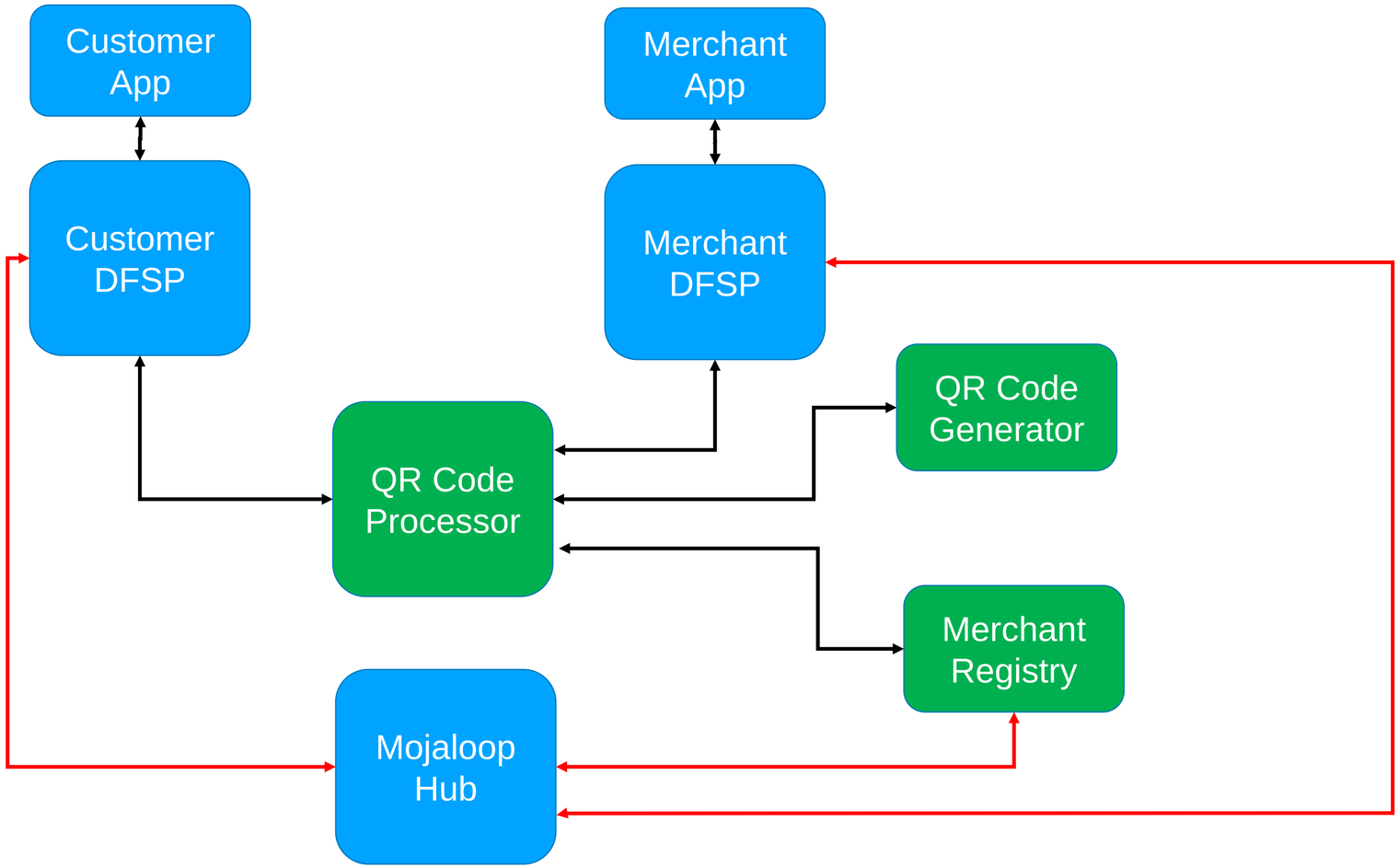
Independent of front-end tech; initially to support USSD and QR Codes (static and dynamic)

Principles

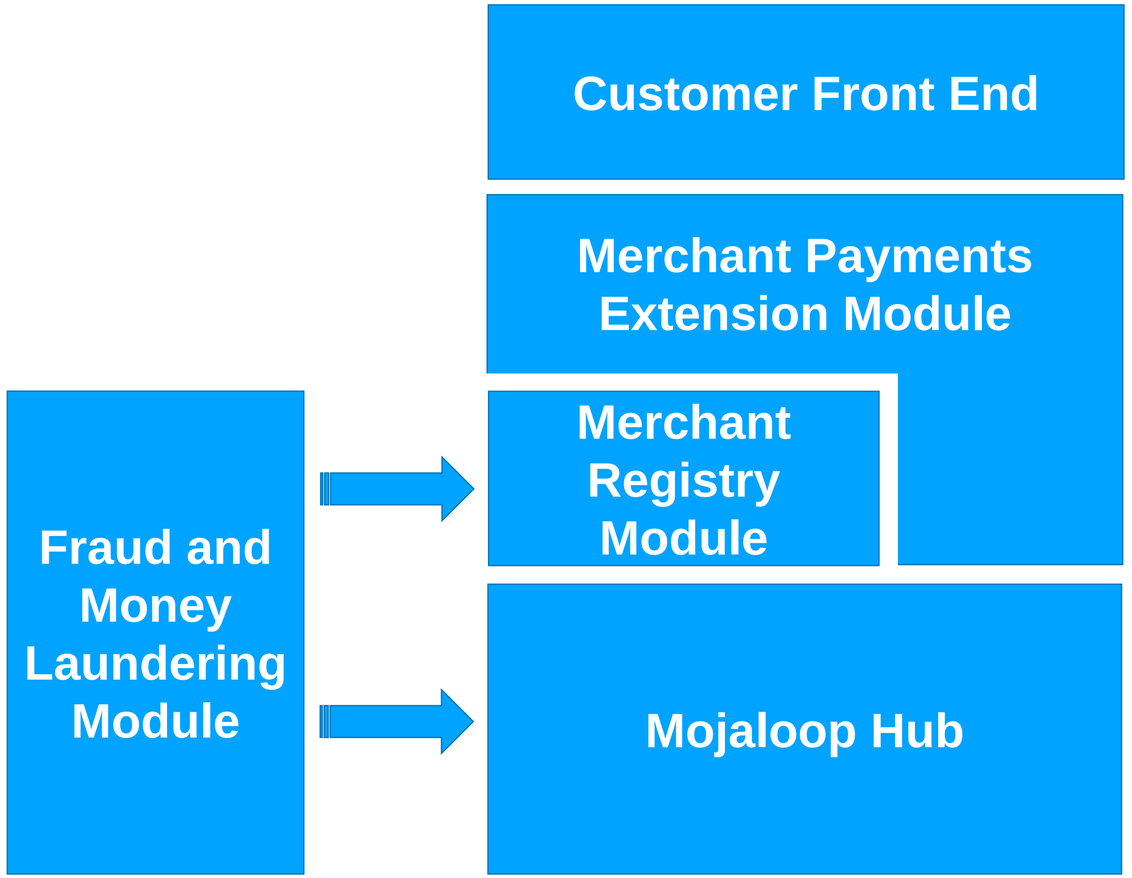
Merchant payments aren’t straightforward. We took the decision to start at the “simple end”, and build from there, using the following principles:

- Build on Mojaloop P2B instant payments
- Expose merchant payment details, but “anonymise” them using merchant IDs
- Require merchant registration, so we can (1) support merchant IDs, and (2) identify and control fraudulent activity
- Develop an approach that supports multiple customer payment initiation methods, as appropriate to each customer (QR Code obviously, but also app, USSD)

Dynamic QR Payments: Architectural Elements



Envisaged Merchant Payments Ecosystem



Considerations: Merchant Registry

- Looked at the Mojaloop ALS/Oracle as a candidate for implementation
  - Looks good – but will need extension
  - Firm up on additional data requirements

- Looks in three directions
  - Supports transaction routing
  - Validates individual transactions against AML, Fraud assessment
  - Works with an external fraud/ML/KYB engine to evaluate merchant status dynamically

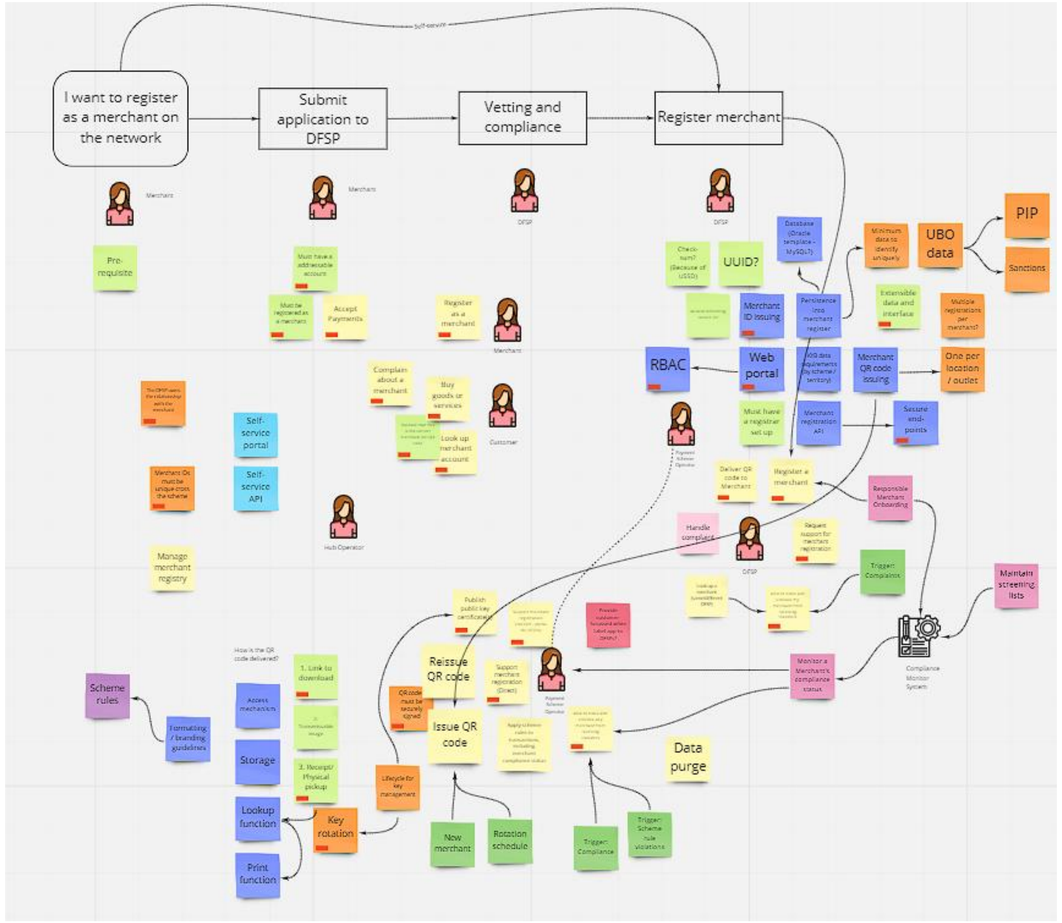
Considerations: Customer Front End

- Merchant Payment Extension Module can support multiple payment front ends
  - (1) Customer-initiated; (2) MRTP and 3PPI/PISP coming later
  - Currently validated against USSD and static QR; design in place for dynamic QR
- Need to engage with major in-country retail payments ecosystem players
  - Push; not pull

Lessons for a Successful Merchant Payment Solution

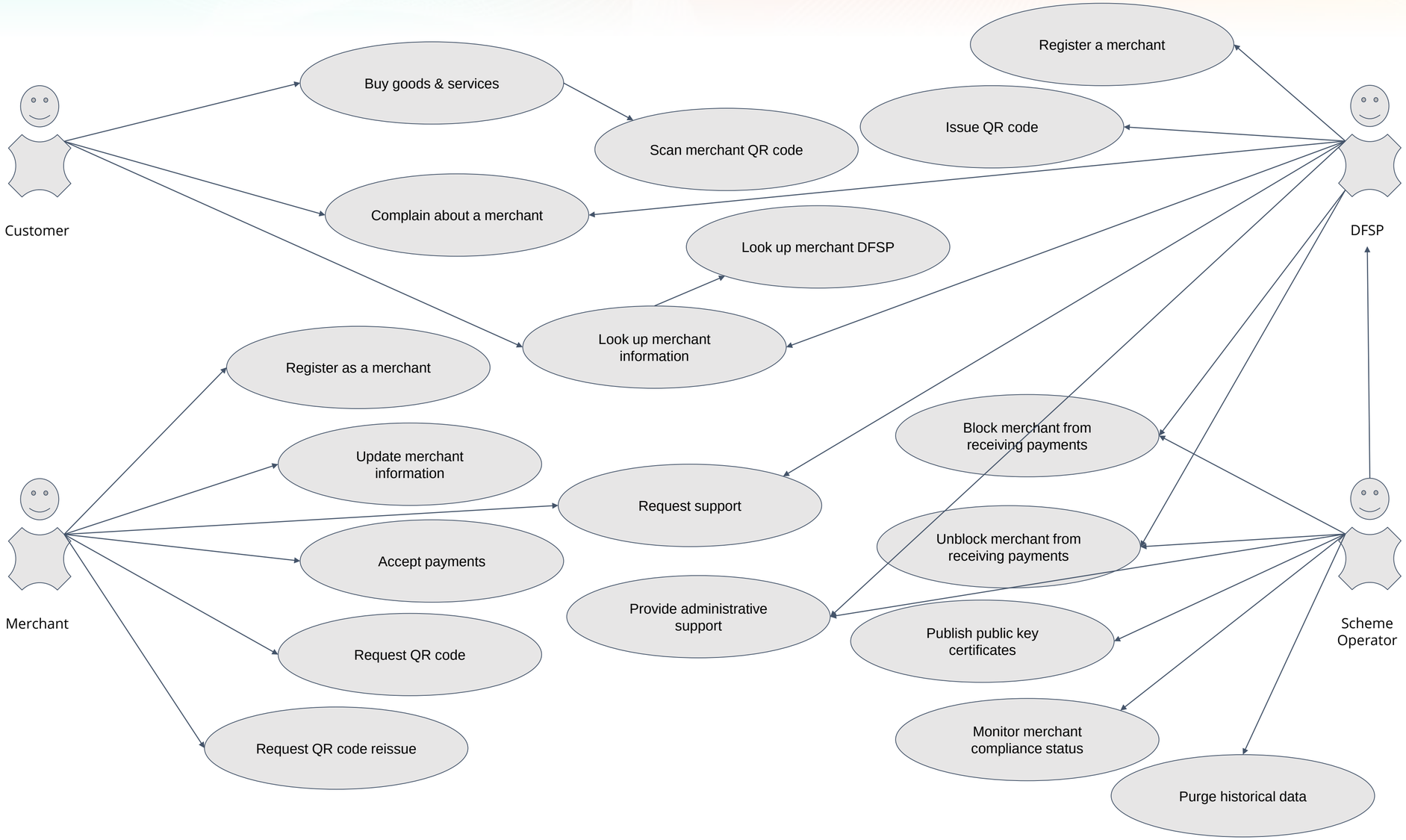
- Separate Rails from Services
- Interoperability – business, as well as technical
- Make it free
- Make sure customers can use it

Merchant Backlog - Approach



- Stakeholder definition
  - Merchants
  - Customers
  - DFSPs
  - Payment Scheme Operator
- Use Case identification
  - Stakeholder objectives
- Enabler specification
  - Infrastructure, services, processes

Merchant Backlog - Use Cases



Merchant Backlog - Features

- API accessible micro-services
- Online portal for user interaction
- Role-based access control
- Multiple mechanisms for QR code delivery
- Extensible data and interface
- Centralised merchant registration and ID issuing
- Securely signed QR codes

Demo

Goal: Simple Static QR Code demo payment flow:

Customer → Scan QR Code → Enter Price → Pay Merchant → Release Goods

Assumptions:

- Pre generated static QR code that encodes a Merchant ID
- No Merchant Registry: just reuse our existing oracle

Thank you!

Comments/Questions?