Merchant Scheme Extensions + QR Support

Justus Ortlepp, Paul Makin, Lewis Daly, Sam George

PI-17 - January 2022

Outline

- PI 16 Summary
- Merchant Payments Context
- Merchant Backlog Progress Update
- Demo Plan

PI 16 Goal - Merchant Blocklisting Showcase

Confidence: ?

Goal

Mocked Merchant Registry + Blocklist Prototype

Key Epics

1. **Demonstration of the blocklist & document CRUD processes - change the state to facilitate interdiction**
A - Actio do up date blocklist state
B - (Hub/Discovery/other) to recognise state and block a lookup; identify best place for this

3. **Roadtest our thinking** at hackathon event in Jan/Feb to get more contribution

2. Build out backlog for full merchant registry/ALS/Oracle implementation

Not Doing now but important next & Risk / Issues

1. Any changes to APIs for merchant lifecycle management;
- but we'll know the gaps by the end of this work in more detail

2. Maybe Static QR demo

Extra resource/skills that would be valuable:
- PROTOTYPERS wanting to support Lewis to imagine the demo in the real world
- Product/Biz analysts - to specify the demo storylines in detail
- Github folk - to help us create the gap backlog for PI-17
- **We could consider a Mojaloop Community Hack Week to get this done end to end like the ISO20022 week? A design challenge?**

Risks you might need to track:
- AVAILABILITY OF SME AND RESOURCES TO ACHIEVE THIS - ITS AMBITIOUS

Success Defined How?

Able to demonstrate merchant blocklisting in action

Do we have a backlog of tasks to do this properly?

mojaloop foundation

PI 16 Goal - Merchant Blocklisting Showcase

Confidence: ?

Goal

Mocked Merchant Registry + Blocklist Prototype

Key Epics

1. **Demonstration of the blocklist [illegible] processes - change the state to fa**
A - Actio do up date blocklist state
B - (Hub/Discovery/other) to recognise state and block a lookup; identify best place for this

Didn't Happen

3. **Roadtest our thinking** at hack [illegible]get more contribution

Didn't Happen

2. Build out backlog for full merchant registry/ALS/Oracle implementation

In Progress

Not Doing now but important next & Risk / Issues

1. Any changes to APIs for merchant lifecycle management;
- but we'll know the gaps by the end of this work in more detail

2. Maybe Static QR demo

Extra resource/skills that would be valuable:
- PROTOTYPERS wanting to support Lewis to imagine the demo in the real world
- Product/Biz analysts - to specify the demo storylines in detail
- Github folk - to help us create the gap backlog for PI-17
- **We could consider a Mojaloop Community Hack Week to get this done end to end like the ISO20022 week? A design challenge?**

Risks you might need to track:
- AVAILABILITY OF SME AND RESOURCES TO ACHIEVE THIS - ITS AMBITIOUS

Success Defined How?

Able to demonstrate merchant blockli

Didn't Happen

Do we have a backlog of tasks to do this properly?

In Progress

mojaloop foundation

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're 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

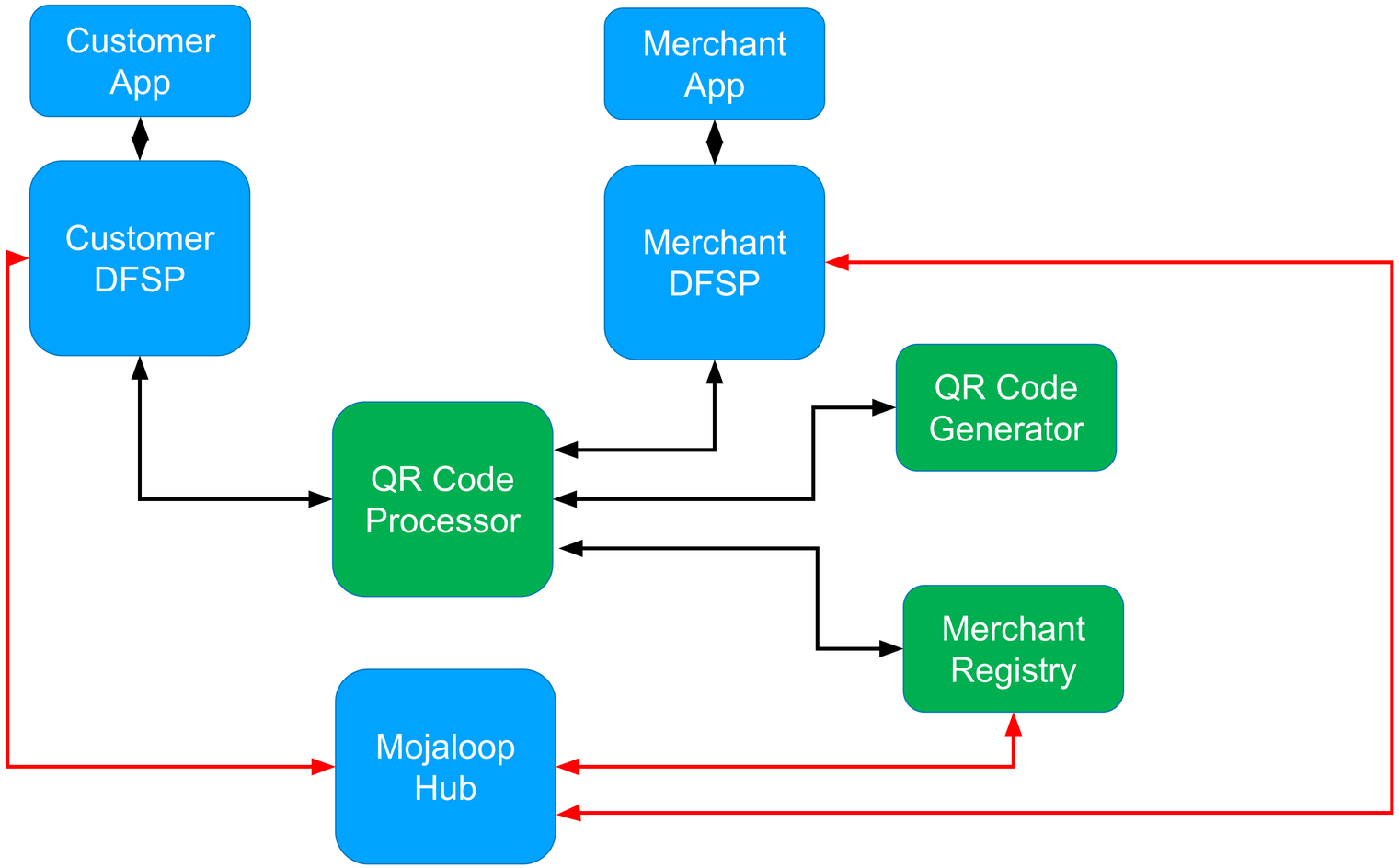
Independent of front-end tech; initially to support USSD and QR Codes (static and dynamic)

Principles

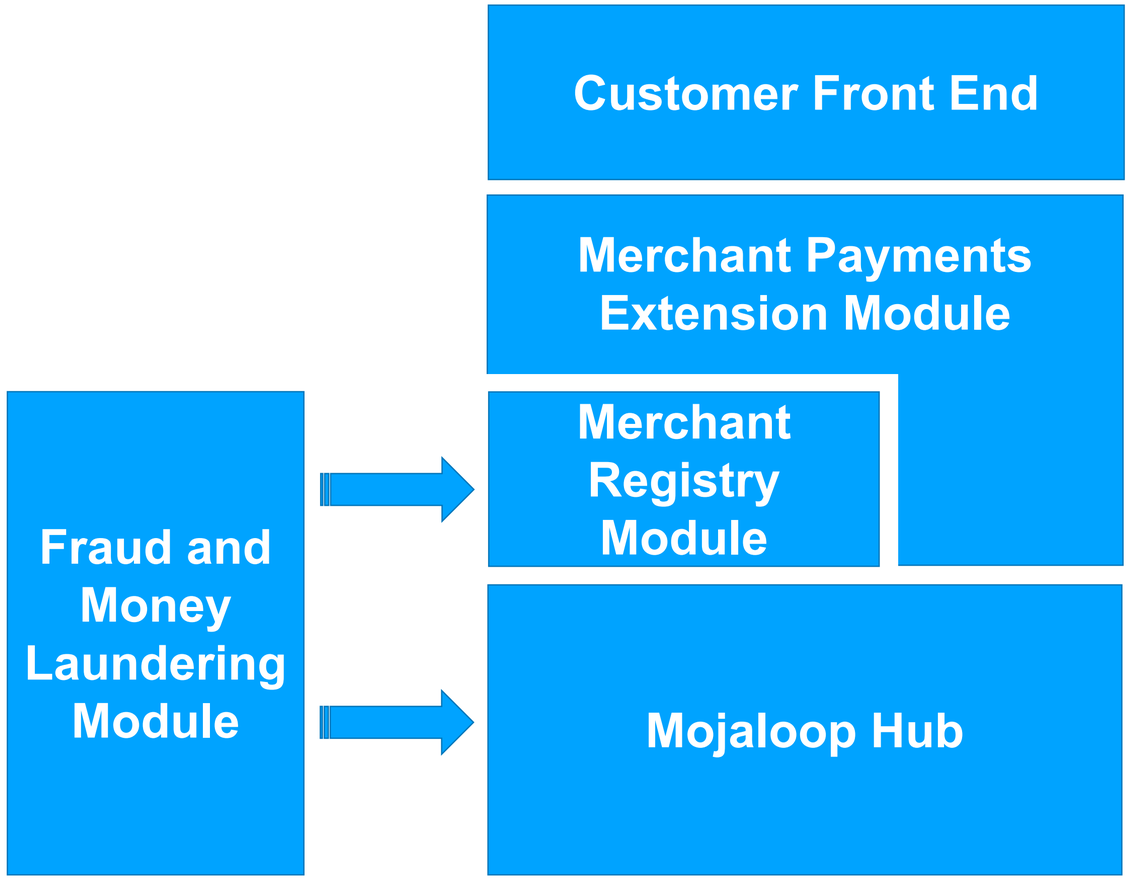
Merchant payments aren't straightforward. We took the decision to start at the "simple end", and build from there, using the following principles:

- 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

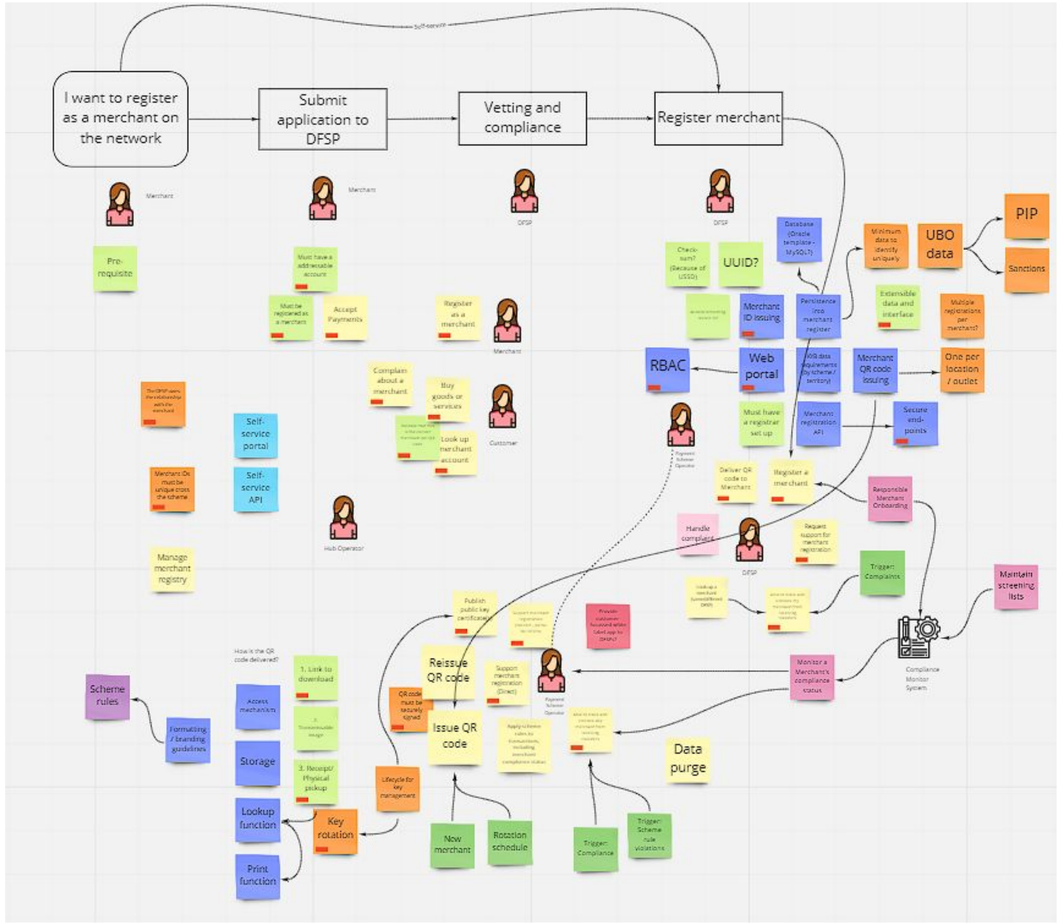
Considerations: Customer Front End

- 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

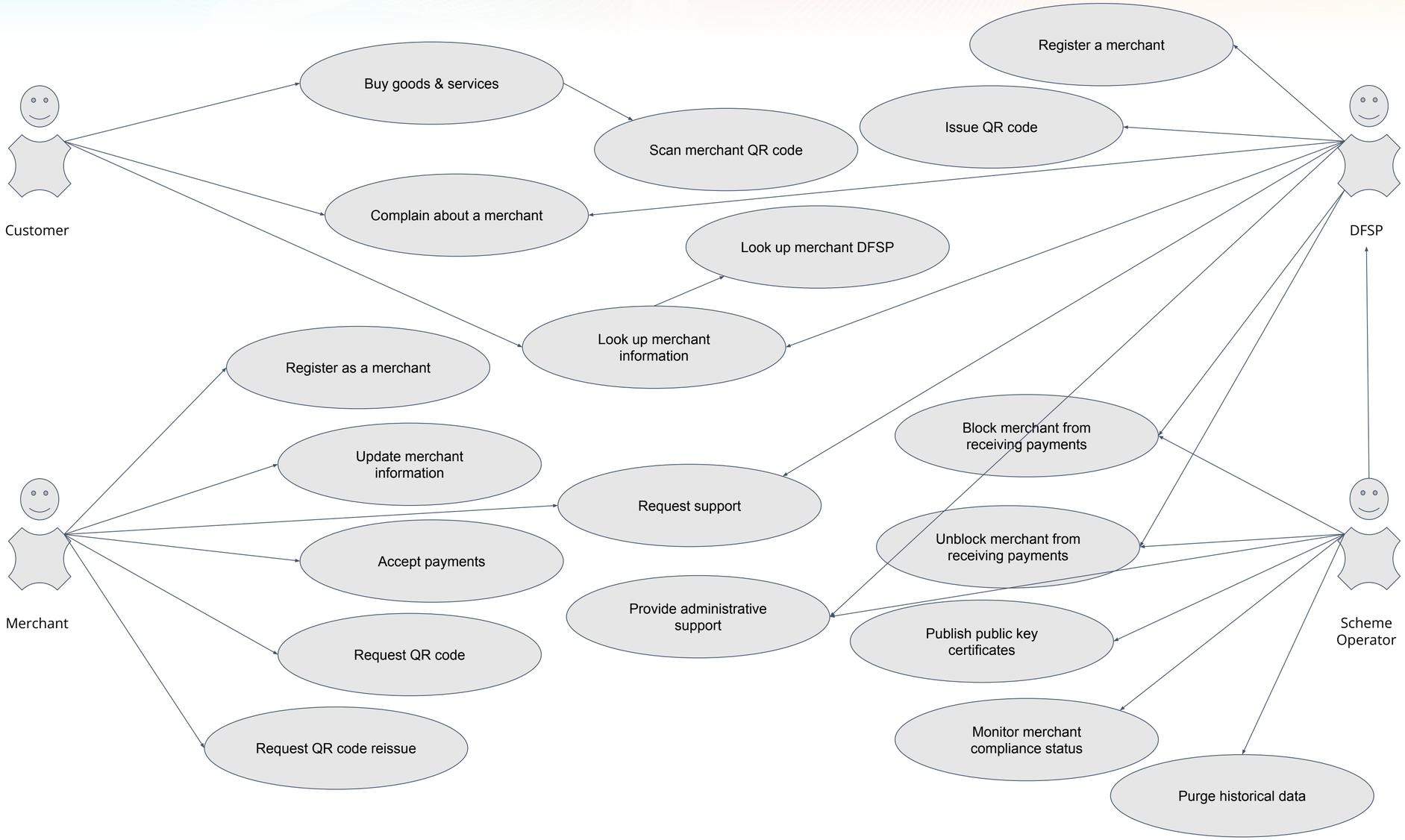
- Separate Rails from Services
- Interoperability – business, as well as technical
- Make it free
- Make sure customers can use it

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*

** - So far…*

Demo Plan

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 a reuse our existing oracle

Demo Plan

- Build upon Coil’s Demo Mojawallet codebase for a web-based demo.

Definition of done:
A locally running wallet that can initiate a QR payment in the Mojaloop Sandbox

We have a trusty intern lined up do this for us!

Thank you!

Comments/Questions?