mojaloop foundation

Mojaloop Fraud Risk Management

Justus Ortlepp – Product Owner, LexTego
Rob Reeve – CEO, LexTego
Greg McCormick – Director, Sybrin
Jason Darmanovich – Architect, Sybrin

AGENDA

- The Introduction
  - The journey
  - The Fraud Risk Management solution
- The Proof of Concept
  - The concept
  - The integration architecture
  - The ***Actio*** architecture
  - The fraud risk typologies
  - The ***APRICOT*** model, part IV
  - The data factory
  - The demo presentation
- The Next Steps

The journey so far…

PI 9

Fraud Risk was selected for further work “To review and classify the typologies to determine which of those strategically fit with Mojaloop’s vision and how to get started building it”.

- The development of a strategic assessment framework
- The detailed classification of the risk typologies already identified
- A detailed cross-reference between the risk typologies and the data dictionary already developed

PI 10

Fraud Risk Management was selected as a work-stream for PI10 with the broad objectives to define, investigate and validate a backlog and MVP for a FRM system/service against the APRICOT modelling for existing/prospect operators; and identify partners to build / implement a FRM system / service.

Objectives for PI11 and beyond were identified and prioritized.

PI 11

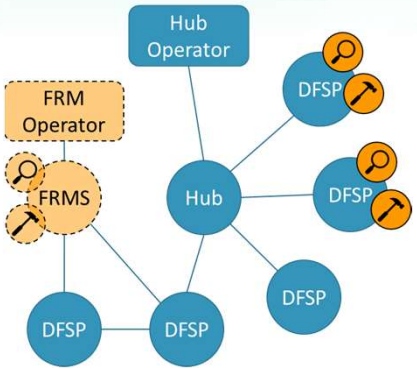
The Fraud Risk Management workstream had following tasks for this program increment:

- Identify the typologies that are visible to the hub and can be monitored by the hub
- Identify additional typologies related specifically to fraud “at the hub”
- Consider the feasibility of an Open Source Software Fraud Risk Management solution

As a result 39 new typologies were identified. These typologies are core fraud risks that a hub faces and have to mitigated at hub level.

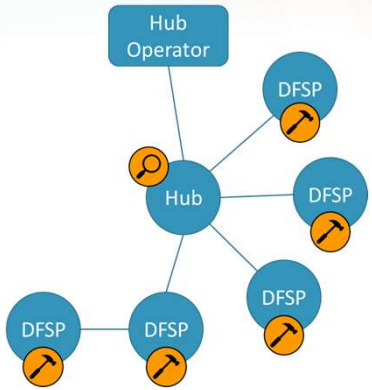
mojaloop foundation

Operating model configurations



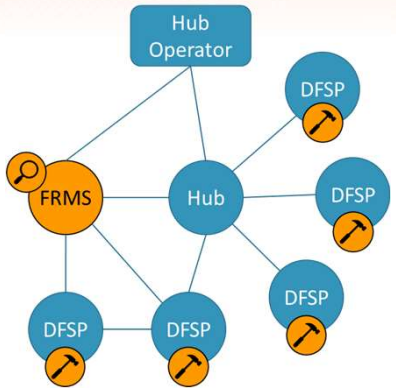
Distributed

- No fraud risk detection or management capability or responsibility by the switch Operator
- Some FSPs perform detection on internal (on-us), incoming and outgoing transactions
- Those FSPs would employ compliance teams to investigate fraud and financial crime risk alerts



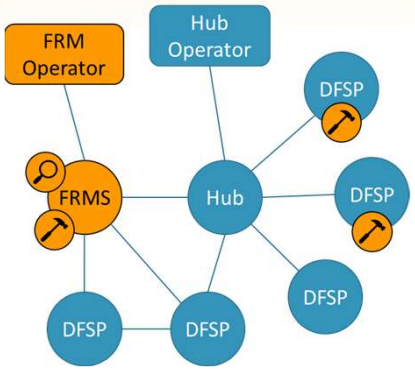
Embedded

- Centralised fraud risk detection service hosted by the switch Operator
- The Operator performs detection on all transactions routed through the switch
- Each FSPs would employ compliance teams to investigate fraud and financial crime risk alerts issued by the Operator



Semi-attached

- Centralised fraud risk detection service hosted by the switch Operator
- Separate interface to receive transactions from switch participants and non-participants
- The Operator performs detection on all transactions routed to the FRM service
- Each FSPs would employ compliance teams

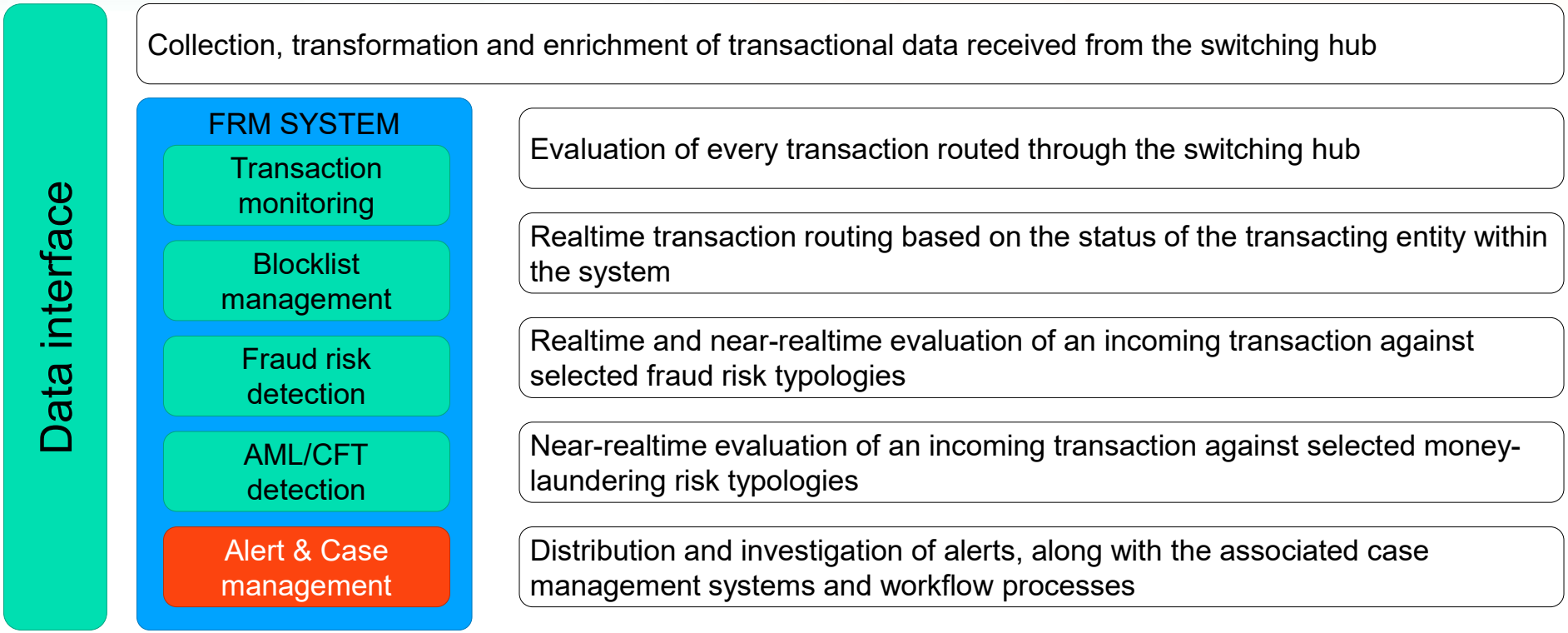


Standalone

- Autonomous and independent fraud risk detection and management service hosted by an FRM Operator
- Discrete fraud detection
- Outsourced fraud management
- The FRM Operator performs detection on all transactions routed to the FRM service
- Shared, centralised compliance services

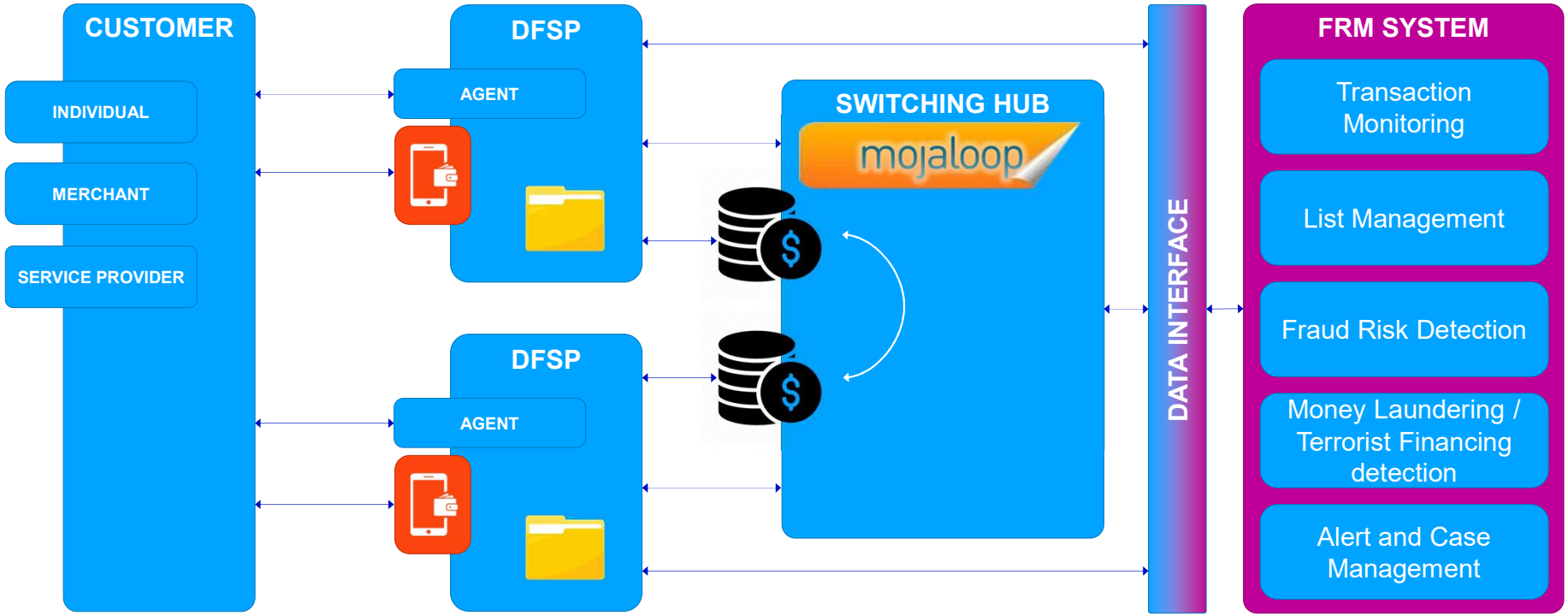
mojaloop foundation

FRM: Programme Increment 12



mojaloop foundation

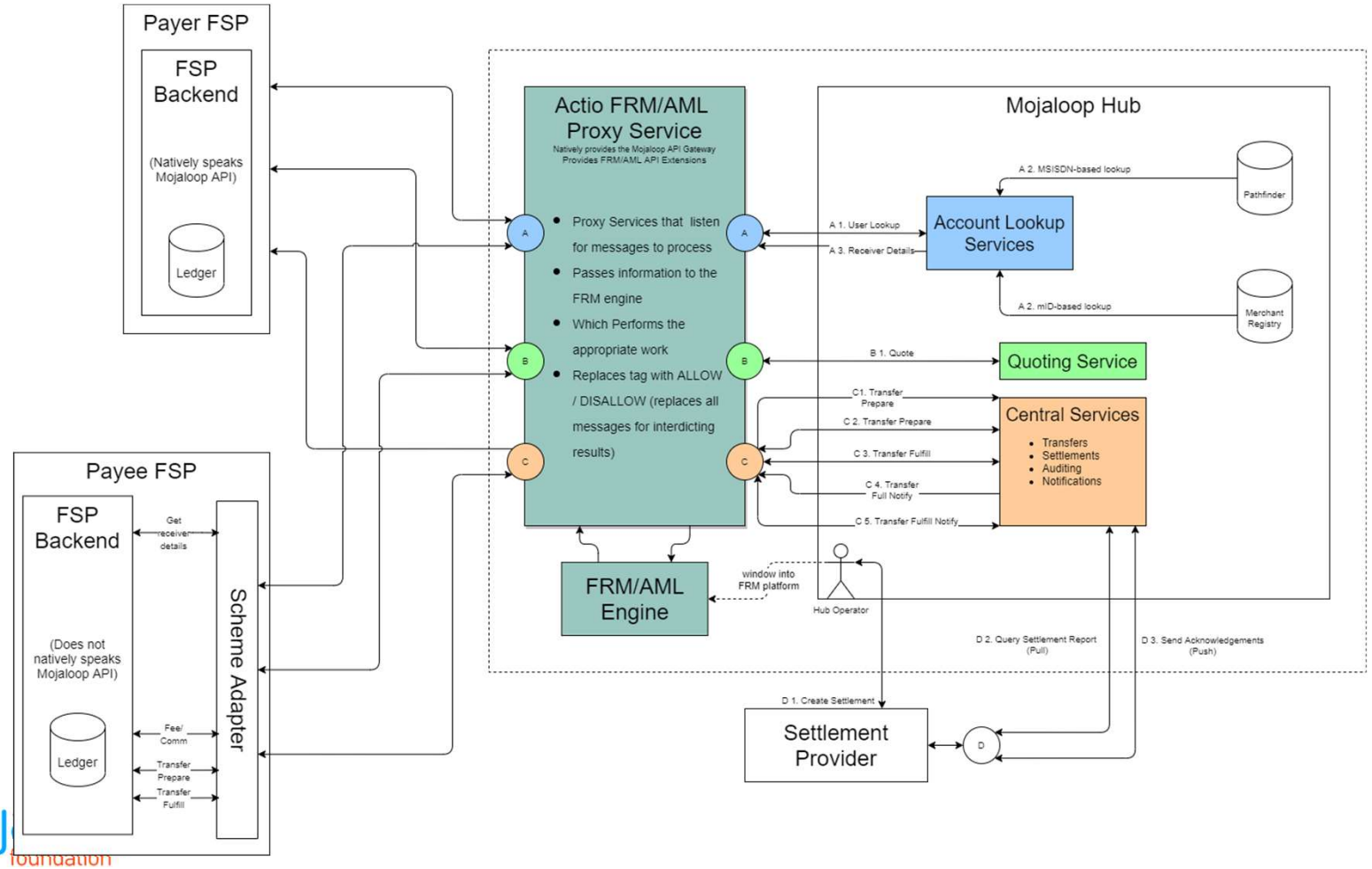
The context



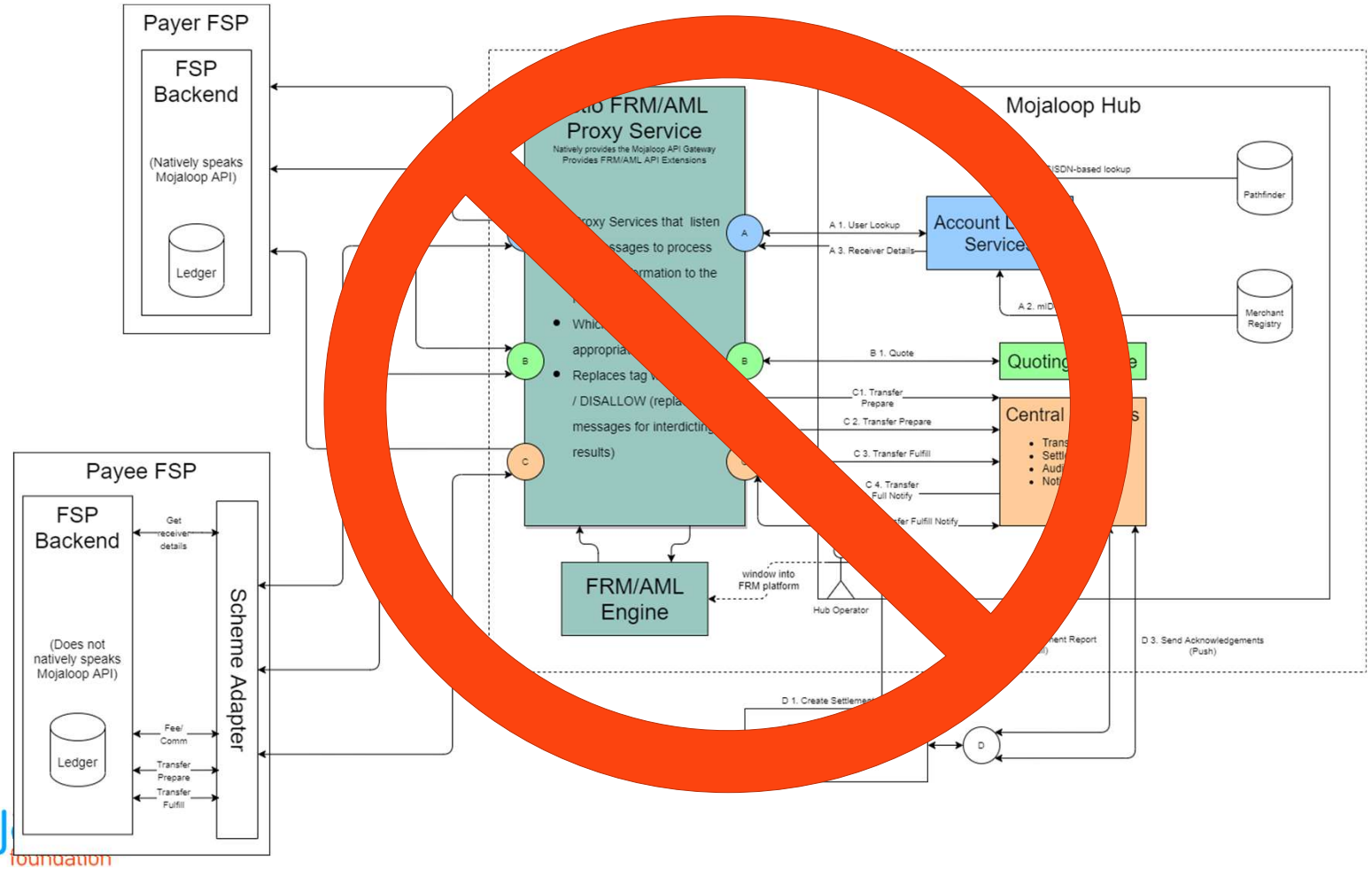
mojaloop foundation

ARCHITECTURE

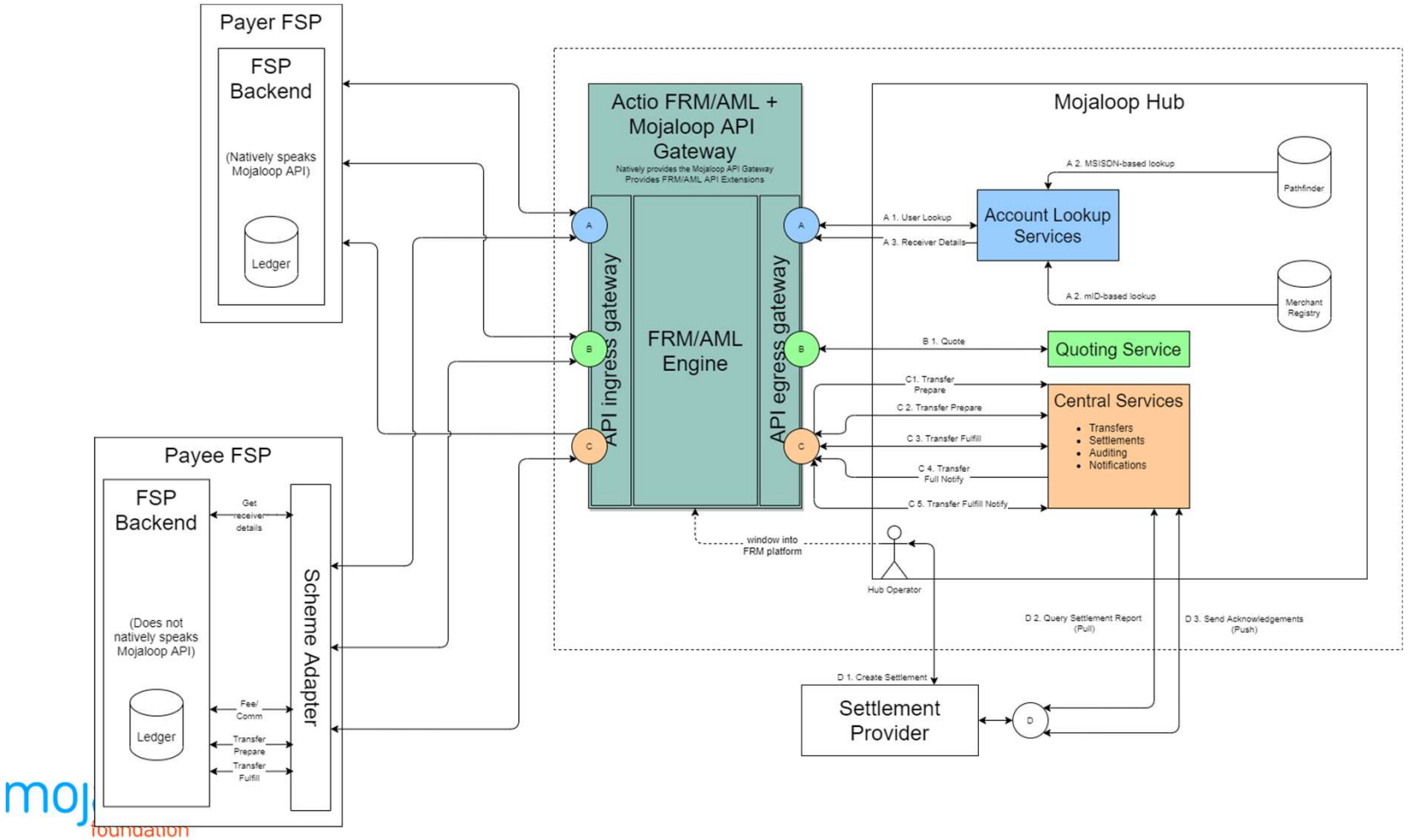
The integration architecture



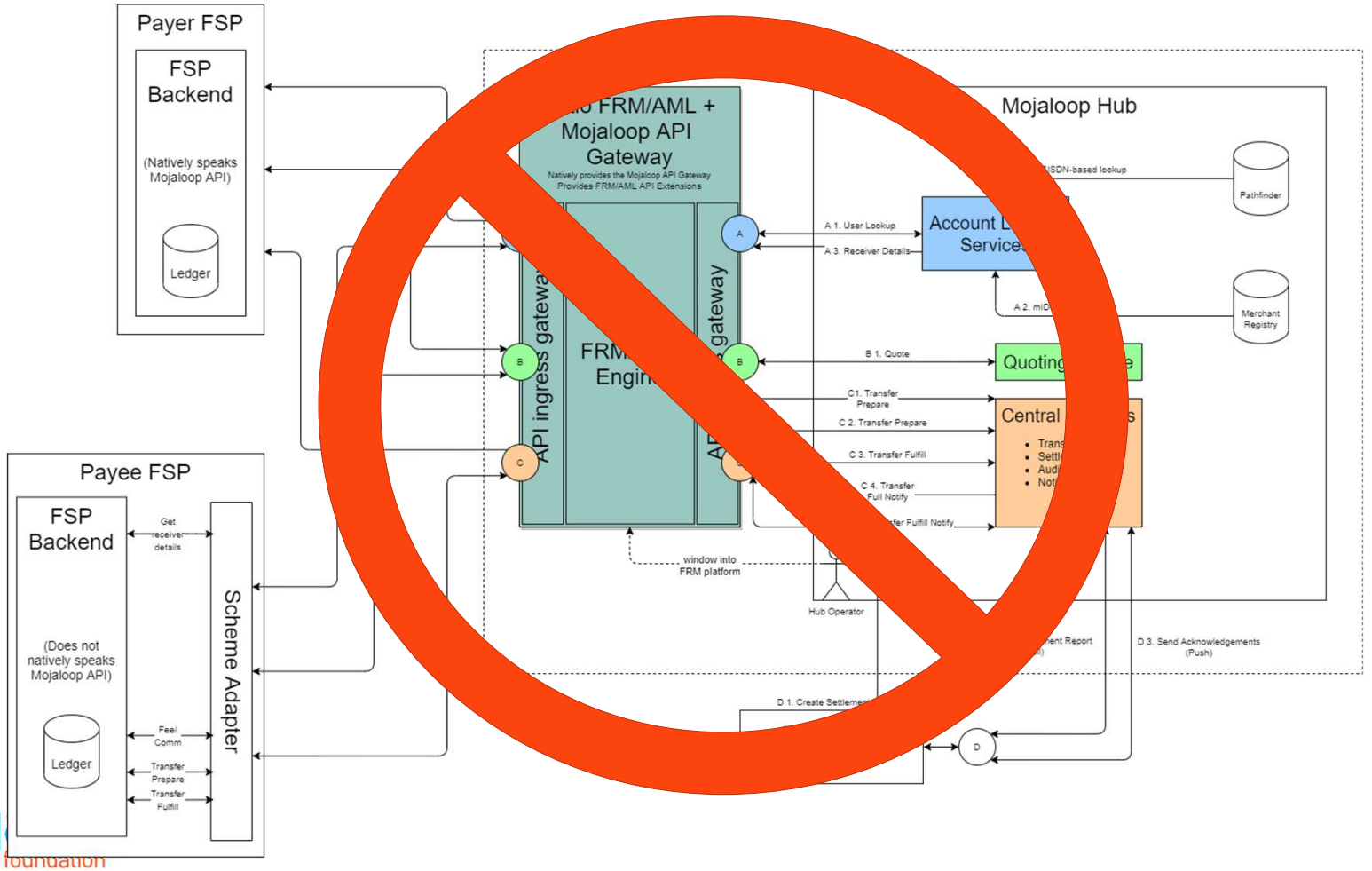
The integration architecture



The integration architecture

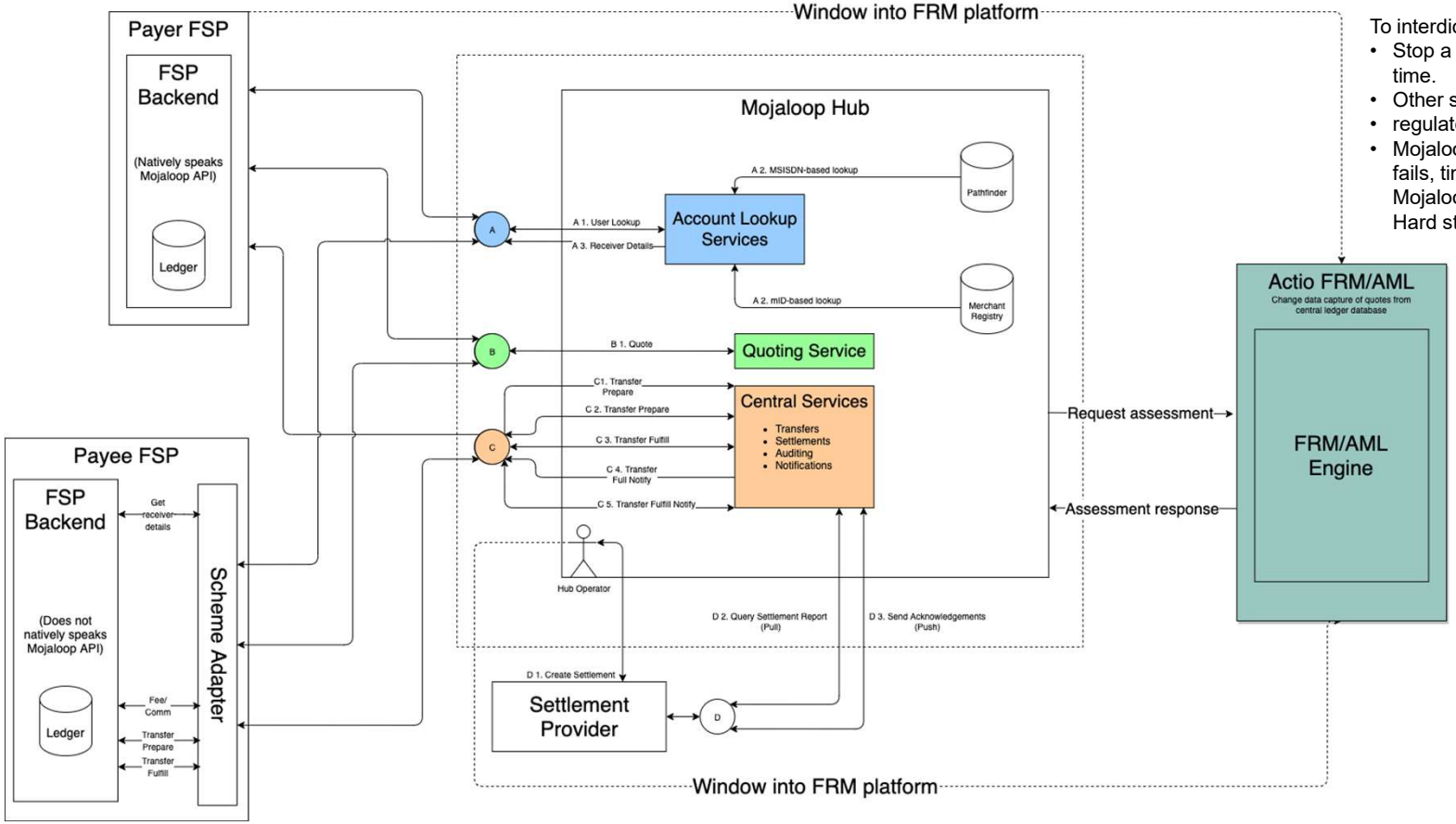


The integration architecture



Streaming model

The integration architecture



To interdict or not?

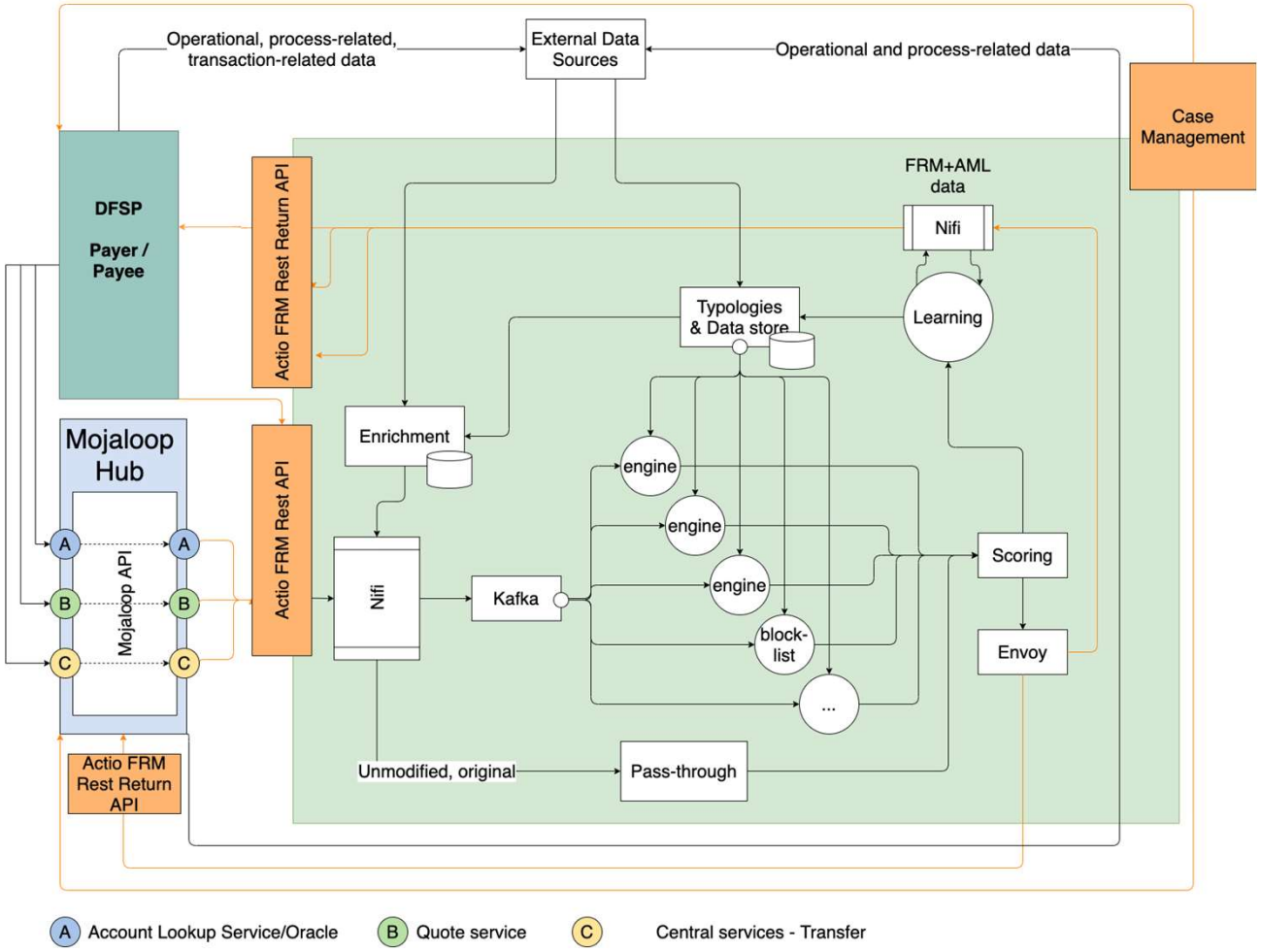
- Stop a transaction in flight real time.
- Other systems need it,
- regulators want it,
- Mojaloop's model completes or fails, times out etc. For Mojaloop do you Interdict, Hard stop?

mojaloop foundation

Service model

The integration architecture

Interface Points in Orange

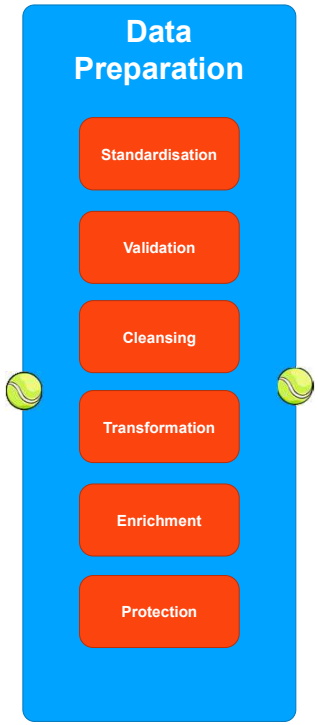


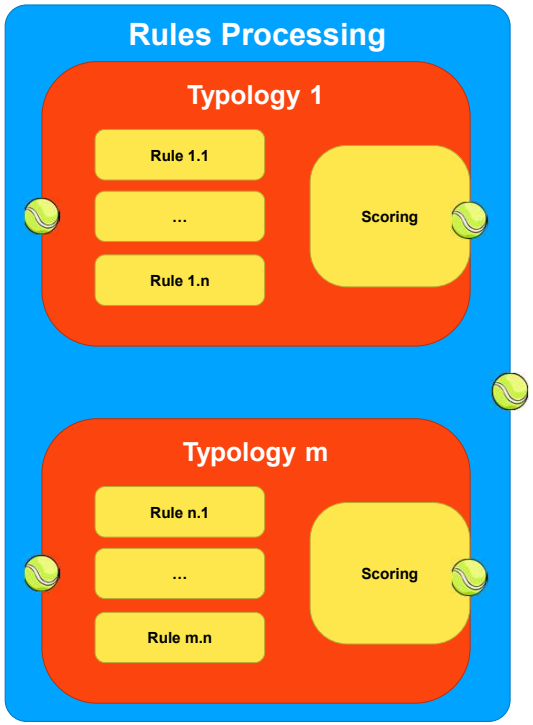
The integration architecture comparison

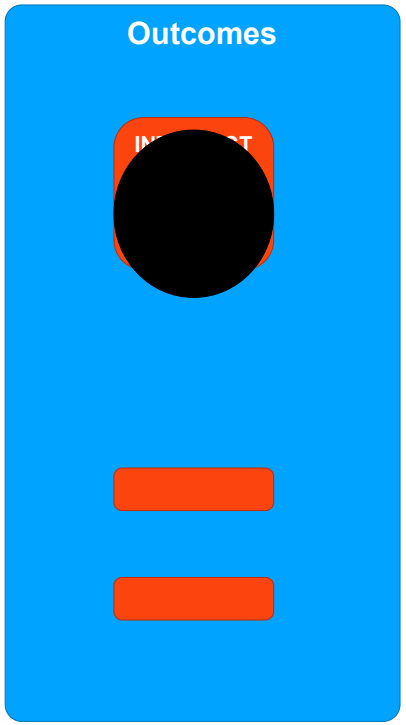
Proxy model	Gateway model	Streaming model	Service model
+ Passive - minimal friction in implementation. + FRM is no longer in critical path of transac… Mojaloop. + D… ot require the …plem…tation of a potential …cuit …reaker.	+ Drop-in replacement for existing component with no external functionality changes + Can be… …a pure pa… …ough fashion so… …t the M… …op software can be i… …alled …thou… …nabling the FRM so… …n, whilst st… …talling the FRM solution fo… …r use	+ A relatively simple implementation approach as it requires minimal integration between ACTIO, DFSPs and Mojaloop. + Ability to ingest all historical data	+ Decoupled, and works with any system via API + Supports the previously agreed to preferred method of semi-detached. + Can support interdiction + Historical data passed to us and we subscribe to data sources via ETL methods.
- Effectively a ma… …n-the-middle …ttack architecture… …ompromis… …e proxy would allow… …nsactions being hand… …e Hu… …perator to be interce… …d, chang… …eplayed or… …d. - Maintenance … …be more difficult over time - Requires both Mojaloop and DFSPs to agree to use the proxy, which makes uptake more difficult and increases friction in the sales and implementation process.	- Potentially mo… …mplex to maintain the ACT… …RM solut… …s a gateway - R… …ires providing alter… …Helm char… …a Mojaloo… implementat… - As with any gateway, if compromised could act as a man-in-the-middle attack vector.	- Does not allow for interdiction. The FRM solution therefore becomes a passive solution only generating information for potential follow-up. - Requires Hub operator to install additional software and potentially additional hardware within their implementation of Mojaloop, and configure the FRM solution appropriately.	- Requires DFSP to install additional software and potentially additional hardware within their implementation of Mojaloop, and configure the FRM solution appropriately. - Works best with a service provider or other central system of shared data.

The FRM concept

… the whole thing can be built in Open Source Software!

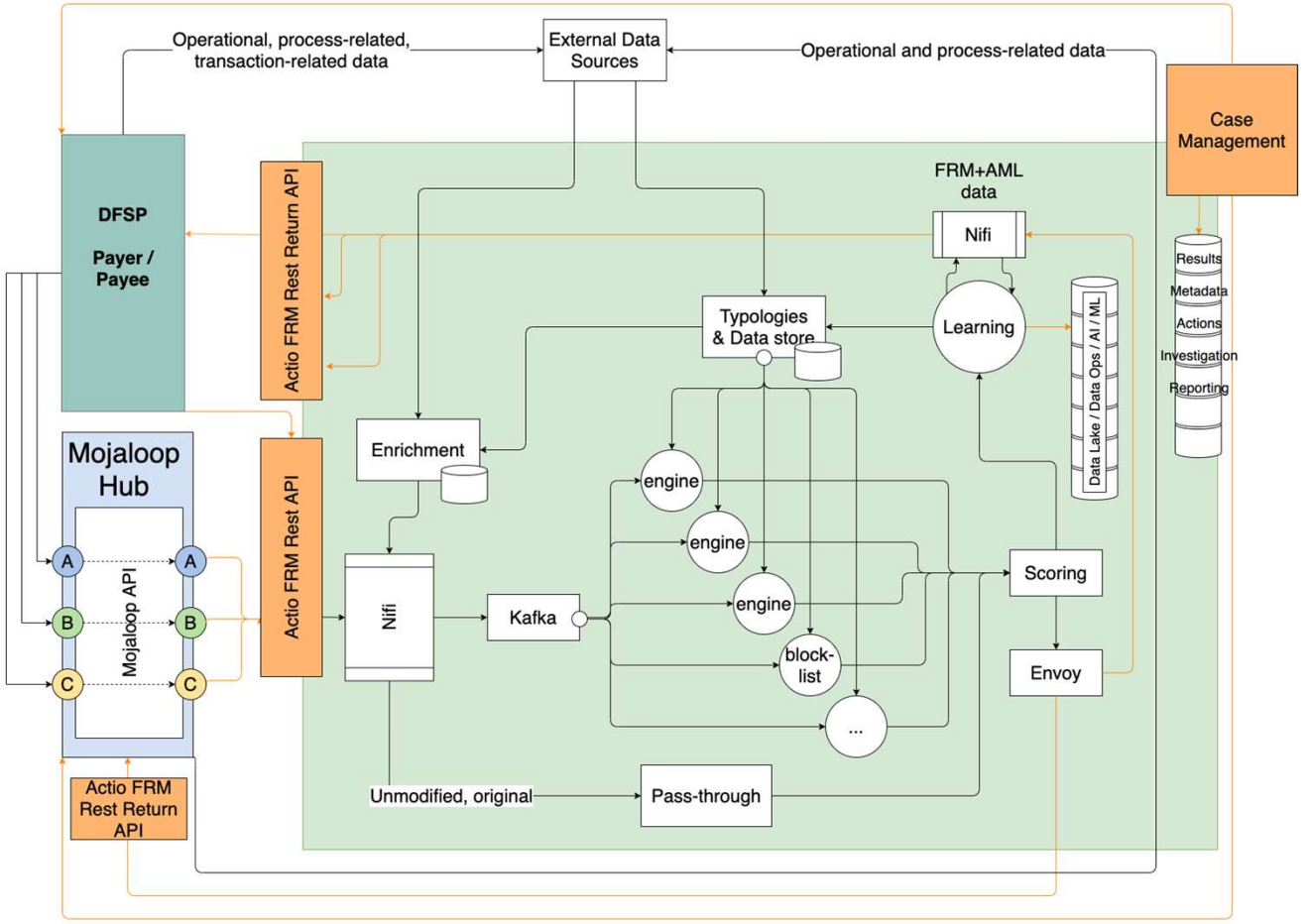




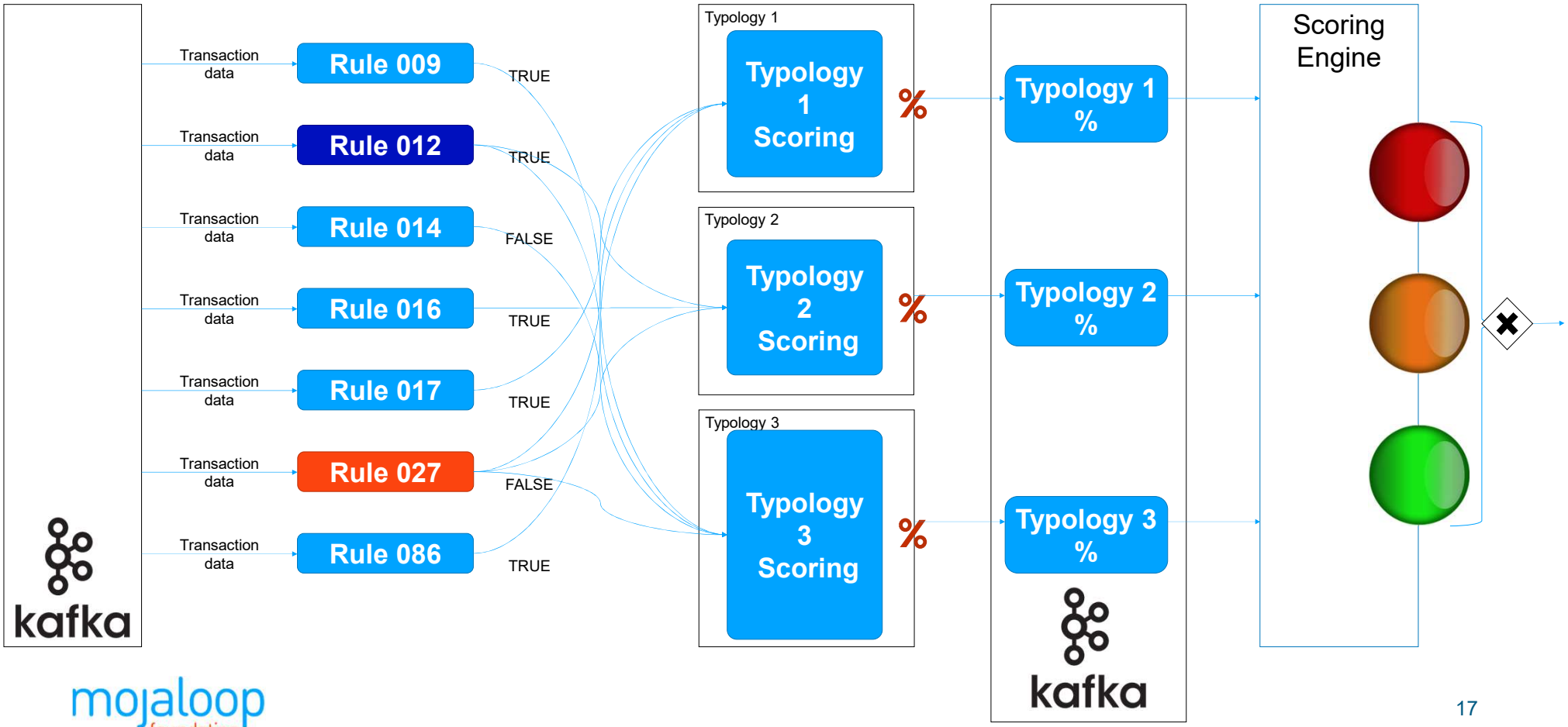


mojaloop foundation

The internal architecture

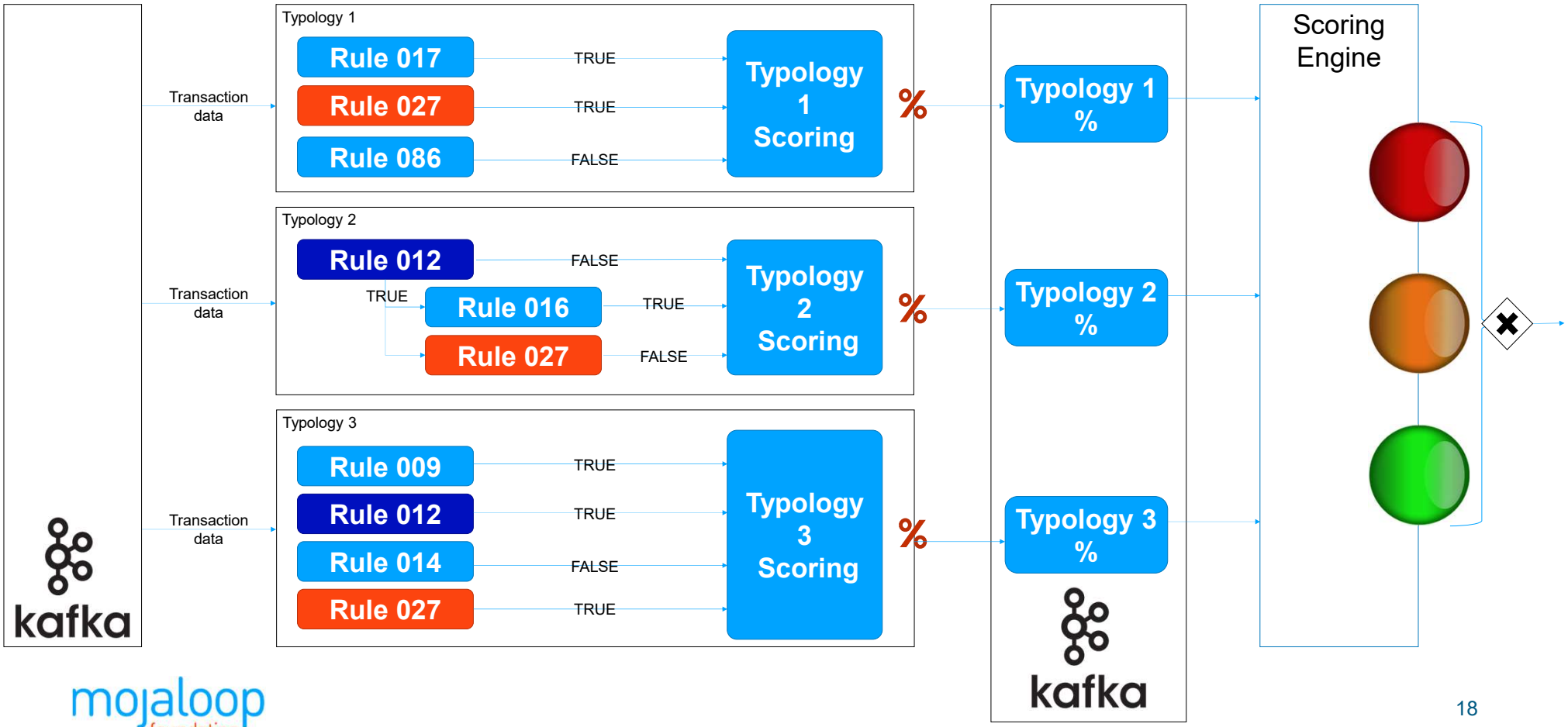


The pipeline-based approach



mojaloop foundation

The typology-based approach



mojaloop foundation

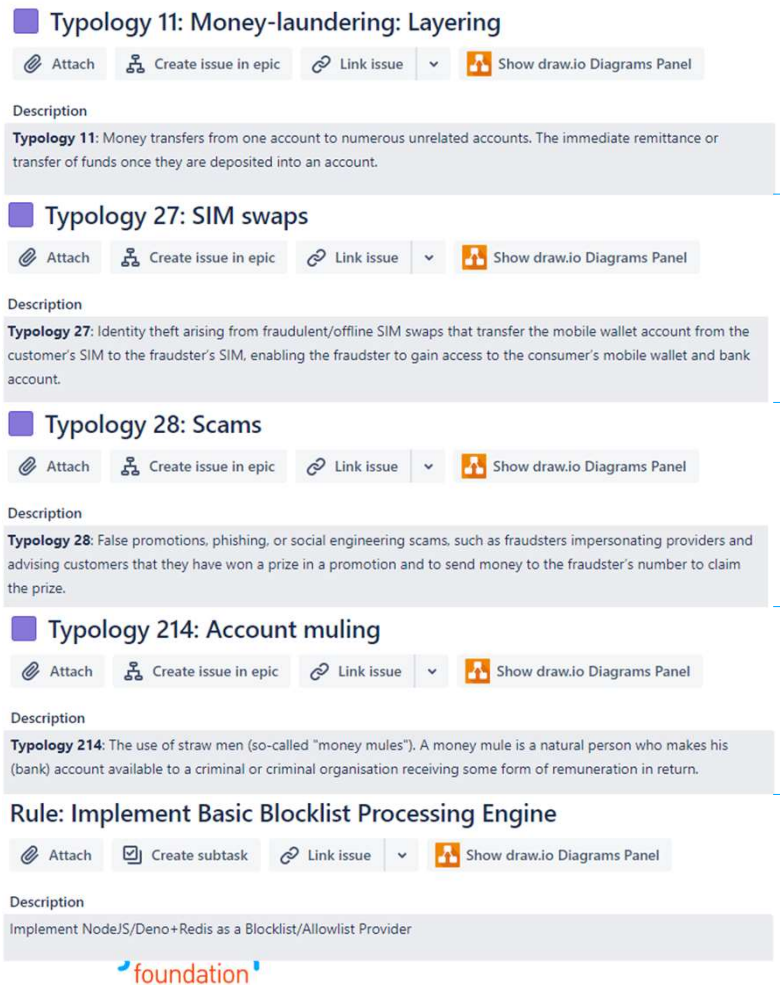
The typology processing and scoring

- Two approaches for processing and scoring typologies were developed. The intent was to deliver one approach during the POC and complete the assessment or the other early in the MVP.

	Pipeline-based approach A design pattern used in large scale MQ ODM Systems	**Typology-based approach** Processing a complete typology as a microflow
Upside:	+ Each Rule Fires Once – Efficient + Normalized Structure Supports Maintainability	+ Easier to build + Denormalized, potential to be very fast + Less computational complexity + Easier individual rule parameterisation
Downside:	- Complex to Build - Orchestration More Involved - Complexity requires a design that can handle the threading and parallelism - Rule parameterisation by typology results in rule duplication	- Runs process engine multiple times, once per typology that needs it - At scale, additional resources may be problematic – unknown and needs further testing - Requires extra work on the admin and maintenance side
Decision:	• Moved to MVP due to complexity and to allow for more time to choose the best fic technologically	• Built first, as it is supported by current infrastructure • Evaluate performance vs scalability

TYPOLOGIES AND RULES

The fraud risk typologies



Rule 017 Rule 086
Rule 027 Rule 087

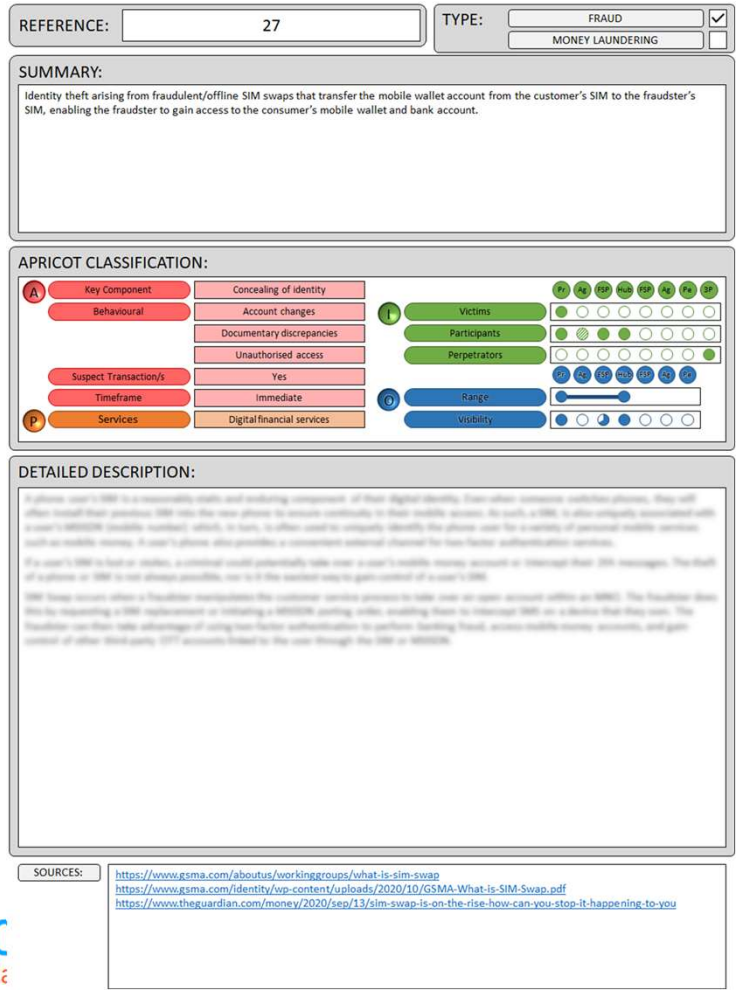
Rule 003 Rule 016 Rule 030
Rule 012 Rule 027 Rule 063

Rule 009 Rule 014 Rule 030 Rule 048
Rule 012 Rule 018 Rule 032 Rule 078

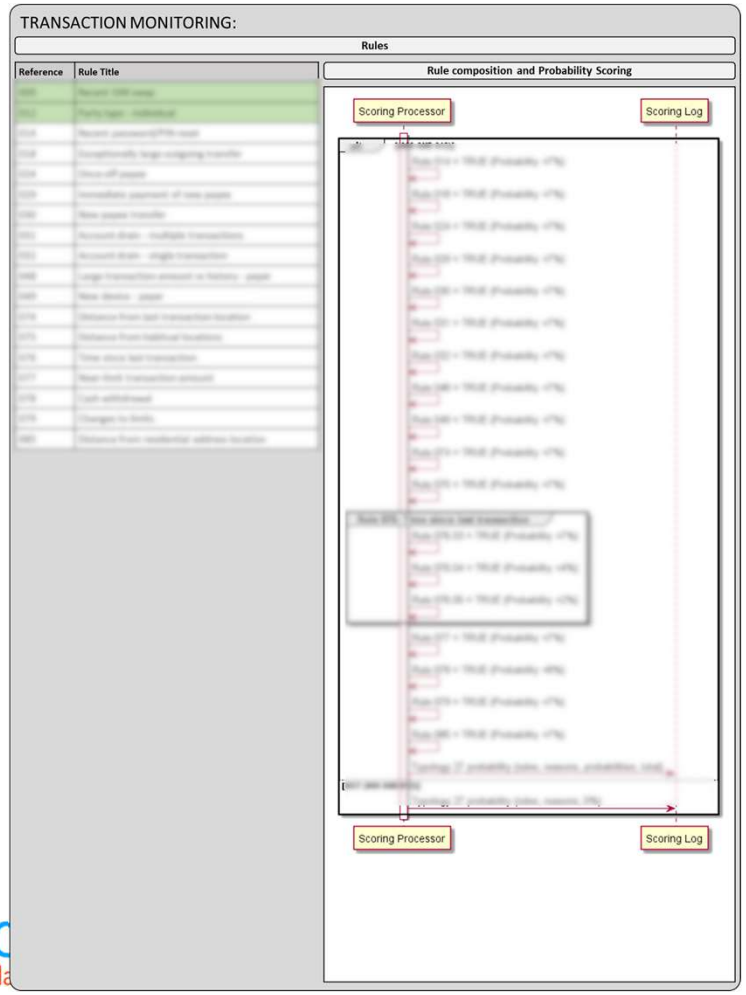
Rule 003 Rule 027 Rule 048
Rule 012 Rule 030 Rule 078

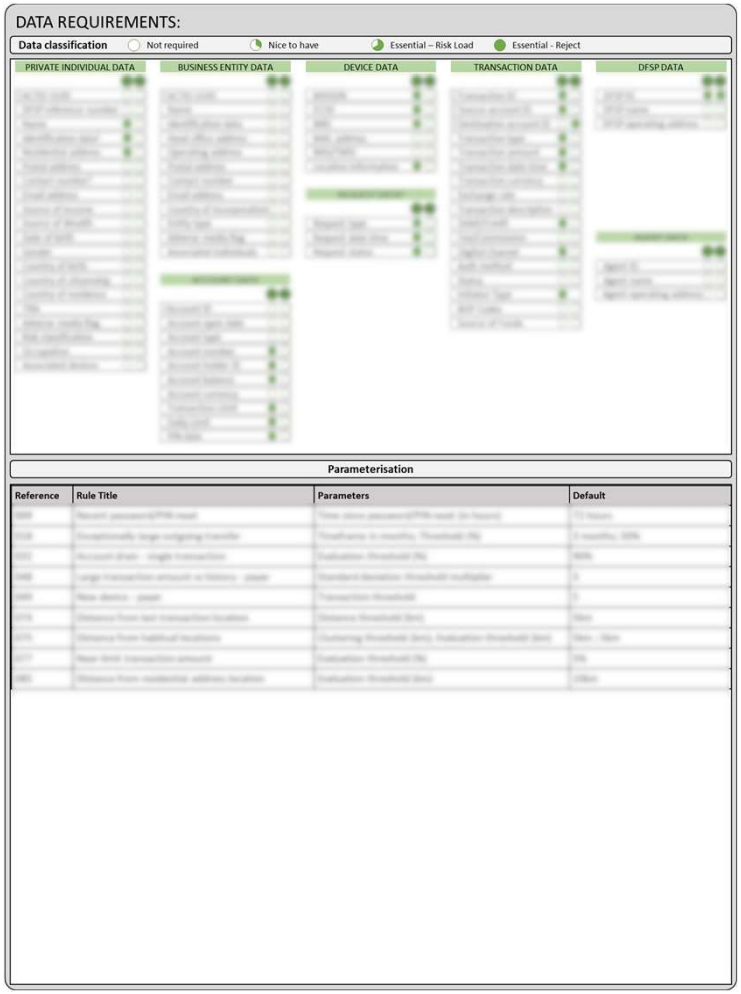
Block rule

The APRICOT model, part IV



The APRICOT model, part IV





The data factory

We need to hide our Needle in a LOT of hay

- Fraud and Typology Management see duplicate transactions as either:
  - A common pattern to be ignored
  - A suspicious pattern to be highlighted

- This creates a challenge - You run out of unique data very quickly at 10K TPS
  - 600K Transactions per minute
  - 36M Transactions per hour
  - 864M Transactions per day

mojaloop foundation

DEMO

Elastic

Search Elastic

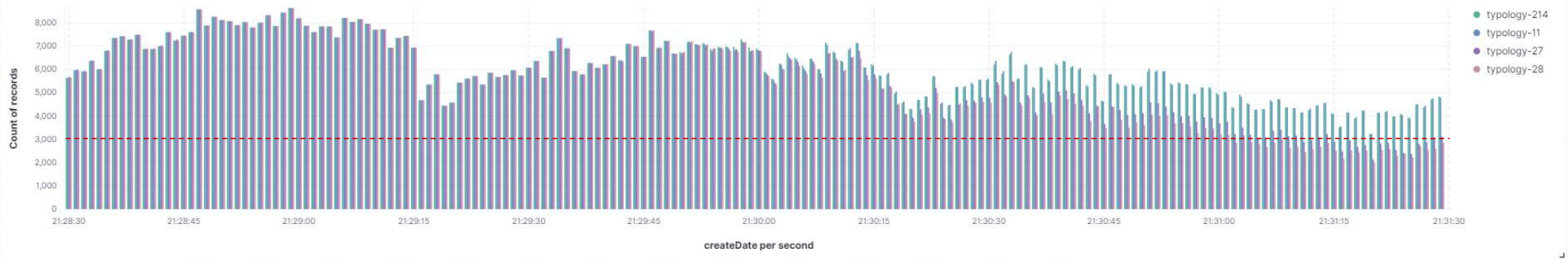
Dashboard / Editing Loadtest (unsaved)

Options Share Add Cancel Save Create new

Search KQL Jan 25, 2021 @ 21:28:30.00 → Jan 25, 2021 @ 21:31:30.00 Refresh

+ Add filter

demo day



avg tps

avg tps	avg tps	avg tps	avg tps
6,094.289	6,088.689	5,602.733	5,475.761
typology-214	typology-11	typology-27	typology-28

transaction count

transaction count	transaction count	transaction count	transaction count
1,096,972	1,095,964	1,008,492	985,637
typology-214	typology-11	typology-27	typology-28

Improving performance

- Optimising the java-based components
- Re-evaluate the Kubernetes auto-scaler
- Define our telemetry/application performance management strategy
- To JSON or not to JSON
- Message compression

Next steps

- Things we're addressing in the next sprint
  - A graph-capable rules engine
  - A performant historical data store
  - Rules configuration management and change control
  - Typology and rules orchestration
  - Improving performance
- Realistic data synthesis on a large scale
  - Simulate fraud without putting it there ourselves
  - Rule/Typology calibration
- Security and privacy
- Increased community participation
  - Secure onboarding and sharing of work and information