mojaloop foundation

Mifos Updates

Building an Entire Digital Financial Services Ecosystem on Open Source

Edward Cable, The Mifos Initiative
Istvan Molnar, DPC Consulting

Agenda

- Who We Are
- Community Need
- Our Assets & Value We Bring
- Why Open Matters
- Go Farther Together
- Demo & Technical Updates
  - Performance Testing
  - ISO 20022 to Mojaloop
  - P2P Transfer & Zeebe Modeler
  - DFSP Operational Control Center

Mifos Initiative & DPC

- 501(c)3 non-profit guiding Mifos OS community advancing Apache Fineract
- Stewards of roadmap & collaborative center
- Industry thought leader & HFOSS pioneer since 2006
- Guide Ecosystem of Solutions & Network of Partners
- 20 million clients reached across 400 orgs

- 20+ years in IT training, consultancy, software development
- Experience with Instant Payment Systems (Singapore FAST, Hungary HCT Inst, SEPA Instant)
  - Including clearing house solutions, payment hubs, shadow balance solutions from key vendors
  - Developing central clearing house prototype, simulator for participants, payment hub

Edward Cable

mojaloop foundation

Mifos Initiative

FINANCIAL INNOVATION AWARDS 2019 — Highly Commended — Best financial inclusion or outreach initiative — Presented by The London Institute of Banking & Finance

GLOBAL FINTECH AWARDS 2020 BENZINGA 2020 LISTMAKER

István Molnár

Kristóf Józsa

Ádám Sághy

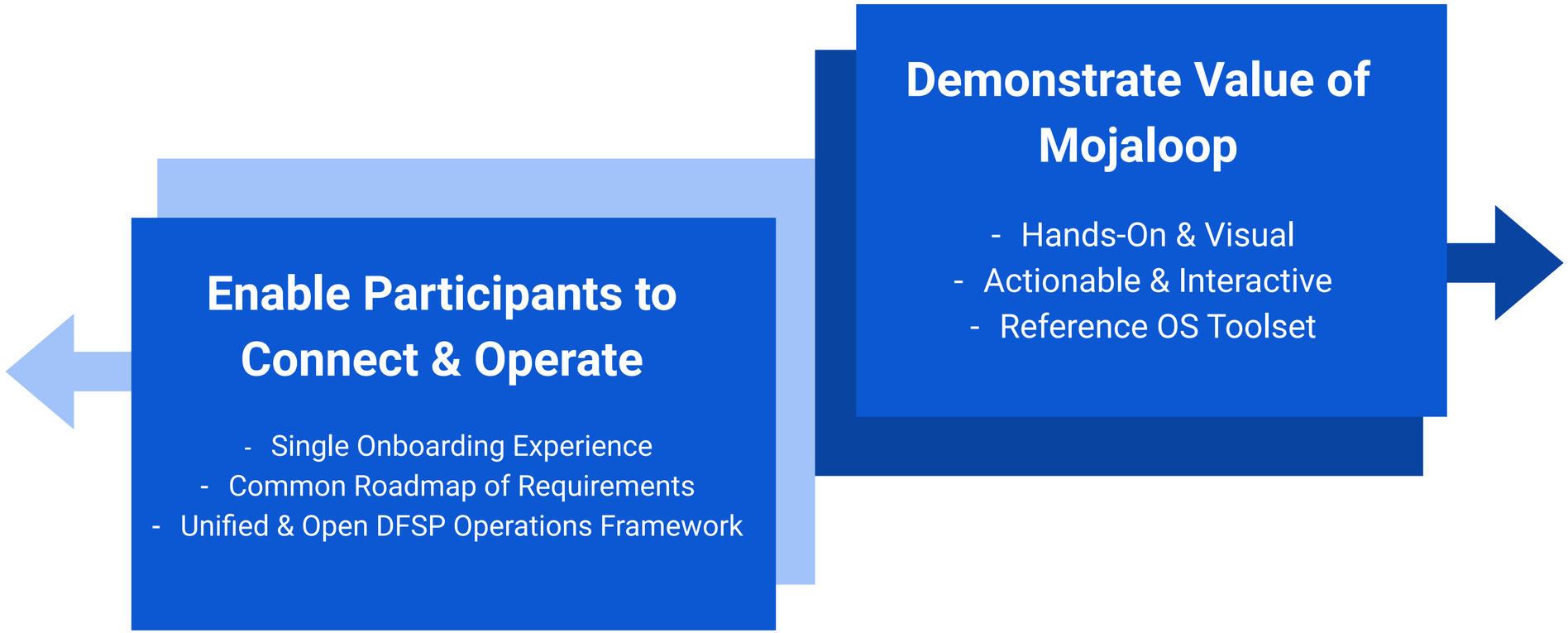
dpc consulting

Zoltán Nébli

Zoltán Mezei

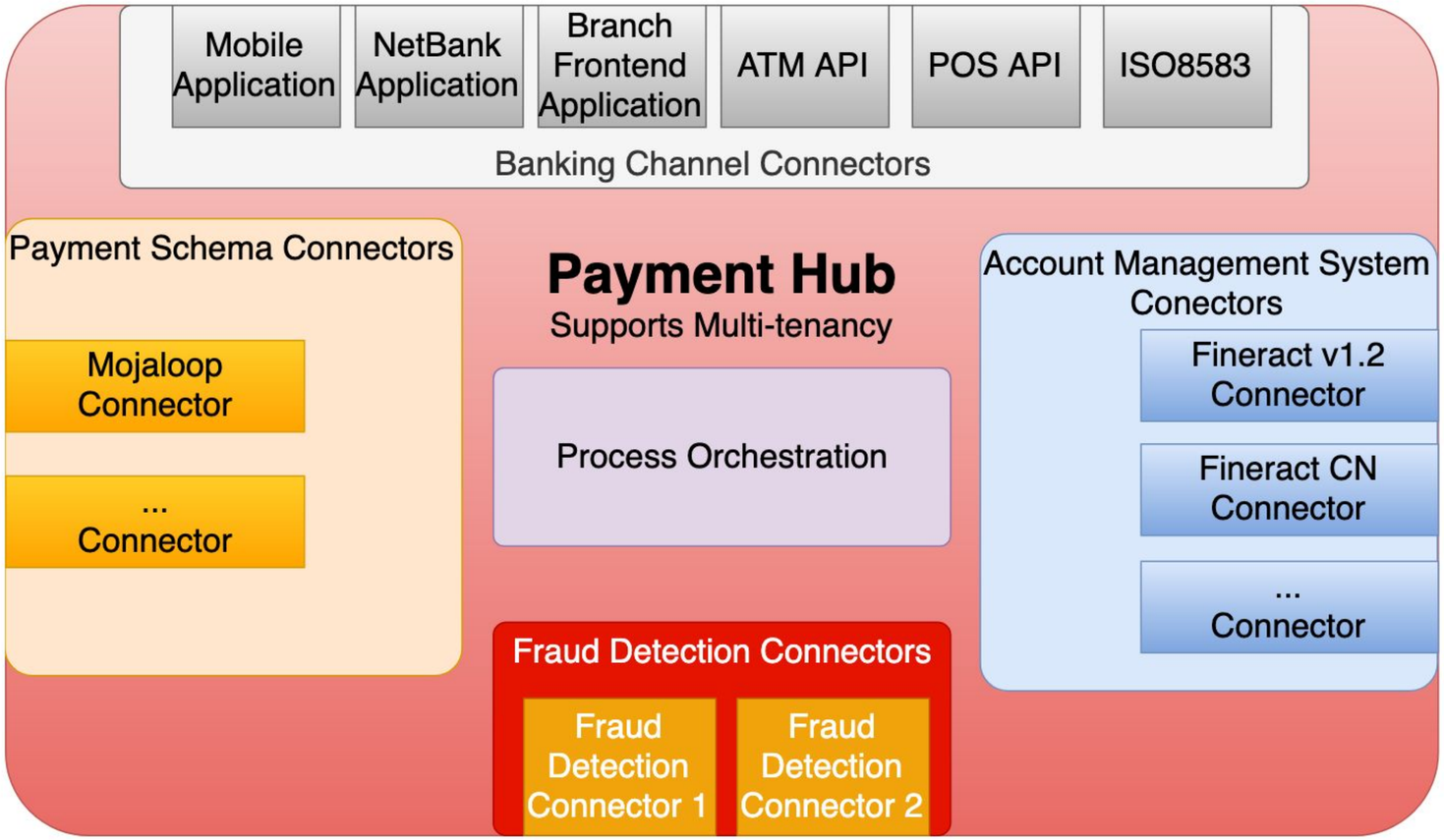
Community Need

Accelerate Adoption of Mojaloop



mojaloop foundation

Payment Hub EE

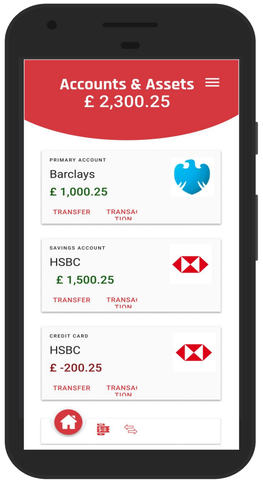


Enable Participants to Connect & Operate

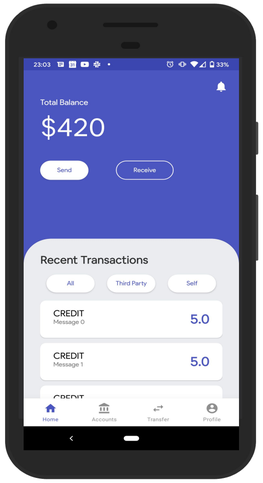
- Single Onboarding Experience
- Common Roadmap of Requirements
- Unified & Open DFSP Operations Framework

mojaloop foundation

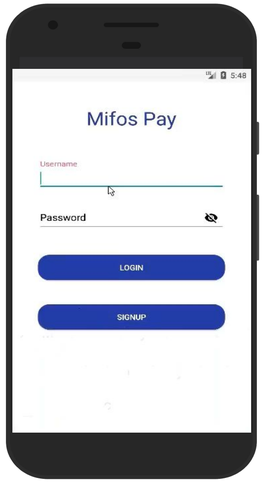
Interactive & Immersive Lab Environment



Open Banking App



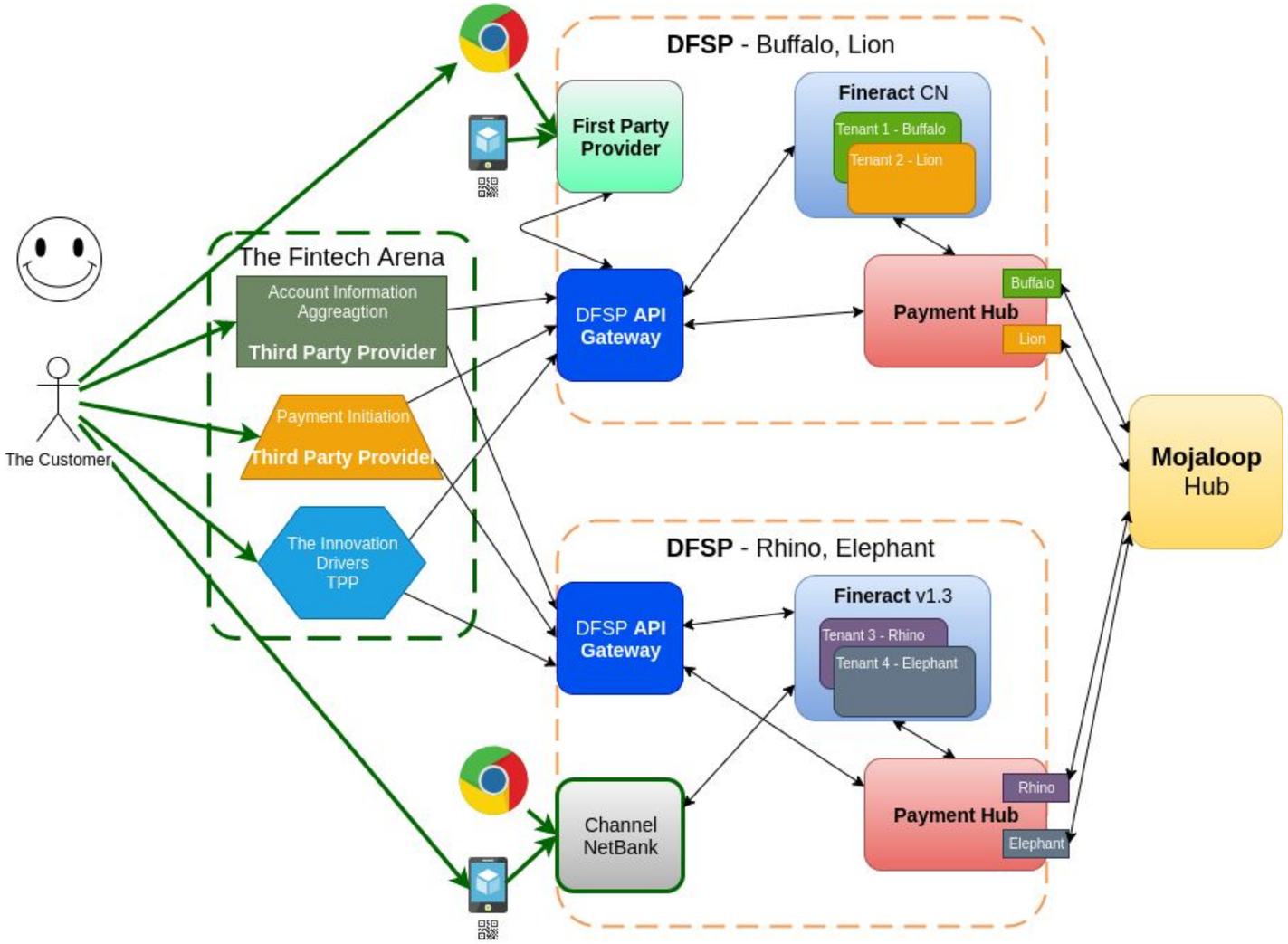
Mobile Banking



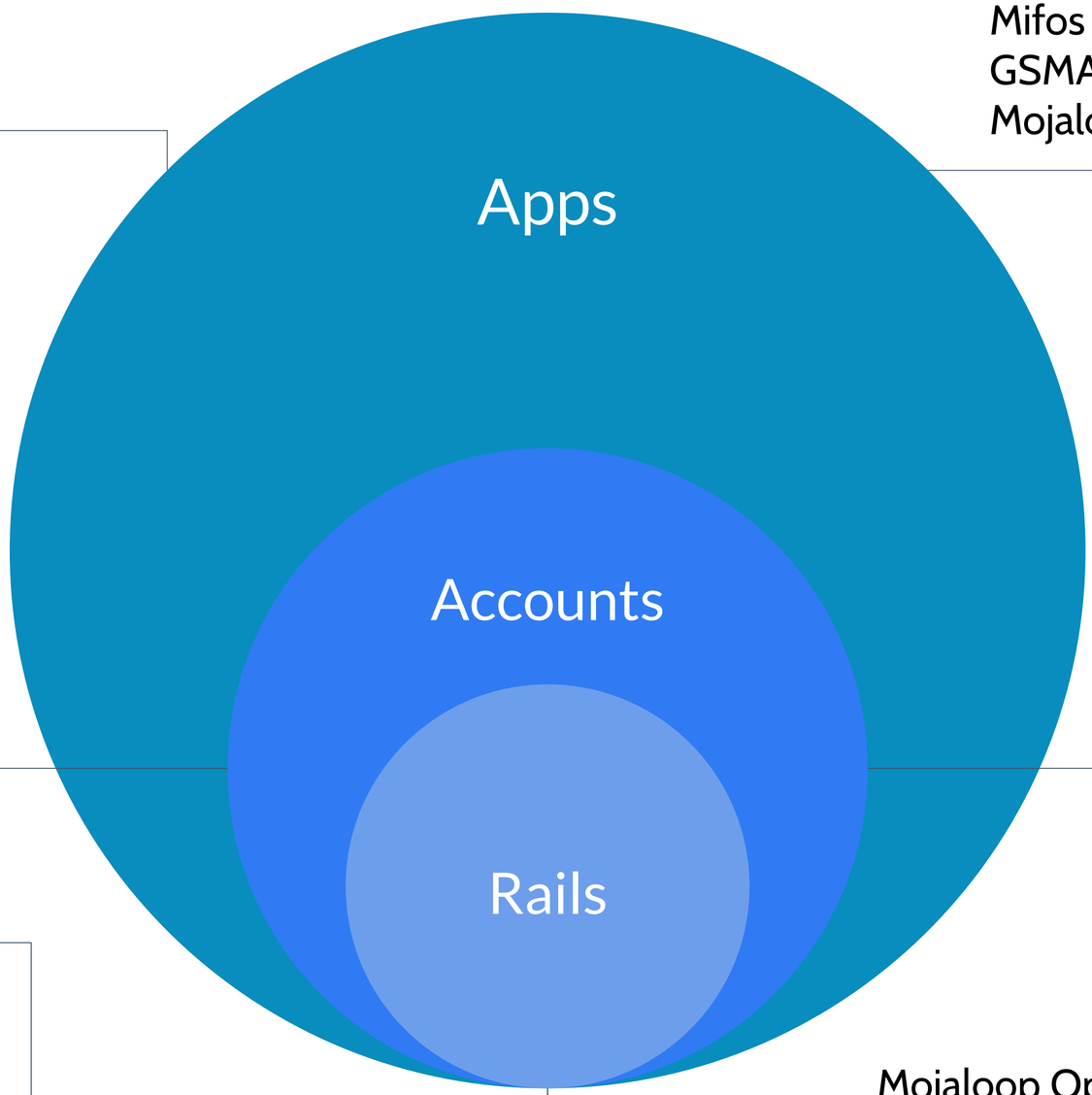
Mobile Wallet

Demonstrate Value of Mojaloop

- Hands-On & Visual
- Actionable & Interactive
- Reference OS Toolset



Long Tail of Adoption and Innovation



Fintechs/PISPs

Mifos Open Banking API
GSMA MM API
Mojaloop PISP API

DFSPs

Mifos/Fineract
OS Core Banking API

Switch Operator

Mojaloop Open API

mojaloop foundation

Why Open Matters

Building an Entire Digital Financial Services Ecosystem on Open Source

mojaloop foundation

Digital Public Goods

Financial Inclusion Community of Practice

United Nations

mojaloop

MOSIP

Mifos

Apache Fineract

OpenG2P

X-ROAD

OPEN CRVS

- **Advance the Secretary General's digital cooperation roadmap to achieve greater financial inclusion and meet the SDGs by 2030.**
- **Financial Inclusion CoP spent the last few months identifying & shortlisting technologies that, in a given country, can be used by a range of service providers and innovators**
- **Technologies can be built on across sectors and have features that allow countries to freely adopt & iterate them to meet local needs**

Digital Public Goods are defined by the UN Secretary-General's Roadmap for Digital Cooperation as "*open-source software, open data, open artificial intelligence models, open standards and open content that adhere to privacy and other applicable international and domestic laws, standards and best practices and do no harm.*"

Digital Public Infrastructure (DPI): *the rails that other solutions "run on top of"* and their implementation typically enables many other solutions and business models to flourish.

End to End Open Source Stack for Digital Financial Services

- Open Stack
  - OS L1-Aligned Payment Switch - Mojaloop
  - OS Bridge - Payment Hub
  - OS Account Management System - Mifos/Fineract
  - OS Reference Mobile Apps - Mobile Banking, Mobile Wallet

THE LEVEL ONE PROJECT

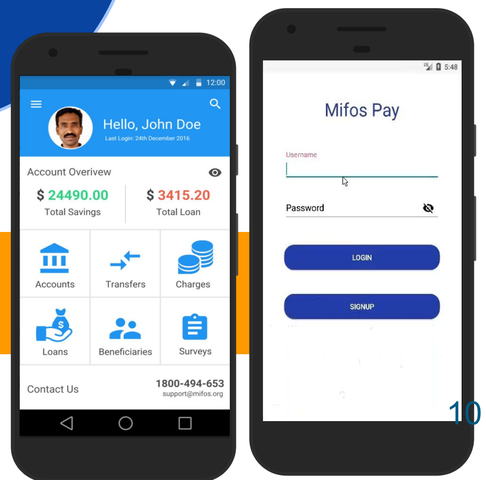
mojaloop foundation

mojaloop

Payment Hub

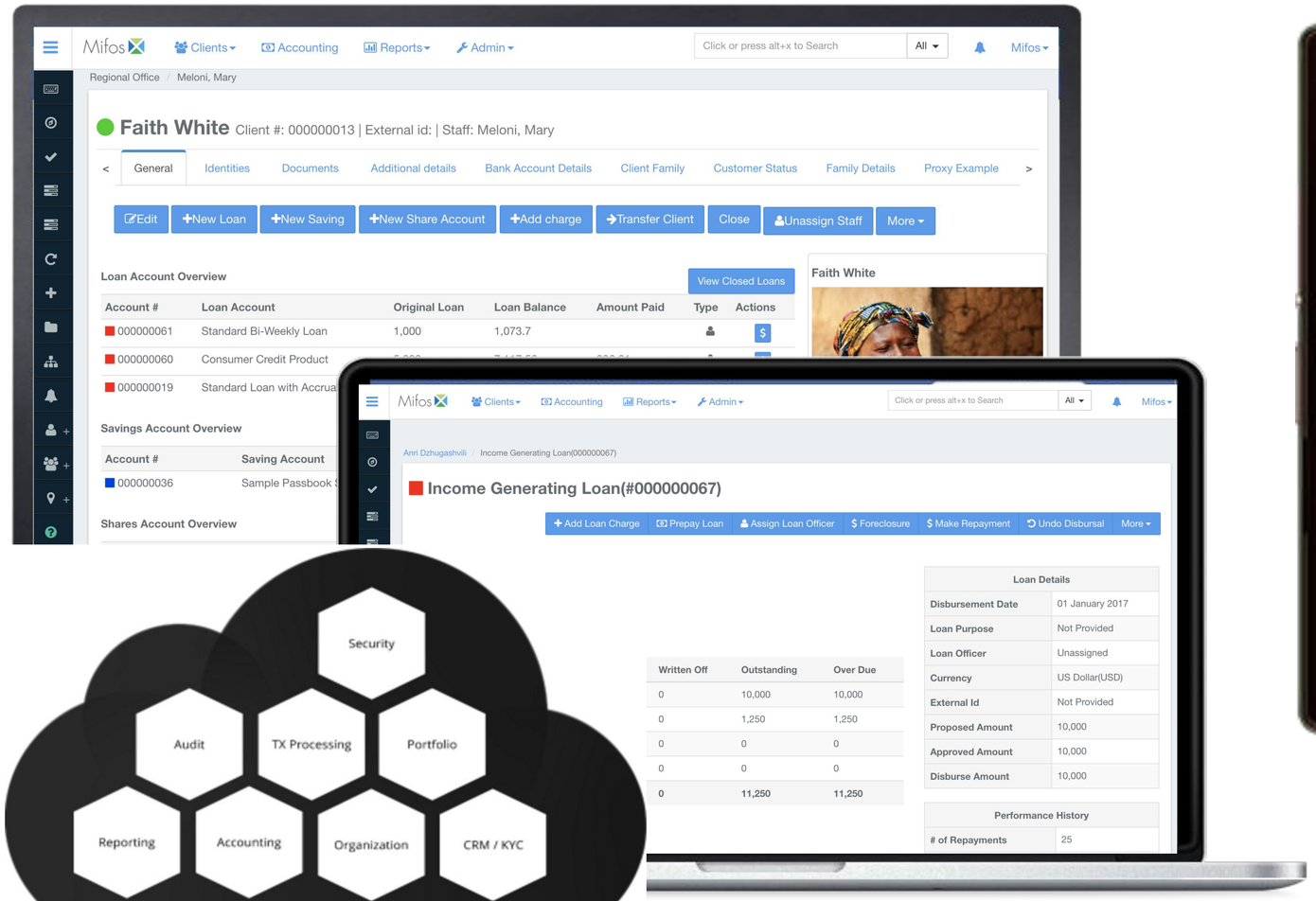
Mifos®

Open Banking APIs



Open Technology Stack for DFS Innovation

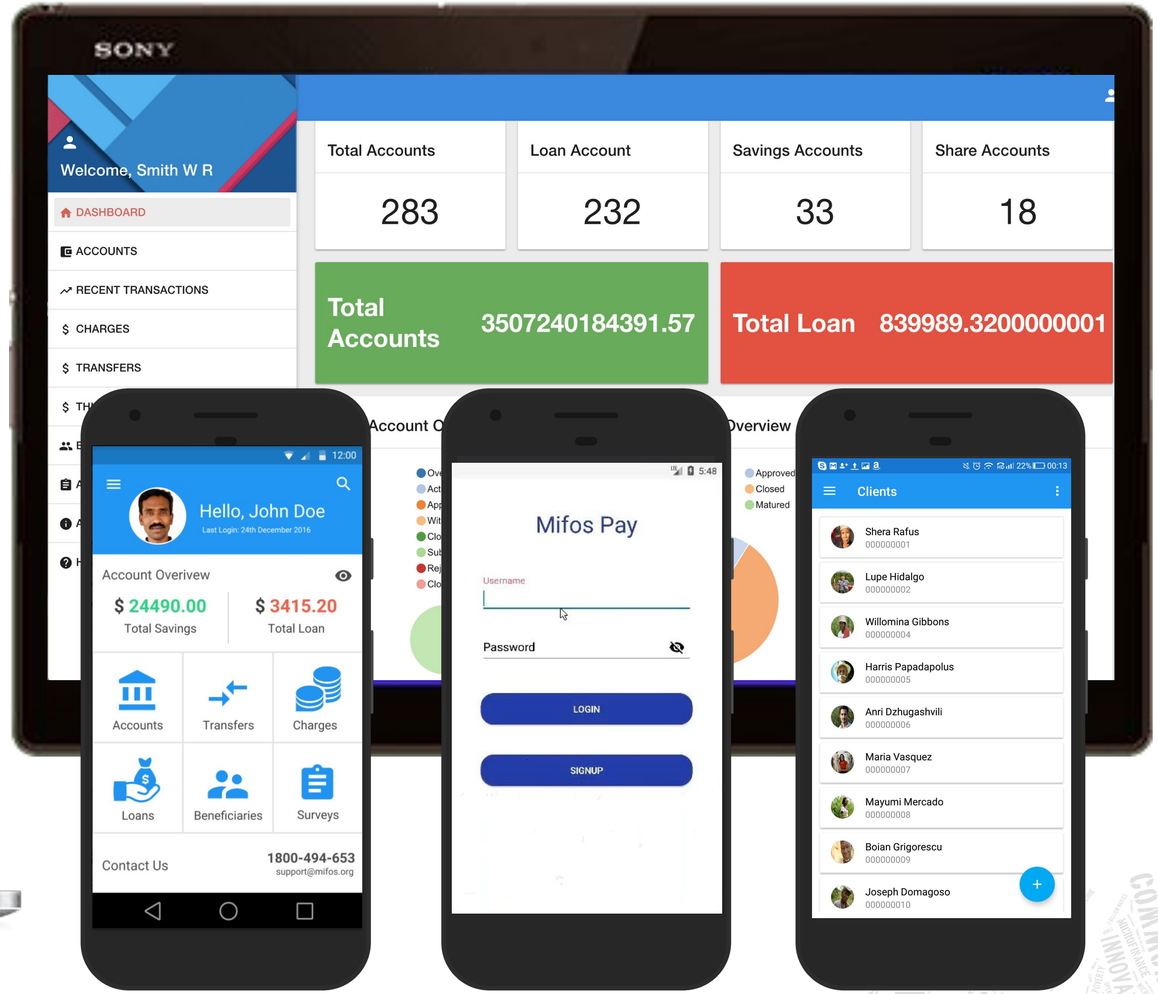
Platform & APIs



Application Framework

Web App

Online Banking App



Mobile Banking

Mobile Wallet

Mobile Field Ops

Mifos

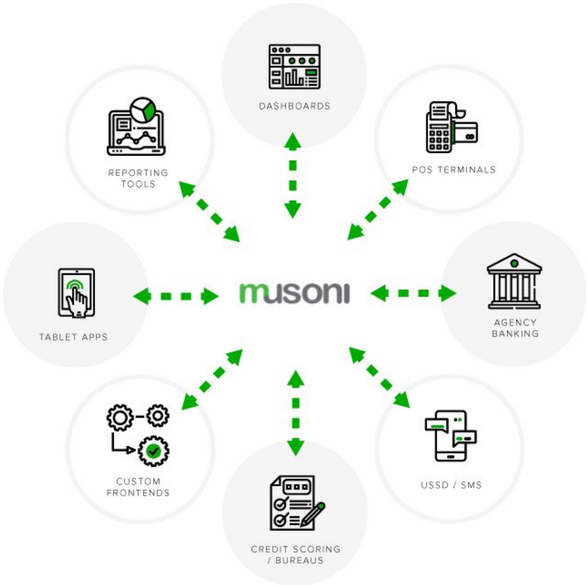
@mifos

11

Cloud Core Banking Powered by Fineract

musoni

1,453,293 clients across 75 customers in 15 countries



FINFLUX
BANKING MADE EASY

2.7 million clients across 50 customers in India & beyond.

kotak Kotak Mahindra Bank
KEB Hana Bank
MONEYTAP
Chaitanya
Money View
CREDIT PAYMENTS
Light MICROFINANCE
Vaya
DIGAMBER FINANCE
Rapid Rupee
slice
Caritas NAIROBI
PHAKAMANI foundation
Sunvest
AL MAJMOUA المجموعة
Habitat for Humanity
MKOBO MICROFINANCE BANK
neev ENABLING EDUCATION
MOBILITY FINANCE

finabile

Financial inclusion loan management for India

mojaloop foundation

Intelligrow Bancsoft

pearlpay™

Digitizing rural banks across the Philippines

Ecosystem - Fintech powered by Mifos and Fineract

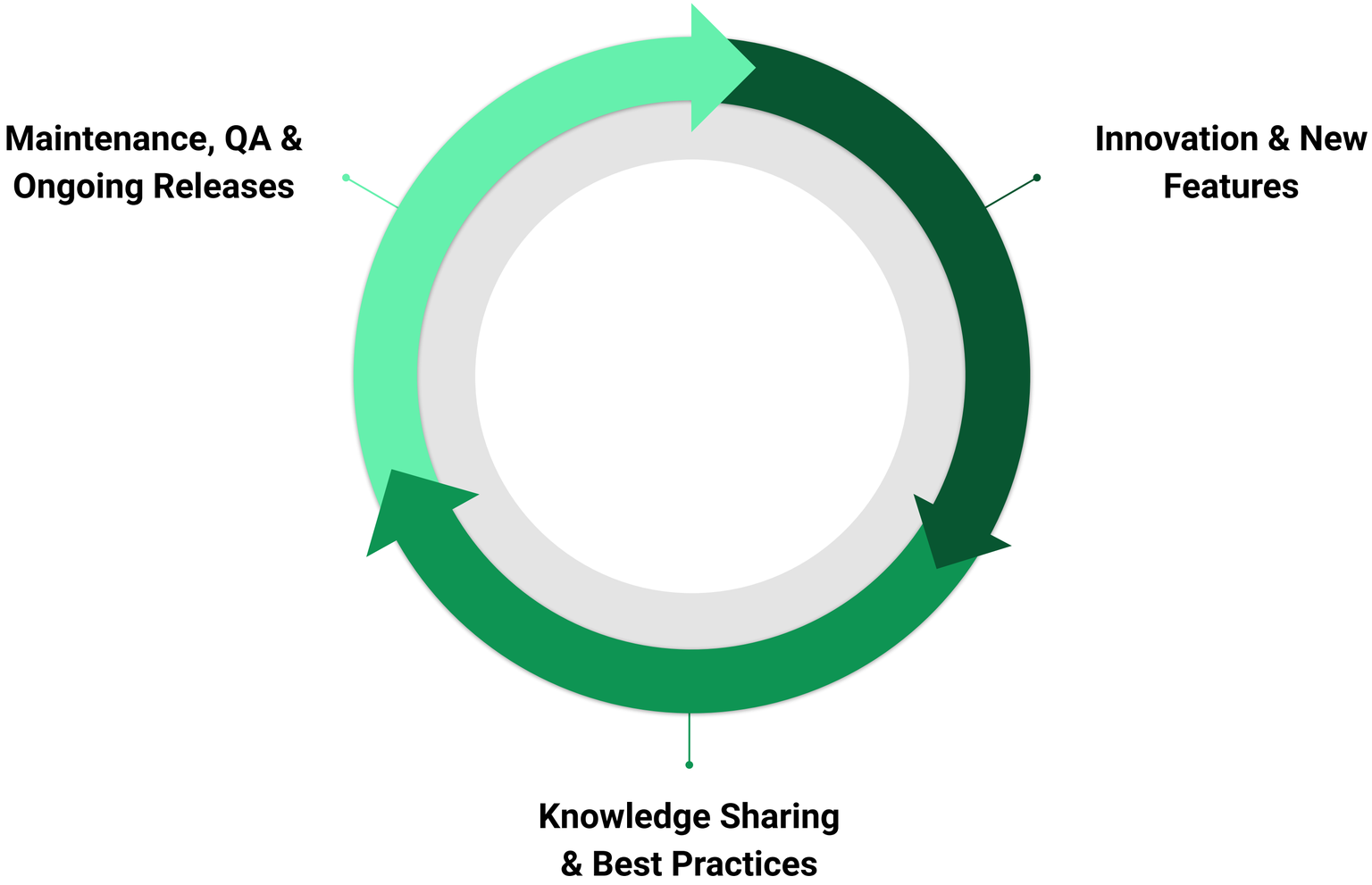
FINTECHEANDO

dpc consulting

fiter

Fynarfin

Enabling a Virtuous Cycle



“You get what you pay for, everyone gets what you pay for, and you get what everyone pays for.”

Paul Ramsey

Four Layers of APIs at 2 Different Levels

Rules

Rails

Accounts

Apps

THE LEVEL ONE PROJECT

mojaloop

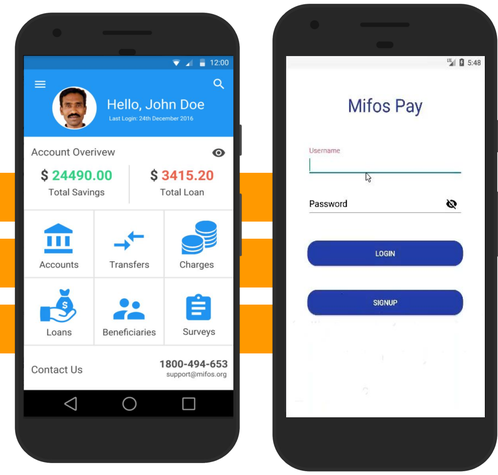
Payment Hub EE

Mifos®

Open Banking APIs

GSMA Mobile Money APIs

Mojaloop PISP APIs



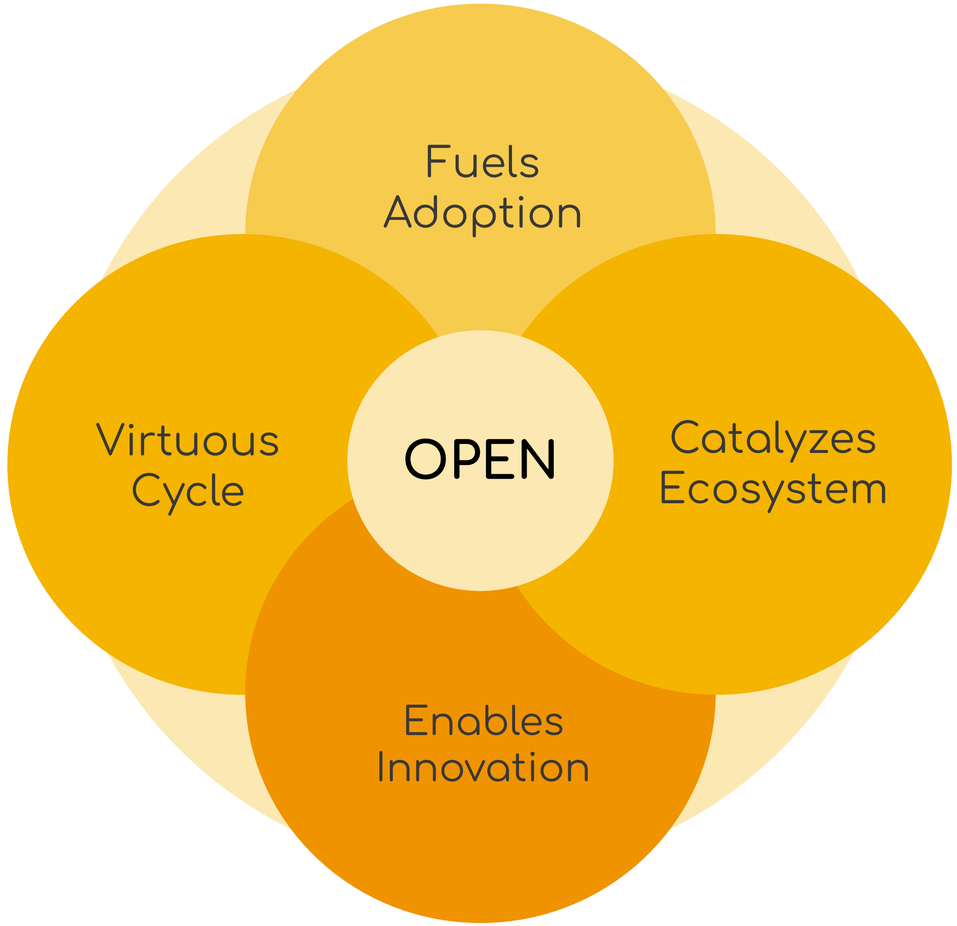
Infrastructure

Application

mojaloop foundation

Why DFSP Business Ops should be Open

- ❑ Lower Total Cost of Ownership
- ❑ Accelerated Development
- ❑ Focus On Differentiation & Value-Add
- ❑ Freedom from Vendor Lock-In
- ❑ Sustainable Business Models



mojaloop
foundation

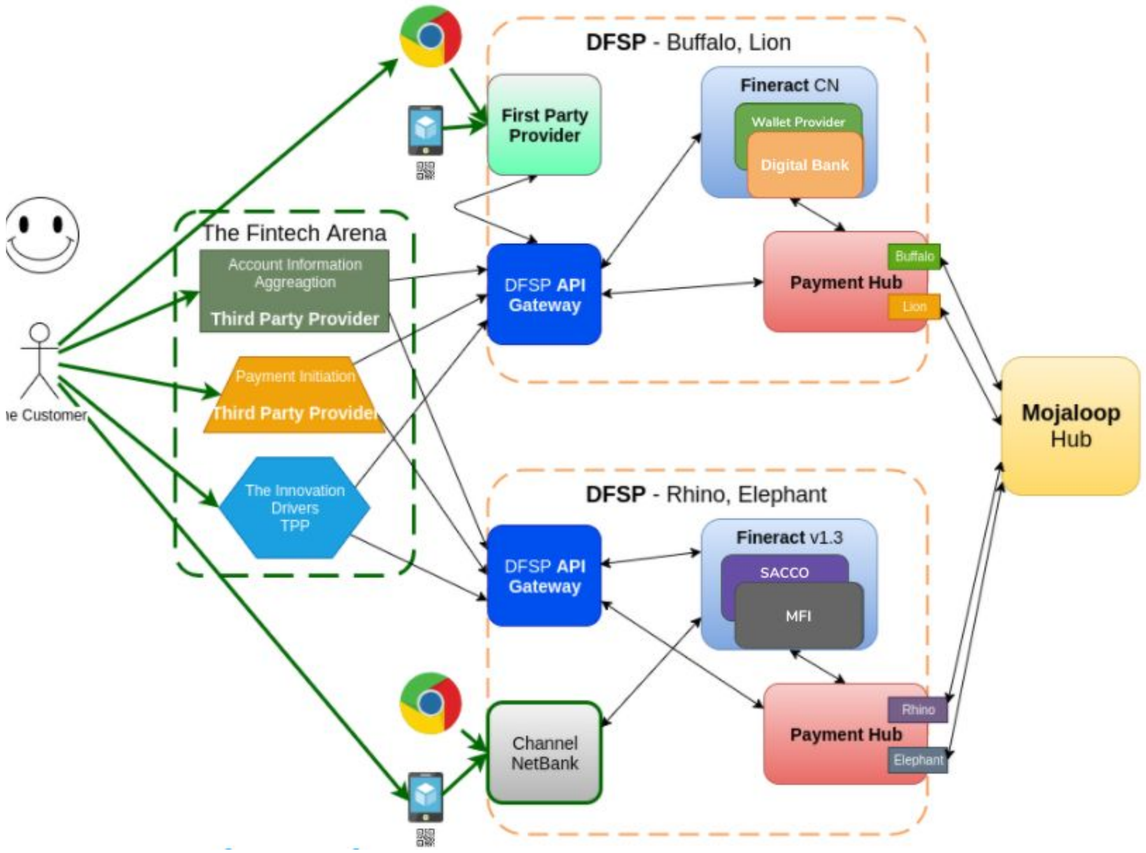
Go Farther Together

- ❏ Better collaboration at a strategic level between Mifos & Mojaloop Communities
- ❏ Leverage our collective assets to drive deployments
  - ❏ Work towards reference deployments of our digital public goods
  - ❏ Be leaders in G2P innovation
  - ❏ Provide open source tooling for regulatory sandboxes
- ❏ Address common challenges at a community and ecosystem level
  - ❏ Defining boundaries of the open core and stimulating a marketplace
  - ❏ Establishing a functional upstream contribution model from vendors on the project
  - ❏ Changing the narrative & procurement process around OS amongst governments
  - ❏ Collaborate together with FINOS to build these resources

Go Farther Together

- ❑ Connect Mifos Lab with central Mojalab Environment
- ❑ Support the Demo Working Group
- ❑ Synergize with and leverage other community efforts
- ❑ Make DFSP Operations Open
- ❑ Accelerate Existing Projects

Connect the Mifos Lab with the Mojalab



Audience	API Layer	OS Toolset
Fintechs	- **Open Banking API Layer** - **GSMA Mobile Money API (in progress)**	➢ Mobile Wallet App ➢ Mobile Banking App ➢ Online Banking App ➢ Open Banking Fintech App
DFSPs Banks, MFIs, SACCO, MMOS	**Mifos/Fineract API Layer** - Identity & KYC - Wallet/Account Management - Loan & Savings Management - Accounting & Ledger - Reporting	➢ API-Driven Open Source Core Banking Platform ➢ Angular Web UI for staff ➢ Android Mobile UI for staff
Hub Operators, Regulators	**Mojaloop API Layer** - Peer to Peer - Merchant Proximity Payment - Merchant Request to Pay - Bulk Payment/Transfer (upcoming)	➢ Payment Hub EE ○ Operations UI ○ BPMN Workflow Engine

- ❏ Provide reference DFSP UIs
- ❏ Provide reference customer apps - mobile wallet, PISP/fintech app
- ❏ Connect Fineract-based systems like Musoni via Payment Hub

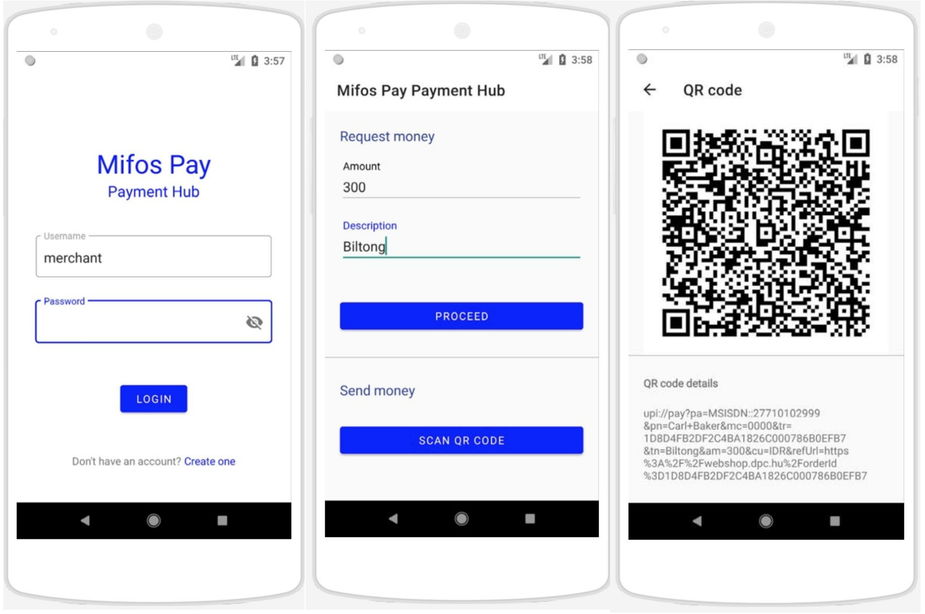
- ❏ Support future DFS Lab hackathons
- ❏ Collaborate with other DPGs like MOSIP & OpenG2P to demonstrate G2P Use Cases
- ❏ Provide open source tooling for regulatory sandbox efforts

mojaloop foundation

Support Demo Working Group

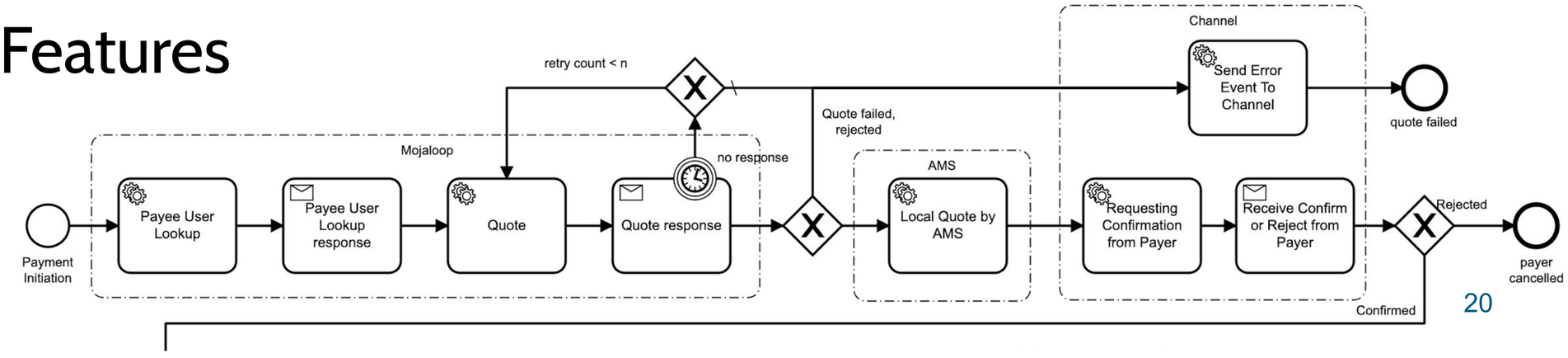
Use Cases

- ❑ P2P Transfer
- ❑ Fintech-initiated P2P Transfer via Open Banking APIs
- ❑ QR Code Payment
- ❑ Merchant Request to Pay
- ❑ Bulk Payments (future)
- ❑ PISP APIs (future)



Core Mojaloop Features

- ❑ ALS
- ❑ Quoting



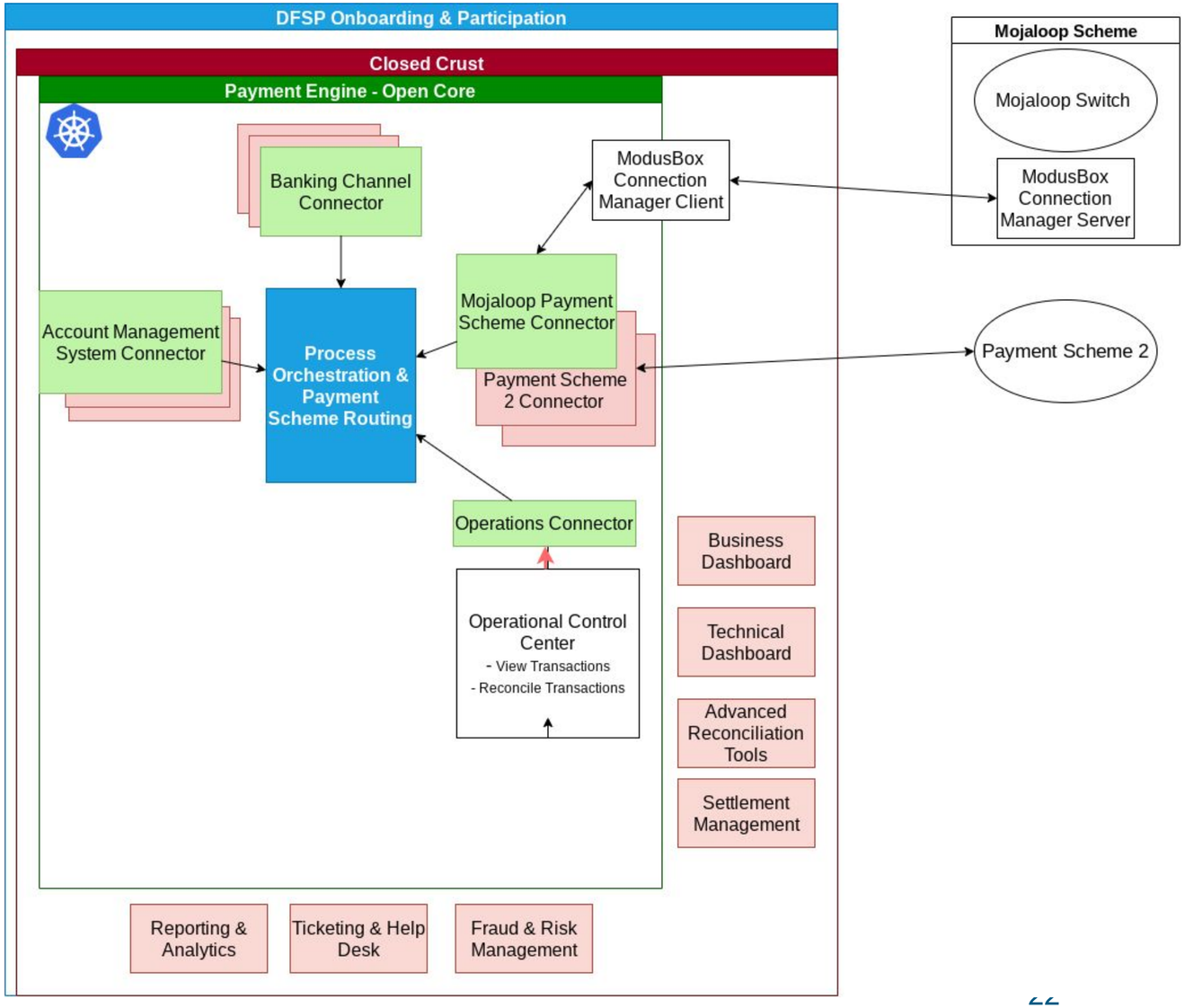
mojaloop foundation

Synergize with and Leverage other Efforts

- ❏ Incorporate Modusbox Connection Manager
- ❏ Leverage Testing Toolkit as simulator for our lab environment
- ❏ Integrate with GSMA Interoperability Test Lab
  - ❏ Already have GSMA MM API Connector
- ❏ Provide DFSP level testing for FRM service

Define Open Core for DFSP Operations

- Define common set of processes/requirements
- Establish unified approach for DFSP Onboarding & Participation
- Collaborate around a common OS payment engine architecture
- Collaborative Space vs Competitive Space
  - Define open core and closed crust where innovation and differentiation occurs
- Engage other system integrators to deploy and build out upstream community around open source toolset at DFSP level.
  - Only way to sustainably scale to onboard FIs



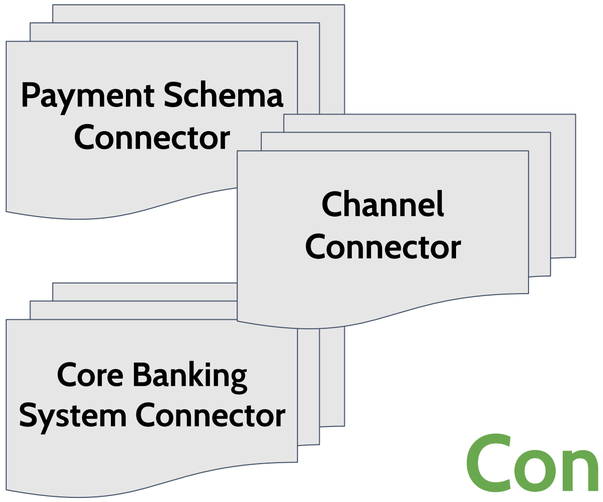
Augment UNCDF Efforts in Myanmar & TZ

Turnkey solution for digitization & digital transformation of MFIs & SACCOs

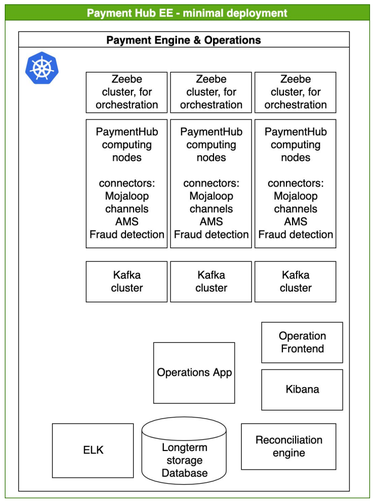
Challenge	Solution
MFIs lack processes & systems to participate in switch	Cloud solution for digitizing and automating core banking operations
MFIs lack systems with 24x7 capabilities	Payment Hub EE can provide stand-in processing
MFIs have wide variety of core banking systems and channels	Extensible by simply building additional core banking system or channel connectors.
MFIs learn in visual manner	Hands-on lab environment with actionable tools
Difficulty connecting dozens or hundreds of MFIs	Multi-tenancy of Mifos and Payment Hub EE enabling economies of scale for shared service providers
MFIs need support and training in getting regulated & connected	Mifos network of local on-the-ground integrators & support partners with deployment & domain expertise.

Benefits of Payment Hub EE

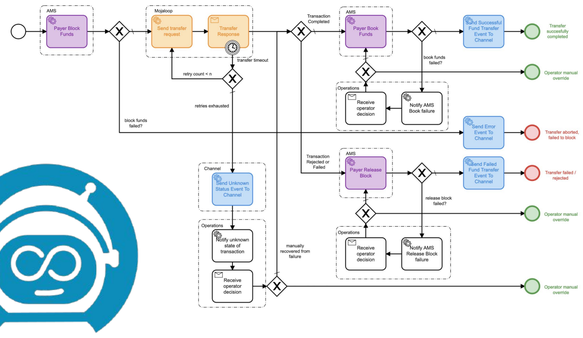
Extensible



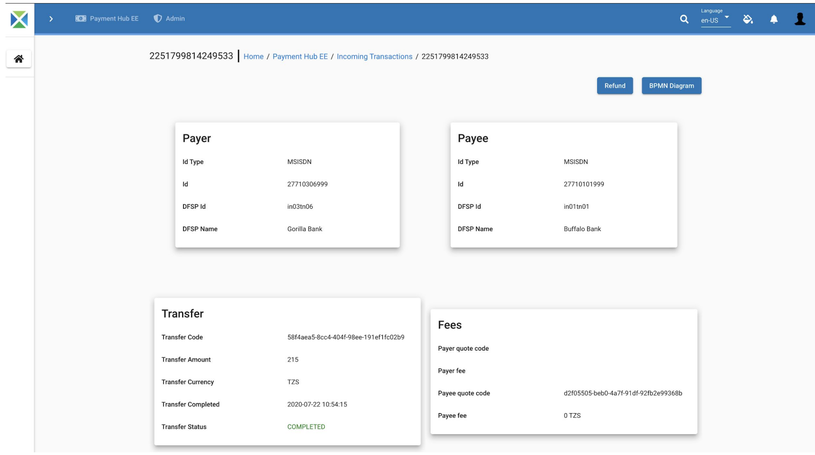
Scalable



Configurable



Practical



mojaloop foundation

Tier 1 DFSPs

Robust Performance
Deployment Flexibility
Ease of Connecting
Operational Controls
Seamless Channel Integration

- Core Banking System Agnostic
- Highly configurable workflow engine
- Extensible via connectors
- Ready to customize and extend
- Vendor-agnostic
- Operational Control Center

Fintechs

Ease of Connection
Rapid Innovation
Sandbox
API Standardization
Use Case Prototyping

- OS Reference Apps
- Interactive Lab Environment
- End to End OS Stack for DFS
- Open Banking API Layer
- GSMA Mobile Money API support

Regulators

Transparency & Visibility
More input into Fraud Detection
Velocity of Cash

- DFSP-level fraud monitoring
- Visibility into on-us transactions
- Generate regulatory reports
- Consistency of data
- User Access for regulators

Advancing Traction & Deployments

Payment Hub EE Traction

Where's it Being Deployed in Mifos Ecosystem

- ❏ Cross-border Payments Network between USA & India
- ❏ Bulk Disbursements for Banco del Bienestar in Mexico
- ❏ Stokvel Digitization in South Africa
- ❏ SACCO Digitization in Kenya
- ❏ Mobile Wallet in Cote D'Ivoire
- ❏ Rural Bank Digitization in Ghana
- ❏ Ministry of Education in Sierra Leone
- ❏ Modernizing payment infrastructure in Europe

Additional Connectors

- ❏ Payment Schema
  - ❏ In-Progress
    - ❏ M-Pesa (Kenya)
    - ❏ SPEI (Mexico)
    - ❏ Africell & MTN (Sierra Leone)
  - ❏ Under Consideration
    - ❏ Aza Finance
    - ❏ Interac (Canada)
    - ❏ MFS Africa
- ❏ Channels
  - ❏ GSMA Mobile Money
  - ❏ Nextgen SIM Overlay

Advancing Adoption of Mojaloop

Fintech Innovation

KANZU BANKING

Sellpay Liberia

Swype.cash

Circle Cash Sudan

darapay
Making prosperity accessible to all Africans

Country-Level POCs & Microswitches

Nigeria

Mexico

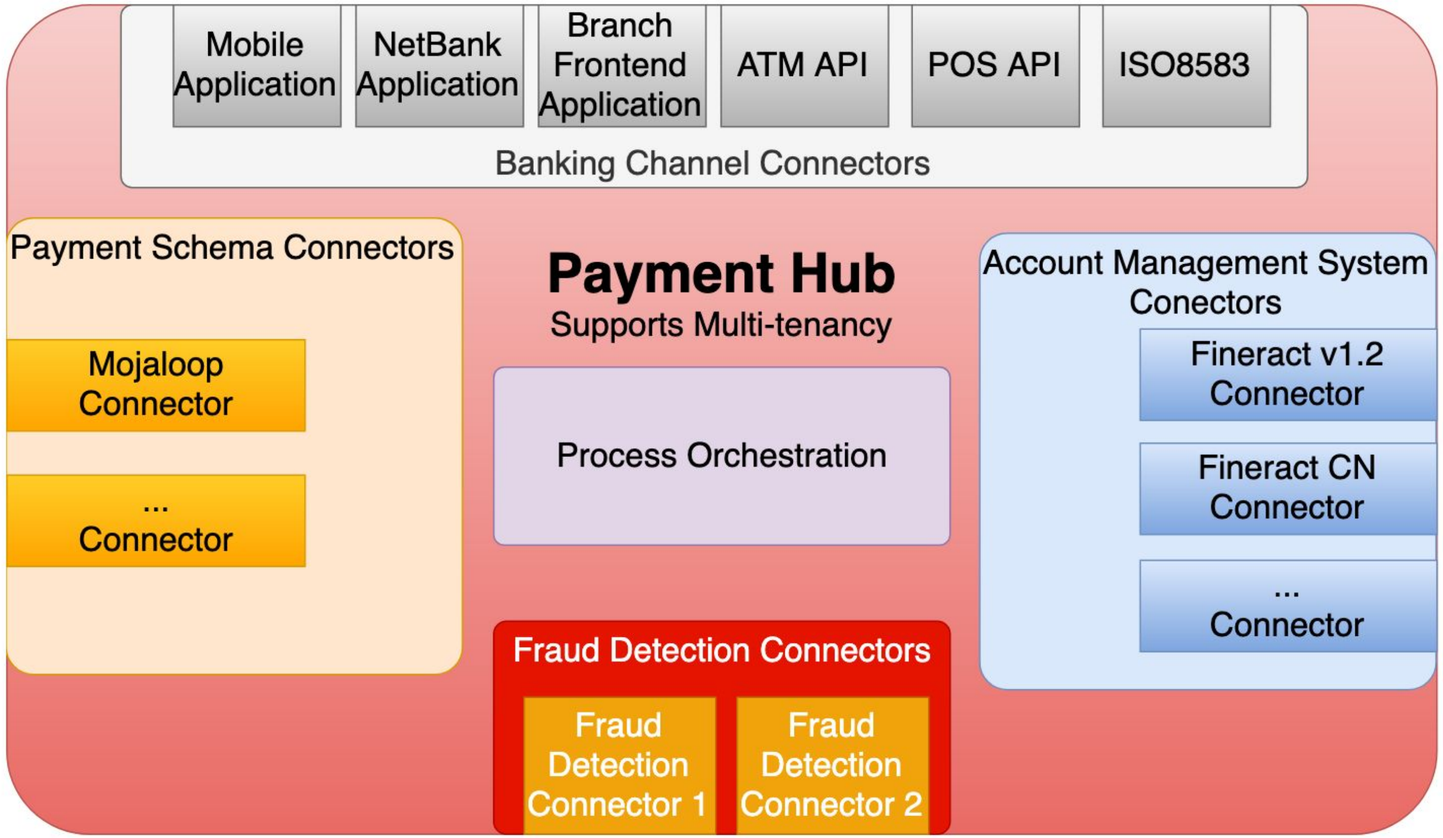
Yemen

mojaloop foundation

Demos & Technical Updates

Payment Hub as an Open Source Asset for the Community

- The role of a payment hub to connect:
  - Financial Institution channels (Mobile, Internet, Branch, Callcenter, ATM, POS, API Gateways)
  - Account Management Systems (AMS / Core banking platform), optionally fraud monitoring tools
  - Payment Schemes, such as Mojaloop
- Need
  - Consistent Way to Connect to Mojaloop
  - Effective Operational Participation
- Additional Capabilities
  - DFSP-level fraud monitoring
  - Bulk Transfer Campaign Management
  - Operational Monitoring
  - Manages the identifier – account relation
  - Trigger notifications
- Built on proven open-source technology:
  - Java, SpringBoot, Kafka, Elasticsearch
  - Apache Camel, Camunda Zeebe
  - Kubernetes

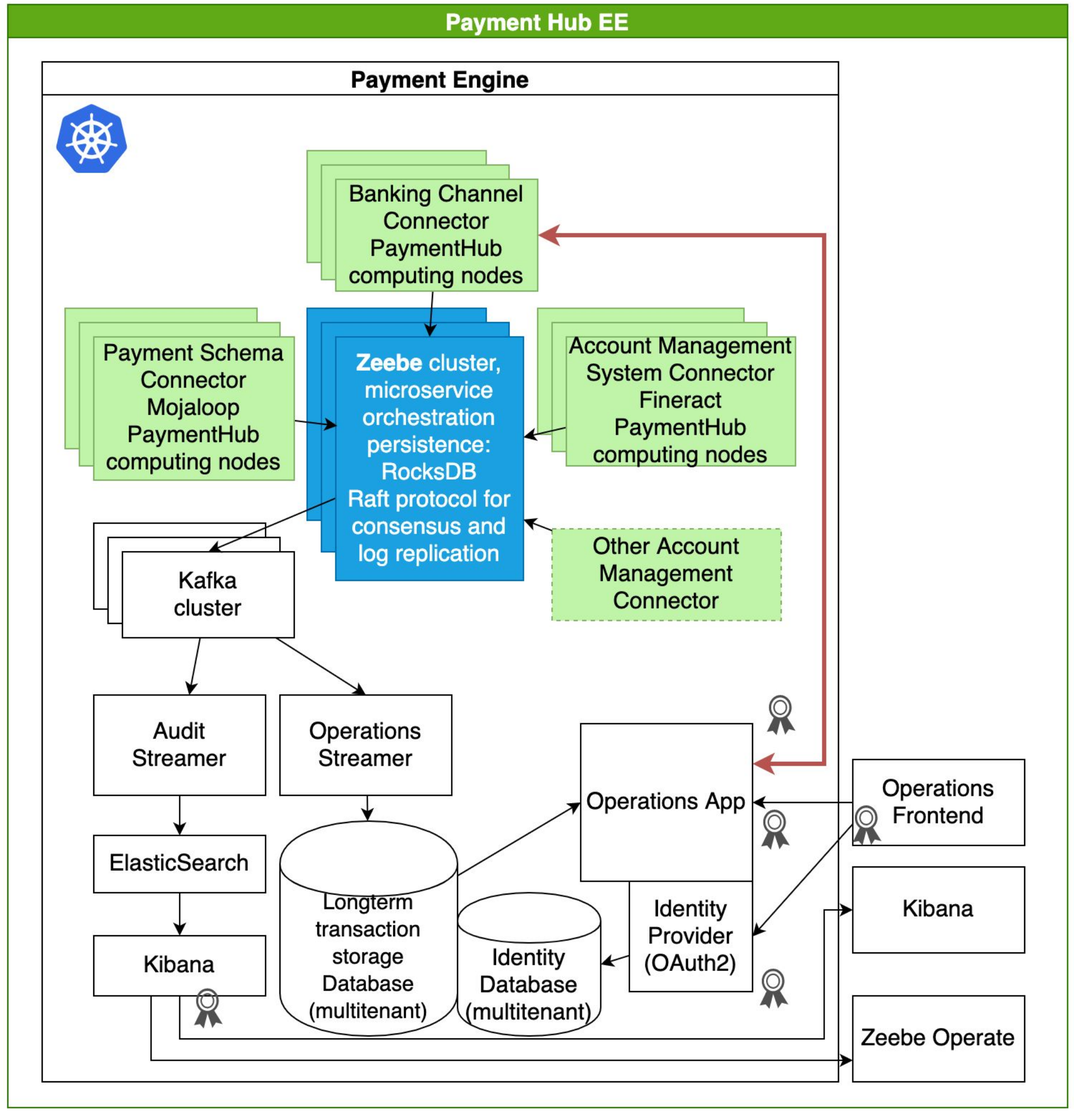


mojaloop foundation

Payment Hub Components

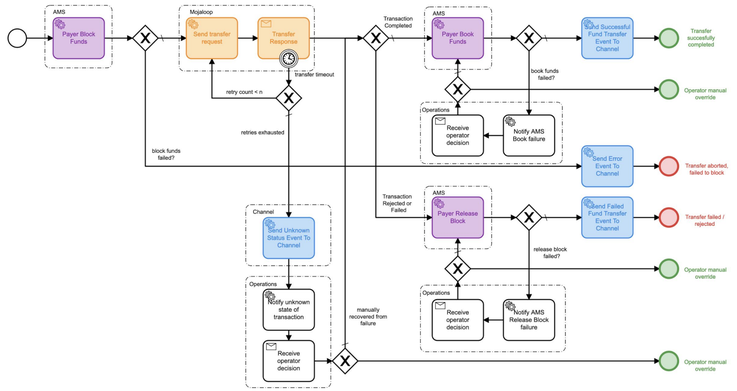
Microservices based architecture to support easy customization for integrators

- microservices orchestration
- microservices for the connectors
- running in a Kubernetes cluster
- operations application for DFSP command center



Focus on system integrators

- BPMN let implementation team **focus on the business flow changes inside the given DFSP**, making adoption and customization much easier
- **Multi programming language support for component implementation**, let implementation team to create the necessary connector components in any of the supported languages, the engine is able to orchestrate the steps. (e.g. multiple AMS integration is required), using gRPC protocol for efficient communication
- **Connector components are stateless, easy and simple implementation**
- No additional components required in the realtime engines, no databases to maintain, all active process states are stored in the embedded RocksDB data store. Auditlogs, and other events are sent to a Kafka cluster before storing them for long term access
- Scalability - using partitioning (sharding), the system scales horizontally to create thousands of workflow instances per second
- Fault tolerance enables, that system recovers from a failure without data loss



mojaloop foundation

Multitenancy

Single deployment serving multiple DFSPs

Isolated audit logs for the different DFSPs served with separate user databases for the DFSP operators

Benefits:

- Single operator could provide services for multiple individual DFSPs while sharing operational costs
- In case using a multi-tenant core banking platform (e.g. Mifos) a single Payment Hub EE deployment could serve all tenants (DFSPs)

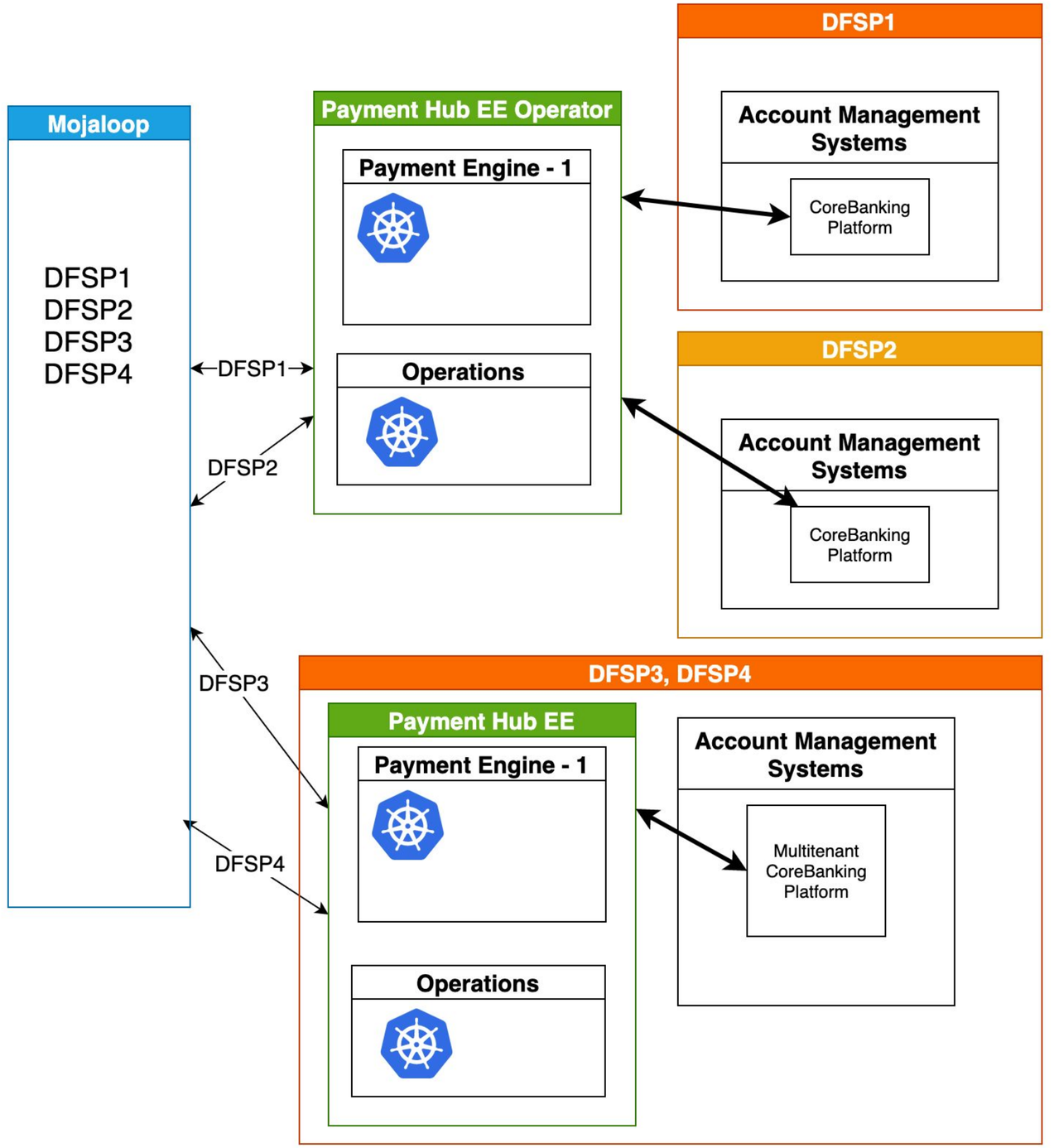
Multitenancy

Helping adoption for smaller MFIs

-sharing operational cost
-simplifying management
-reduce implementation effort

Aggregator model for multiple smaller DFSPs

Multitenant model for already aggregated DFSPs served from a single multi-tenant core banking

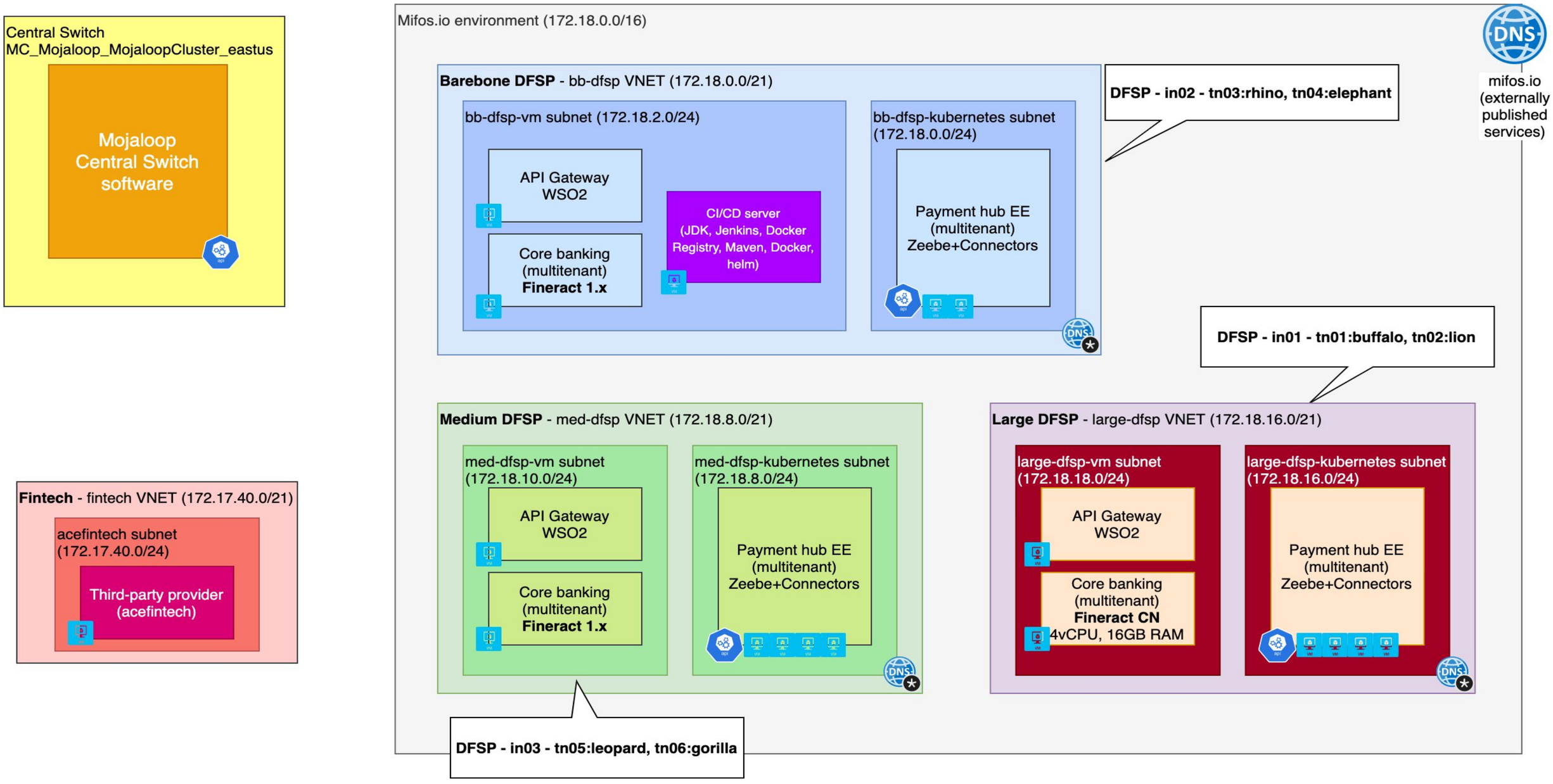


mojaloop foundation

Integrated LAB Environment

- 3 independent deployments
- 2 Mifos 1.x and 1 Mifos CN multitenant core banking deployments
- 2 DFSPs in each deployment unit utilising the multitenant capability of Fineract (one tenant corresponds to a DFSP)
- 1 Mojaloop instance to enable instant payments across the 6 DFSPs
- 1 Fintech application to utilise the OpenBanking APIs provided by the DFSPs. This enables to demonstrate an account information aggregation and payment initiation third party
- 1 CI/CD server to be able to build and deploy the various microservices of the payment hub
- Join the Mifos Slack to get access to the environment!

Mifos Lab - Full featured lab environment



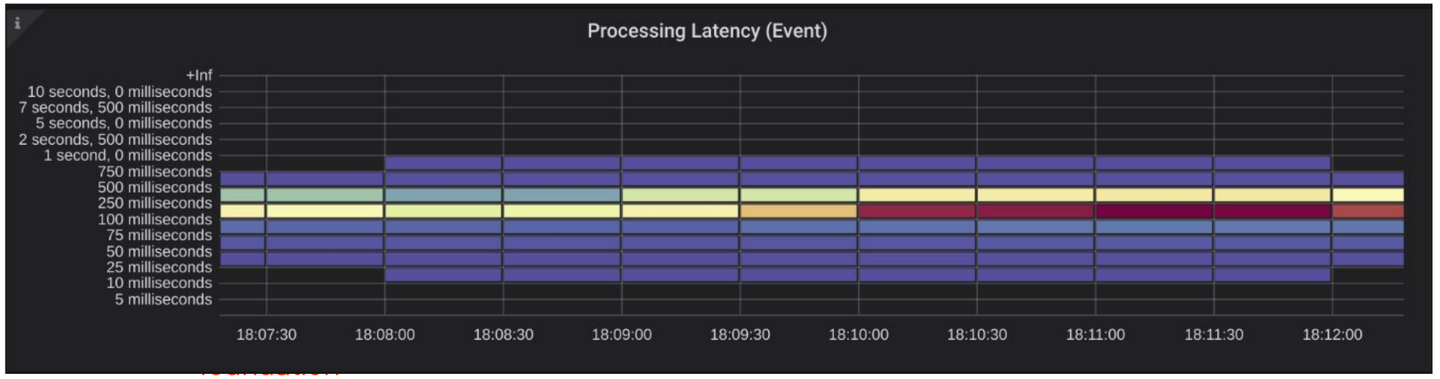
Performance measurements

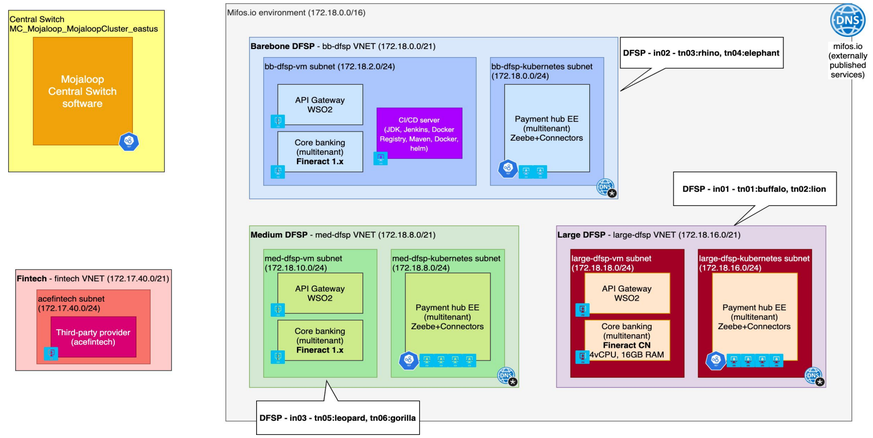
Payment Hub EE Large Lab Environment

- 4 nodes Kubernetes cluster on Azure (3 x 8 cpu core, 56Gb ram, 1 x 2 cpu core, 2Gb ram)

Test benchmark

- 10000 complete payment transaction flows
- number of parallelly running workflow instances peaked over 5200 workflows
- all transactions finished within 3:15, hitting the average of 51 cfTPS
- single event process latency primarily within 100-250ms
- executing payment transfer operations with over 300 fTPS





Tuning

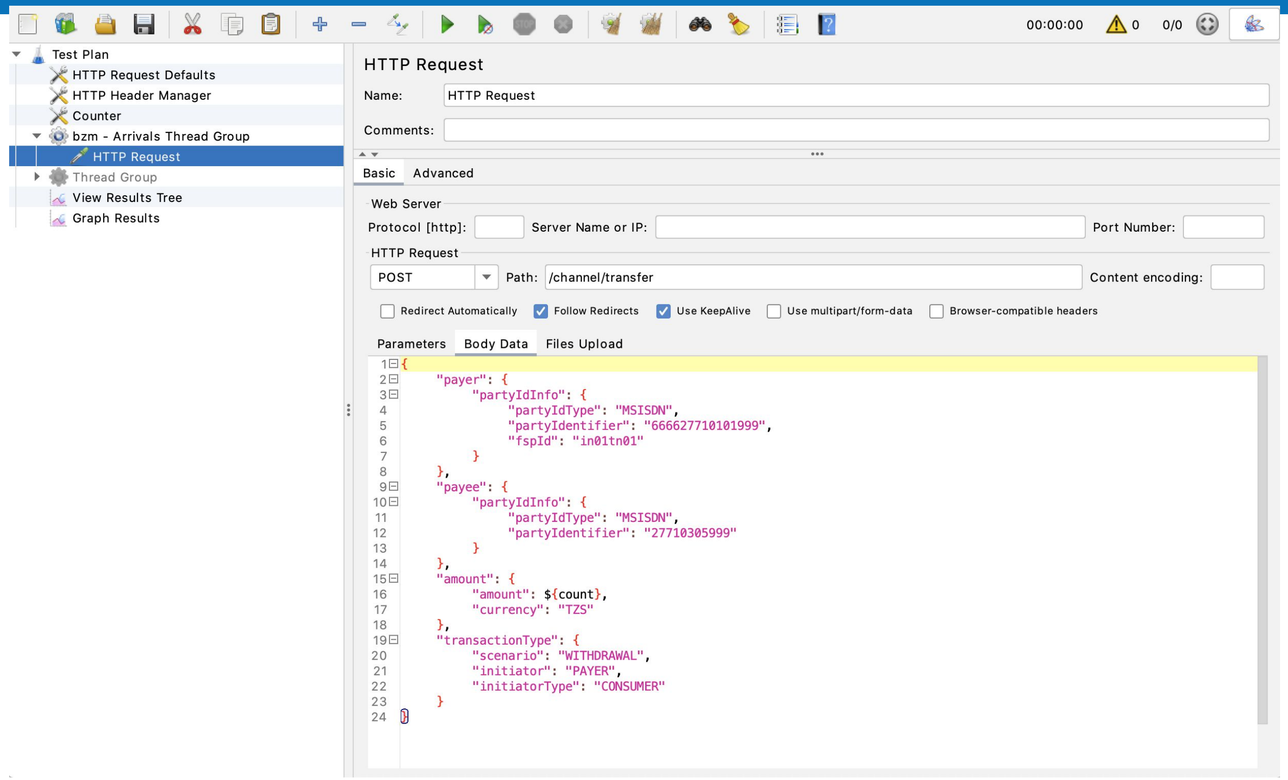
Performance tuning documentation in the PHEE Handbook:

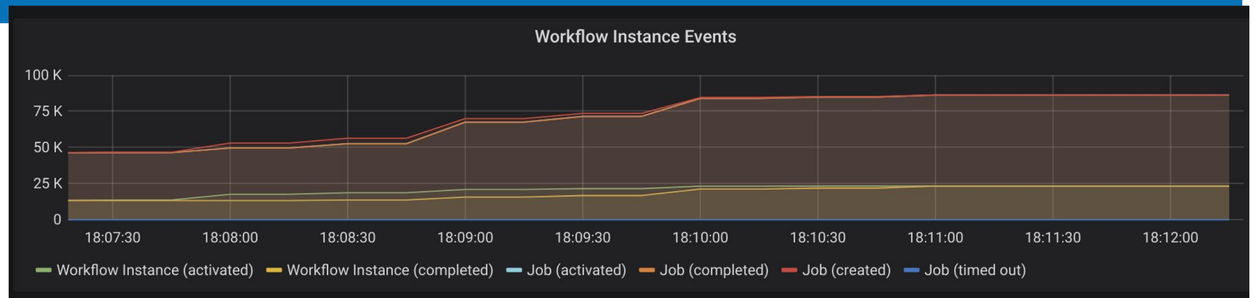
https://mifos.gitbook.io/docs/payment-hub-ee/overview/tuning-and-performance

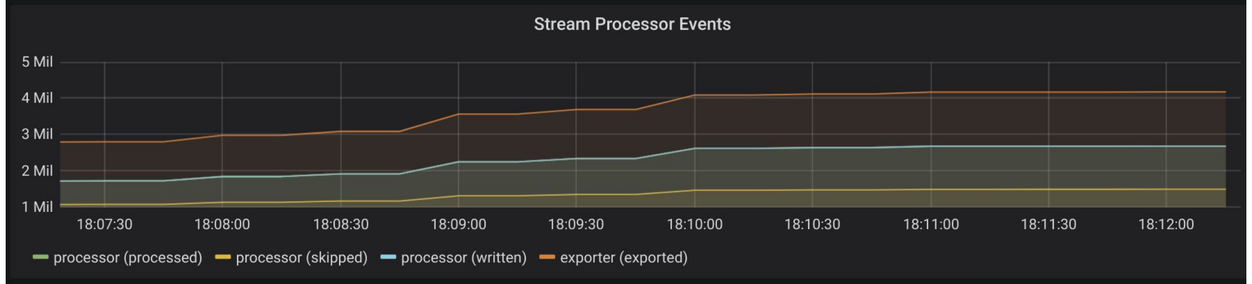
Primary tuning considerations

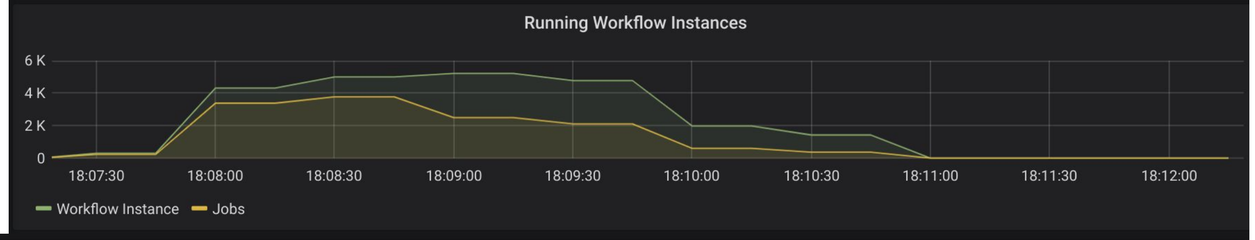
- Zeebe orchestration engine parameters (partitions count, replication factor, CPU / IO thread counts, JVM garbage collector options)
- replace built-in Elasticsearch exporter with PHEE's improved asynchronous Elasticsearch exporter
- tune backpressure options / disable for throughput benchmarks
- enable Prometheus metrics to get the necessary insights
- tune the number of connector pods & worker thread counts in the clusters to match the target load
- running on AMD cpus seems to have a serious impact on performance (under validation)

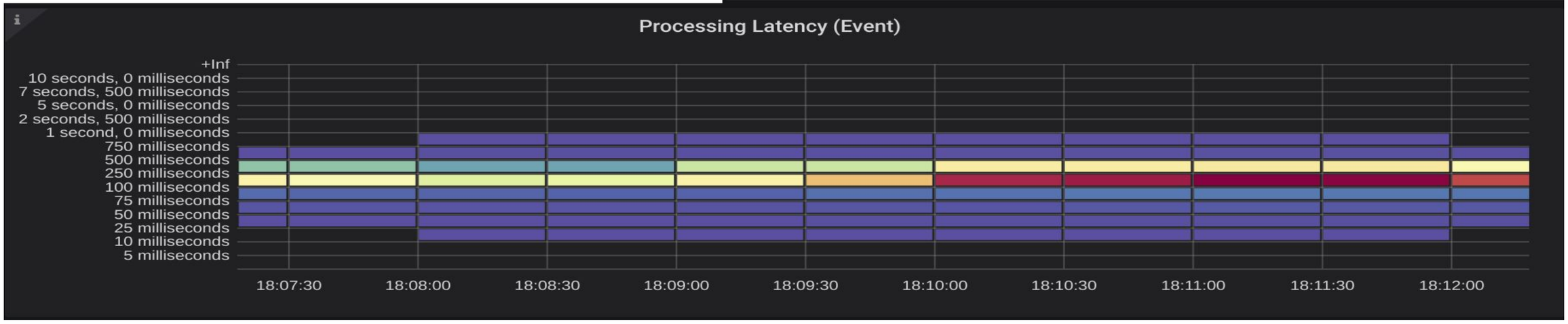
Load testing and measurements



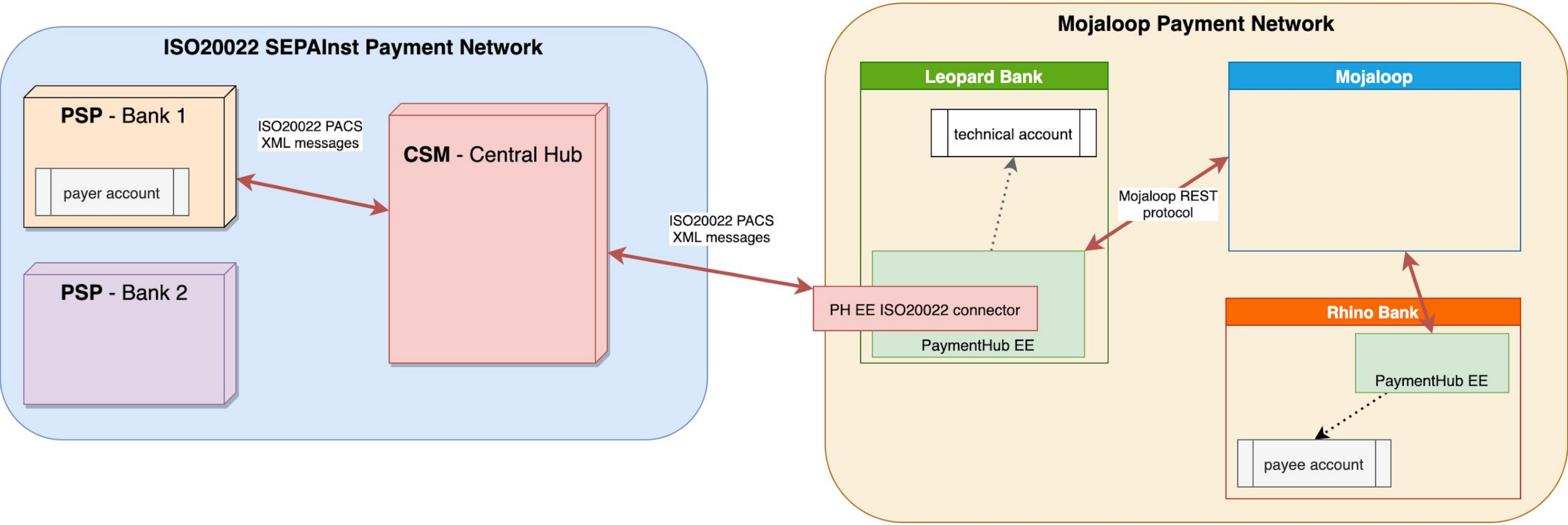




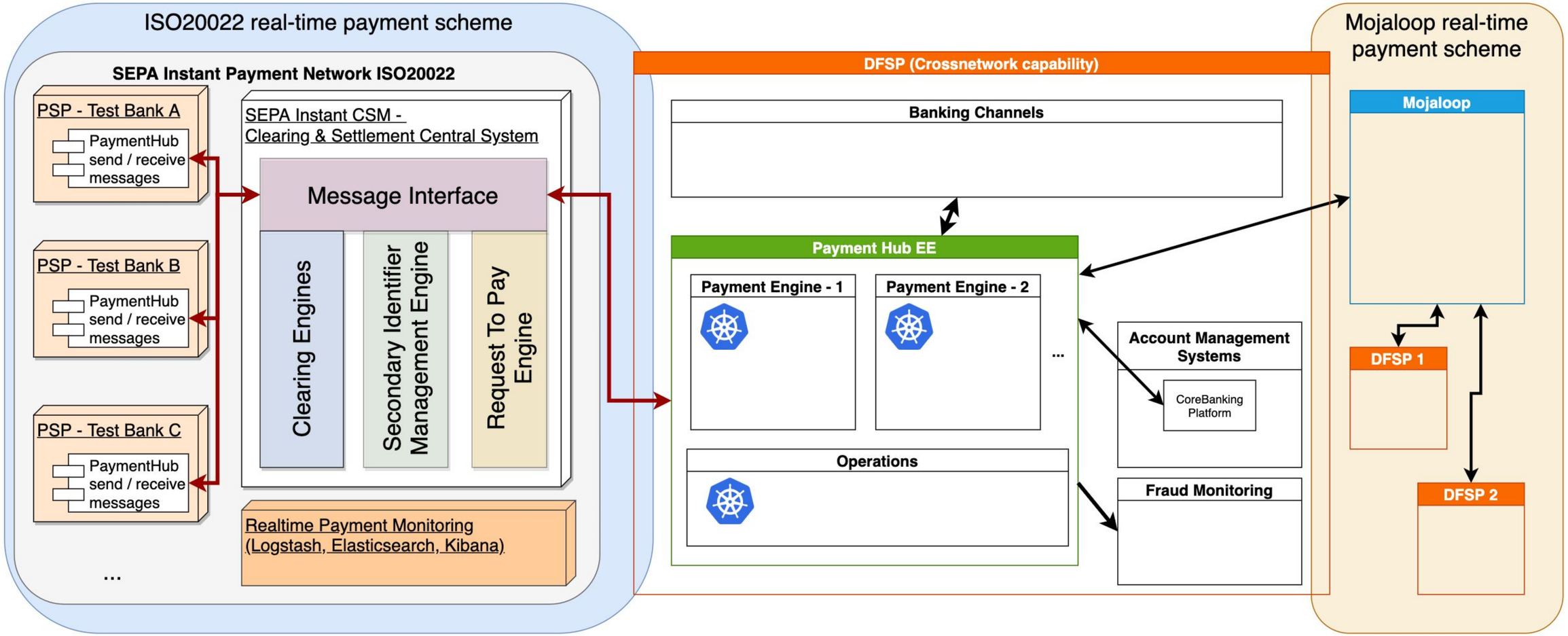




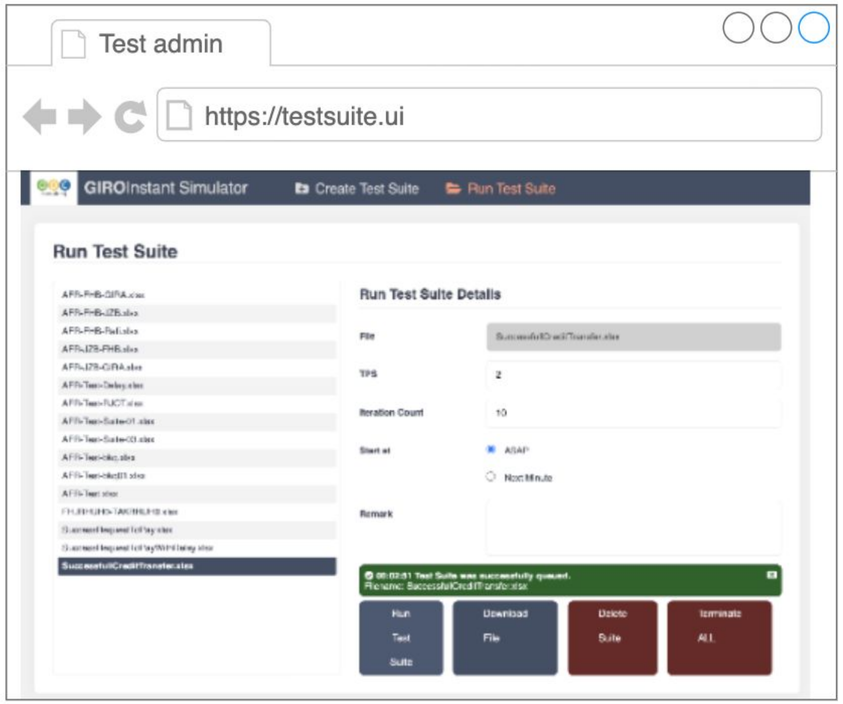
Crossnetwork - ISO20022 (Payments Clearing and Settlement - pacs.008, pacs.002) - Mojaloop

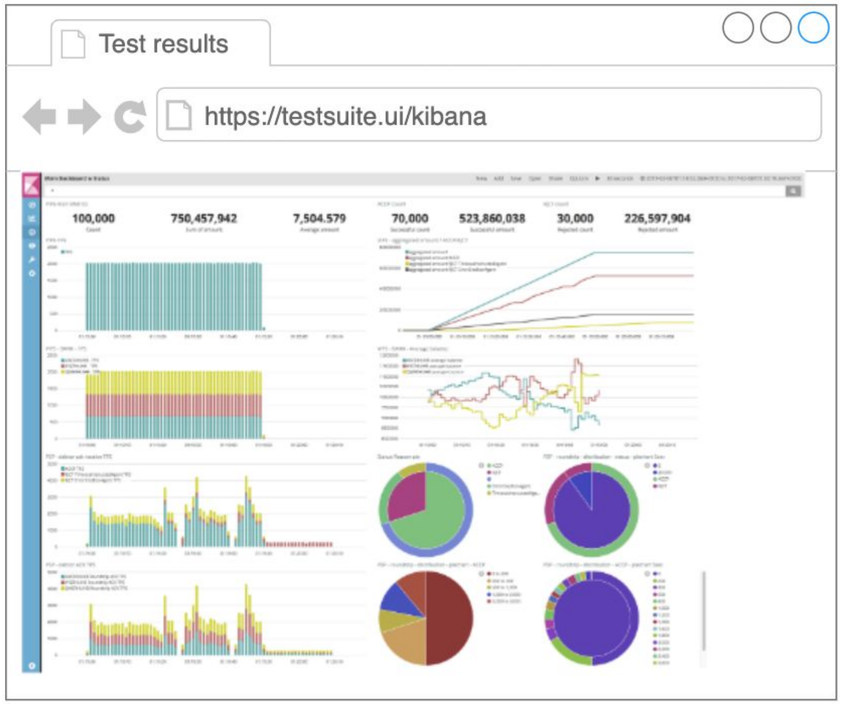


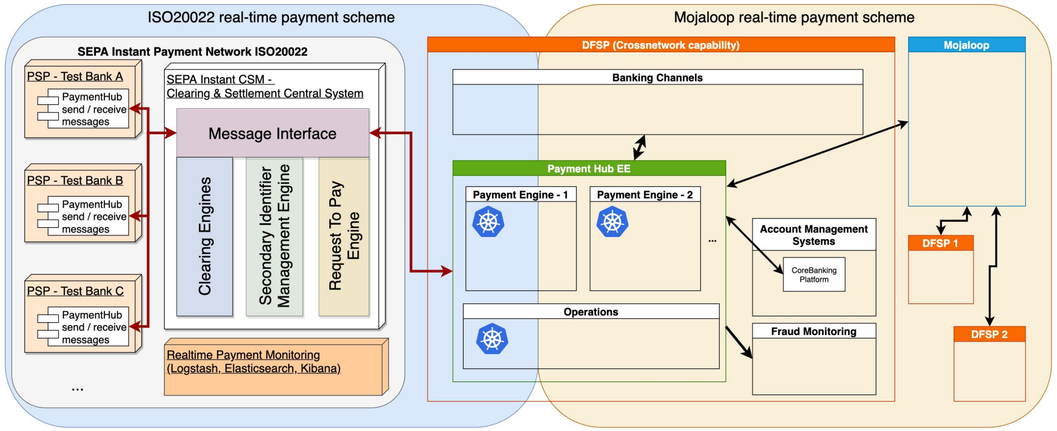
Crossnetwork - ISO20022 (Payments Clearing and Settlement - pacs.008, pacs.002) - Mojaloop



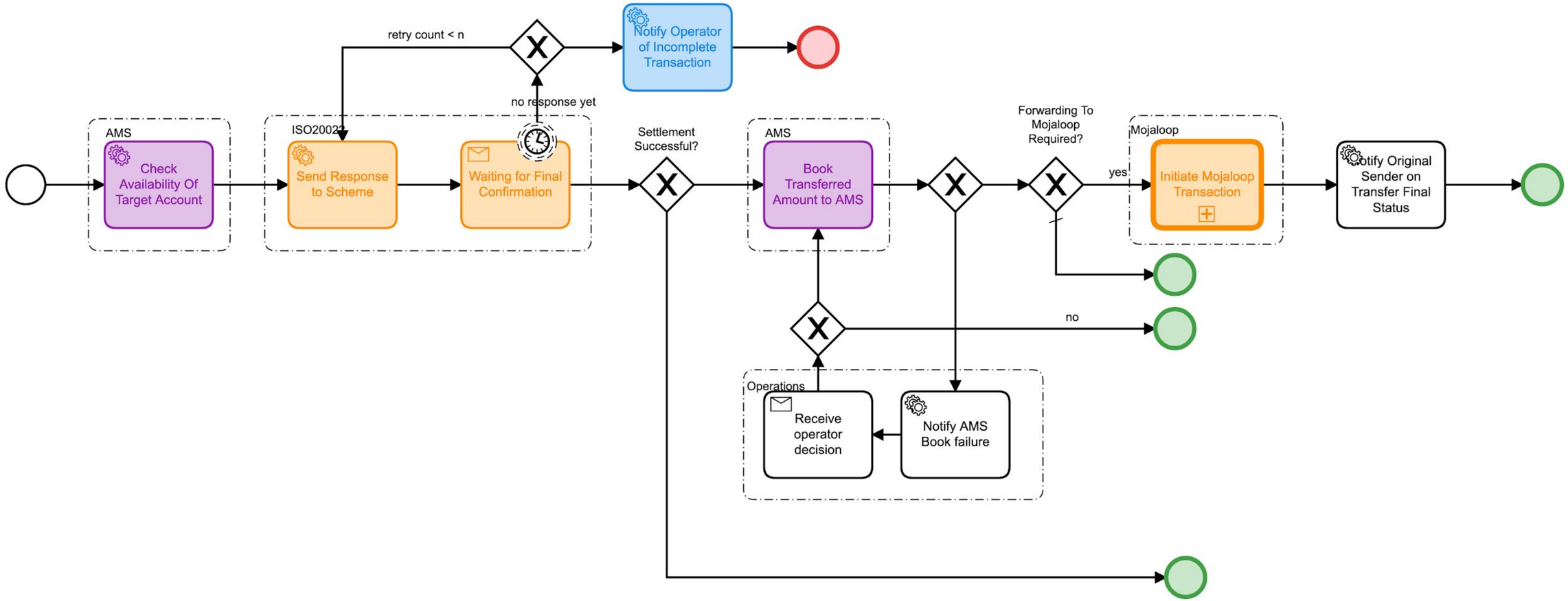
Crossnetwork - ISO20022 - Testing







Crossnetwork BPMN flow

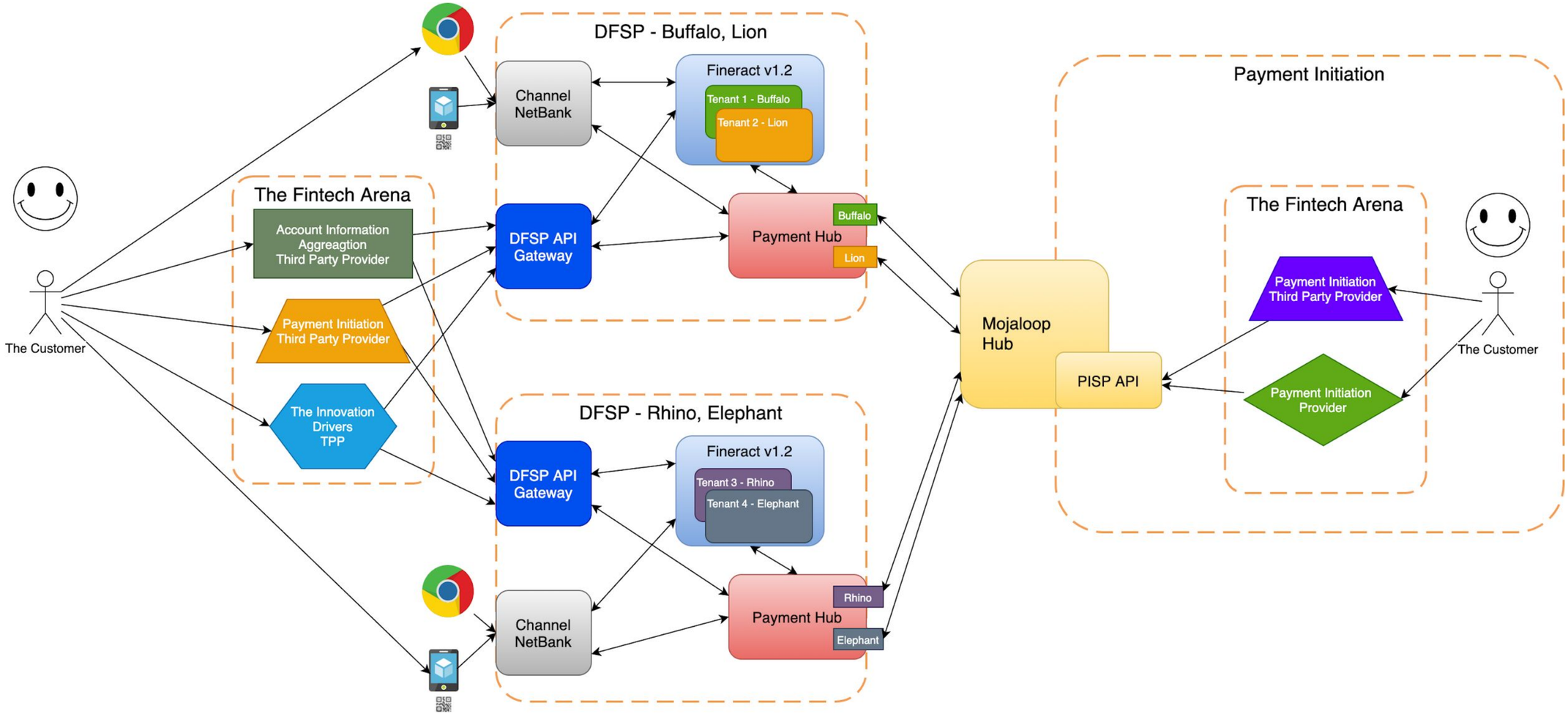


Addressing an account

- How to address the ultimate payee in a structured way in the Mojaloop transfer message structure?
- ISO20022 has terms Creditor and UltimateCreditor
- We used the RemittanceInfo in the pacs.008 message to identify a Mojaloop Account:

```
mjlp:msisdn:27710203999
```

Providing Openbanking API to Fintechs



- Supporting the OpenbankingAPIs at DFSP
- Working towards to enable the PISP API implementation

mojaloop foundation

Roadmap for Payment Hub EE

Continue working on bulk payments and OpenG2P

Continue performance testing and high availability testing

Providing stand-in capability

Reconciliation

- Real-time transactions, Mojaloop reports, Corebanking ledger

Bulk Payment Support

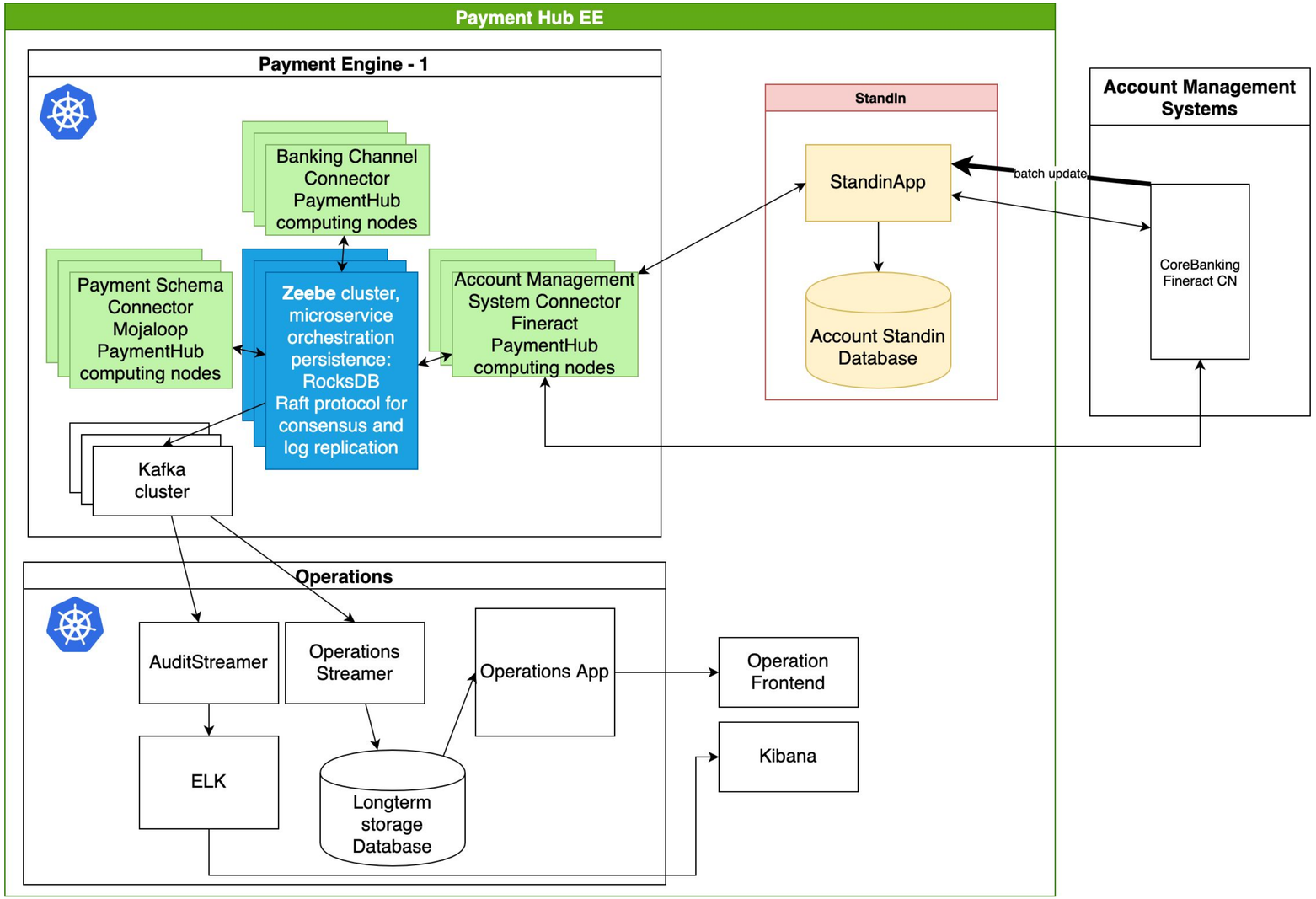
- Preprocessing bulk payments
  - Lookup Payee's DFSP ID for the transactions to determine target DFSP
  - Splitting the incoming bulk into smaller batches per target DFSP
    - On-us transactions can be handled differently
  - Manage communication with Mojaloop for the batches
  - Aggregate incoming results to provide response to the bank's channel
  - In case transfering from a single account (pension, aid), booking can be individual, aggregated by target DFSP or single grand total

Stand-in

Integrated into the Payment Hub EE solution a stand-in system could provide functionality in case the Account Management System is not available.

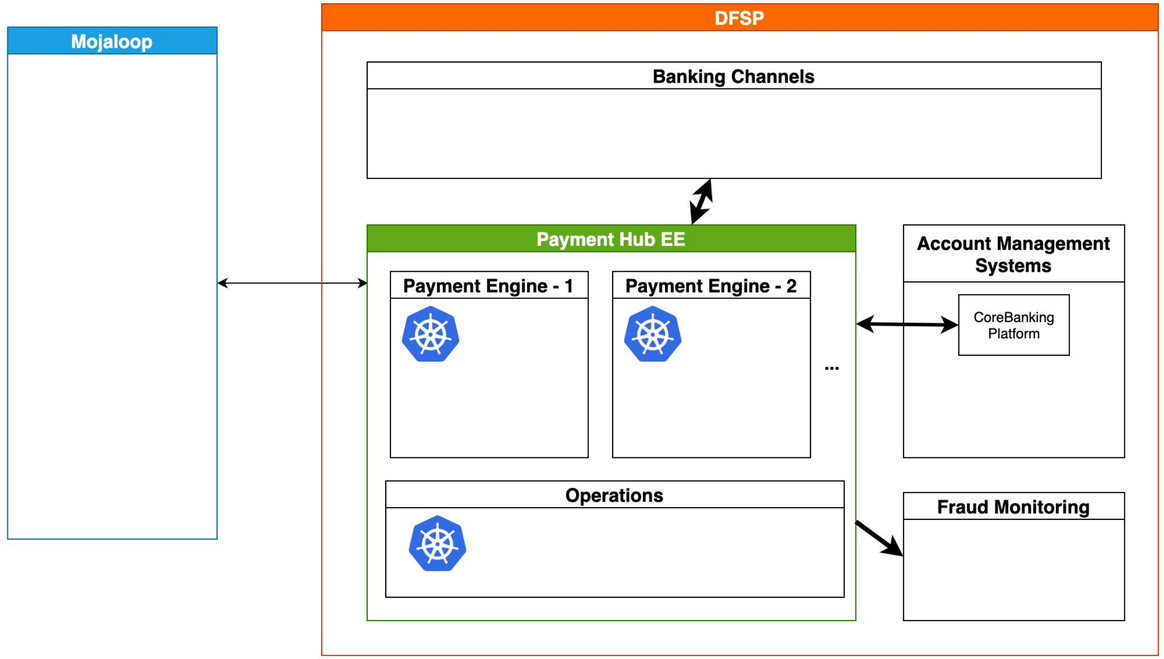
MFI's systems are often not 24x7.

- only incoming transactions - simple solution, requires only synchronizing the valid account and party ids
- both incoming and outgoing transactions - requires more complex solution to minimize risk of overdraft



DFSP Level Fraud Detection - Regulatory Fraud Detection

- The payment process could support the DFSP level Fraud monitoring tools
  - to stop transactions or alert operators
- There are requirements to keep the on-us transactions inside the DFSP
- PHEE can be the component to provide information to regulators with the required amount of granularity of on-us transaction details (considering GDPR and similar regulations)
  - send all transactional information as individual reporting messages
  - aggregate on-us traffic data for a required period (minutes to days)



Documentation

Business Overview

- Vision
- Benefits
- Target Users

Technical Overview

- Deployment models
- Lab Environment
- Source code
- Installation instructions
- Tuning and performance

mojaloop foundation

app.gitbook.com/@mifos/s/docs/payment-hub-ee/overview

Edits 3
Activities
Share
Design
Teams
Integrations
Analytics
Advanced

LionFintech FPP demo application
How to

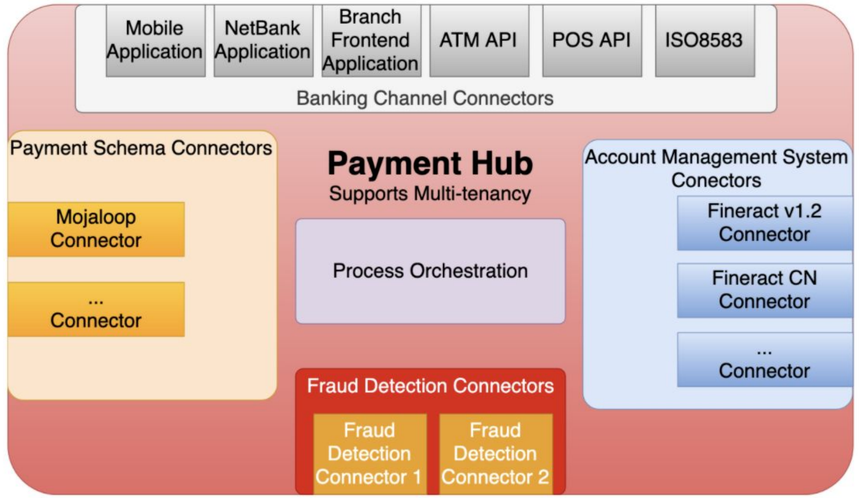
PAYMENT HUB EE

Business Overview
- Vision
- Benefits of Payment Hub EE
- Target Users of Payment Hub EE
- Using Payment Hub EE
- Use cases

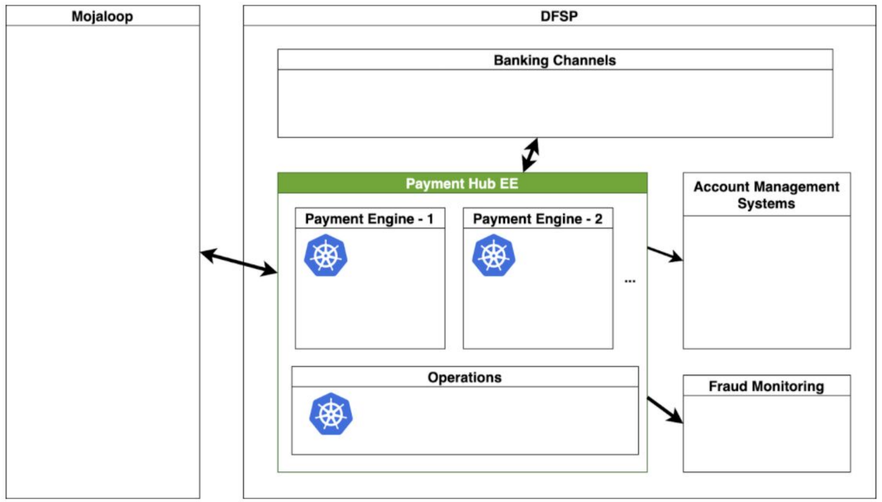
Technical Overview
- Deployment models
- Microservices orchestration - Zeebee
- Lab environment
- Example
- Secondary Identifier Storage and Registration
- Payment Hub APIs
- Source Code repositories
- Installation instructions

New
Files

Design



Logical Model of the Payment Hub



The role of the Payment Hub EE at a DFSP

Thank You

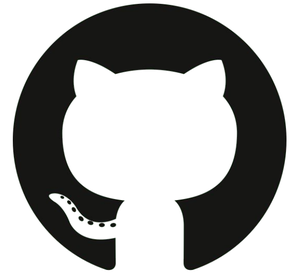
- Miller Abel, Kim Walters & Ariel Delaney
- Core OSS Team

Edward Cable
edcable@mifos.org
https://mifos.org

Istvan Molnar
istvan.molnar@dpc.hu
https://dpc.hu

mojaloop foundation

github.com/openMF
github.com/apache/fineract
https://fineract.apache.org



Browse the Docs:
https://mifos.gitbook.io/docs/payment-hub-ee

Explore the Code:
https://github.com/openMF?q=ph-ee

Discuss on Slack: https://bit.ly/3eMoVS1

Request Access to the Lab:
https://mifos.gitbook.io/docs/payment-hub-ee/overview/lab-environment

Backup Slides

Deployment models

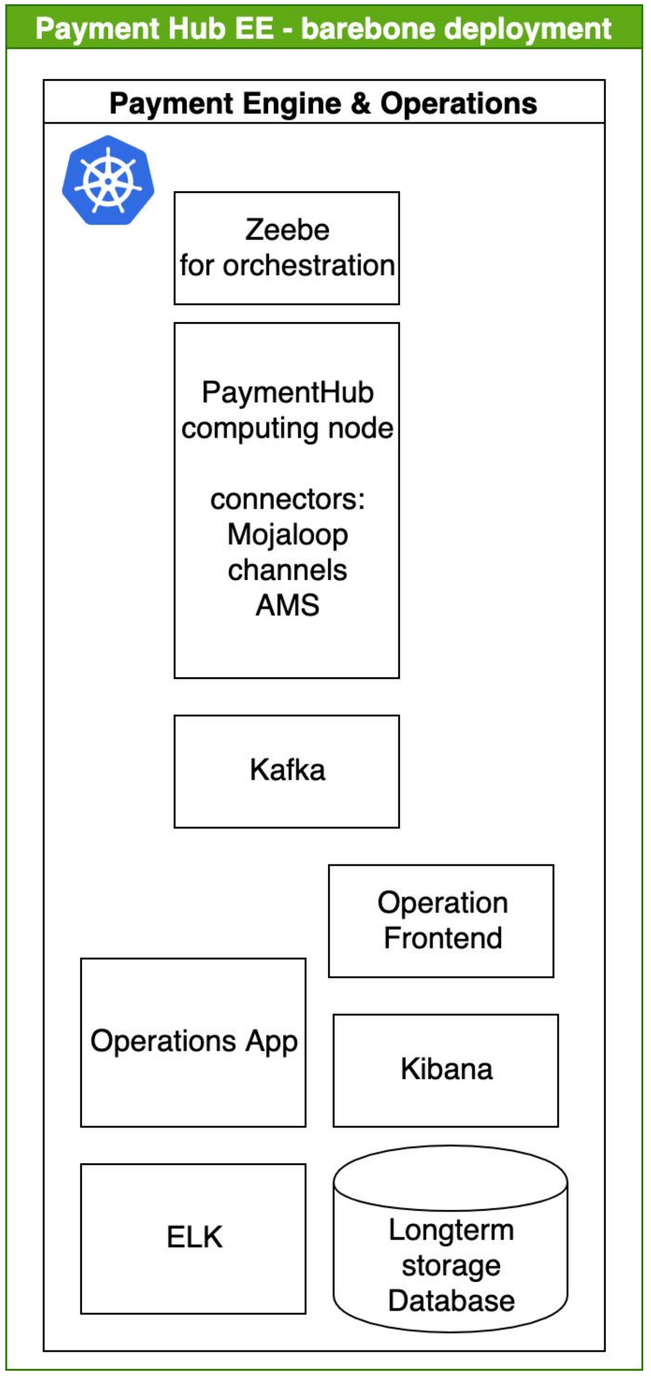
On-premises and any of the cloud providers

Shared service serving multiple DFSPs run by an aggregator (multiple SACCOs, credit unions on a multitenant setup) or dedicated setup

Depending on the DFSP requirements it could be deployed as

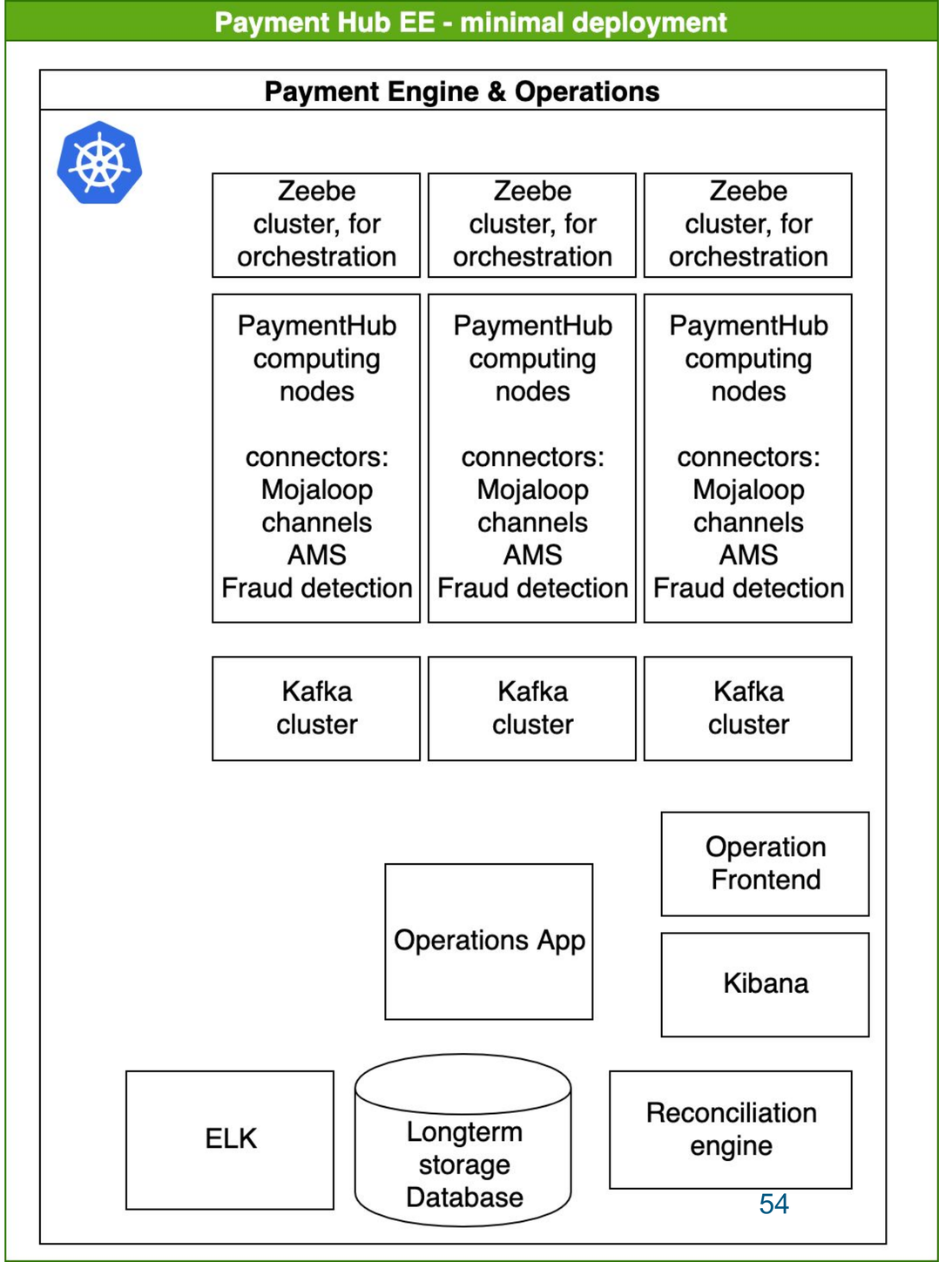
- **barebone** - single instance of components, minimized resource usage, no loss of functionality
  Might not run on a feature phone, but we will get there.
- **medium** - single realtime engine
- **fully scaled** - multiple realtime engines

The difference is in availability, fault tolerance and the volume of transactions, which can be handled.

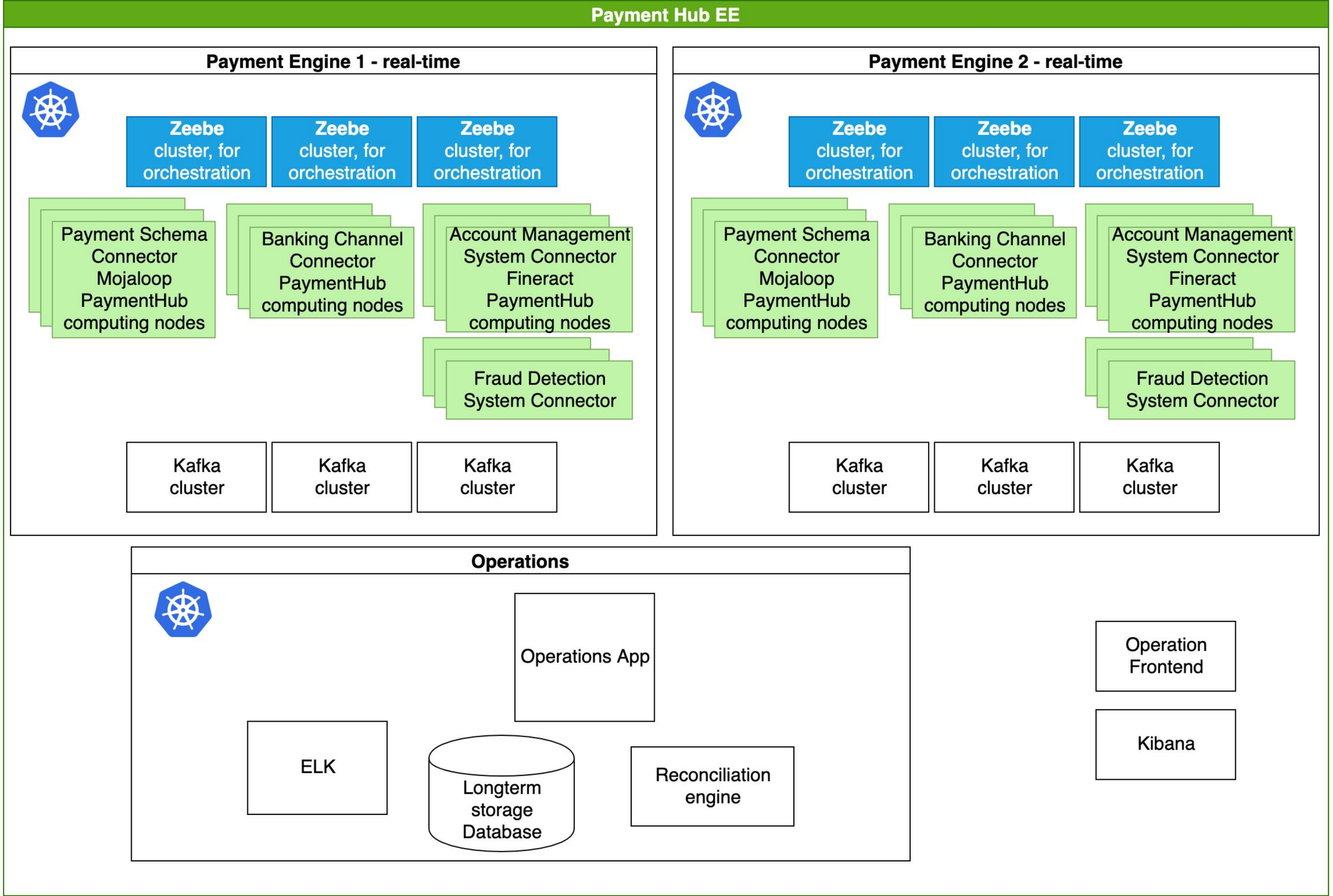


Deployment model - medium

- Medium deployment
  - Single kubernetes cluster to contain all the necessary components
  - Fault tolerance provided by the clustered components
  - Stretched installation across data centers possible, but not ideal
- Full scale deployment
  - Multiple independent payment engines (a single engine is collocated for performance), enabling complete version upgrades without service interruptions
  - Running in different data centers on independent network connections (high availability, fault tolerant even in case of disaster scenarios)
  - Partitioning the load across the engines



mojaloop foundation

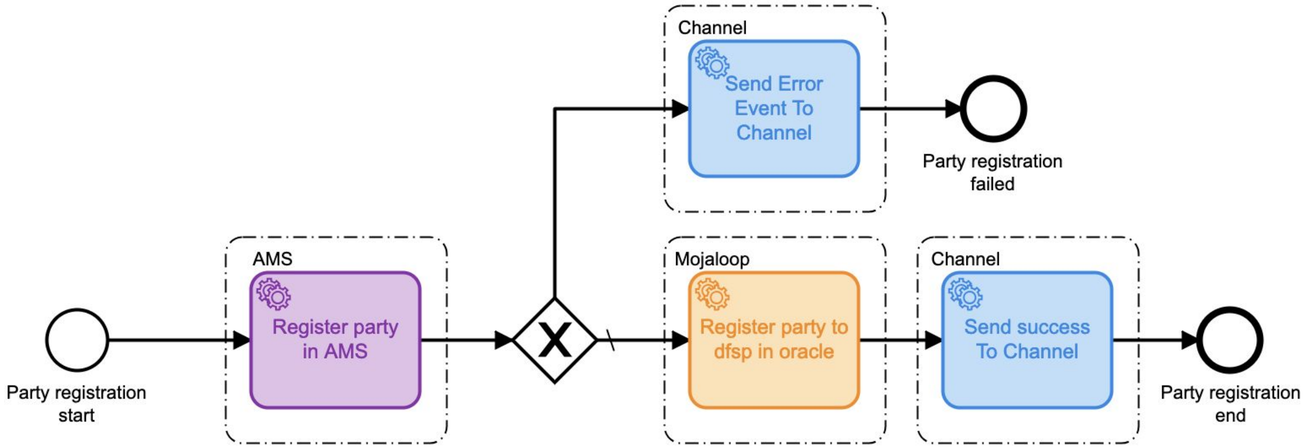

Payment Hub EE
Payment Engine 1 - real-time
Zeebe cluster, for orchestration
Zeebe cluster, for orchestration
Zeebe cluster, for orchestration
Payment Schema Connector Mojaloop PaymentHub computing nodes
Banking Channel Connector PaymentHub computing nodes
Account Management System Connector Fineract PaymentHub computing nodes
Fraud Detection System Connector
Kafka cluster
Kafka cluster
Kafka cluster
Payment Engine 2 - real-time
Zeebe cluster, for orchestration
Zeebe cluster, for orchestration
Zeebe cluster, for orchestration
Payment Schema Connector Mojaloop PaymentHub computing nodes
Banking Channel Connector PaymentHub computing nodes
Account Management System Connector Fineract PaymentHub computing nodes
Fraud Detection System Connector
Kafka cluster
Kafka cluster
Kafka cluster
Operations
Operations App
ELK
Longterm storage Database
Reconciliation engine
Operation Frontend
Kibana

Examples of Use Case Workflows

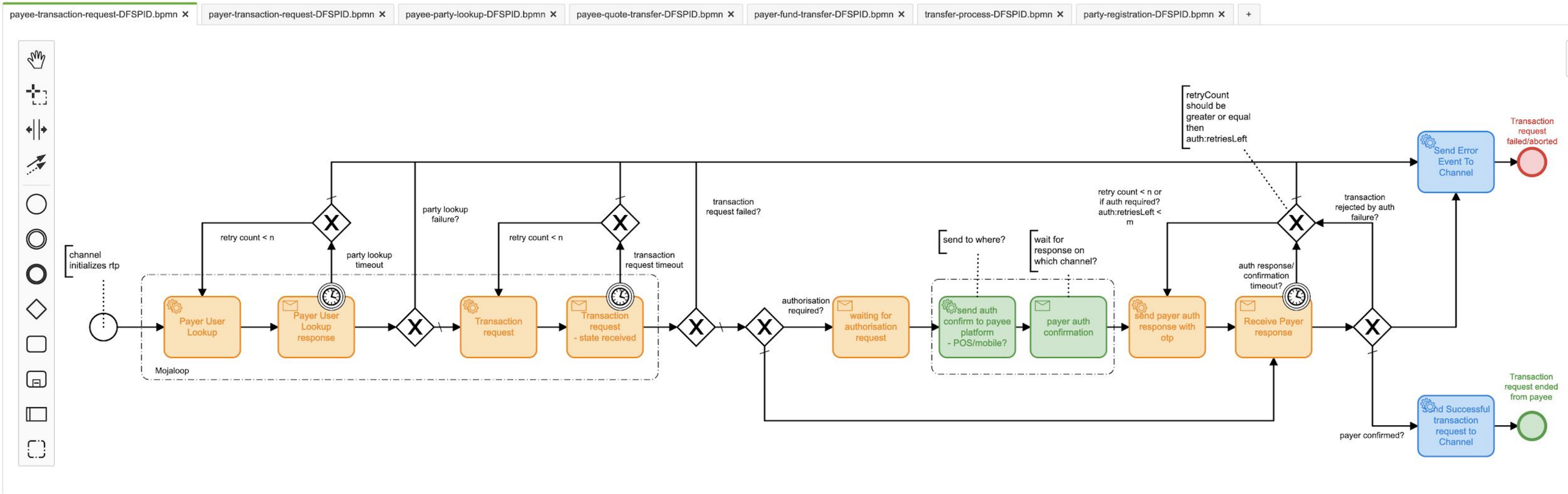
Using Zeebe Modeler for process orchestration at heart of Payment Hub EE

Party Identifier Registration

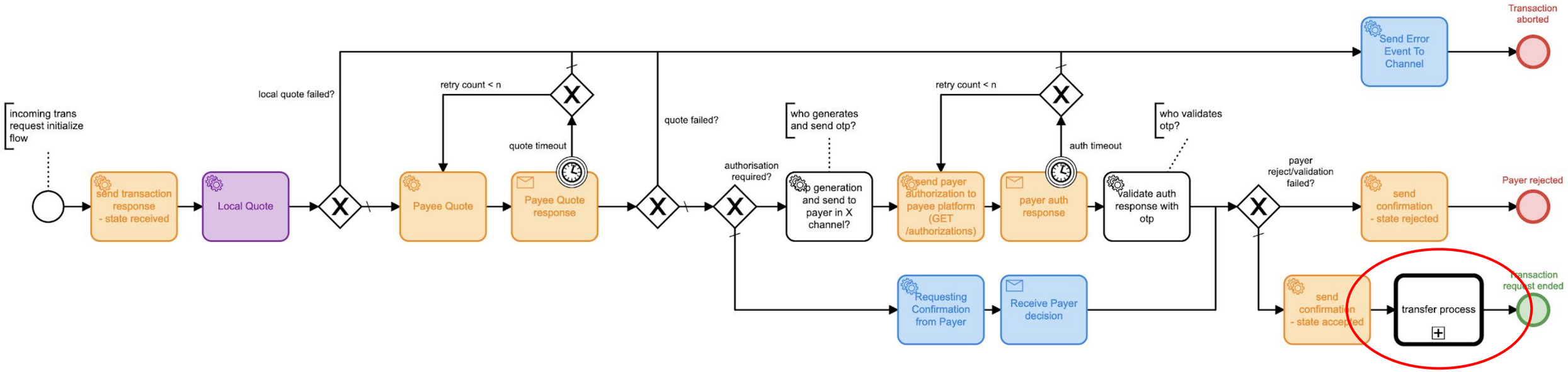
- Initiate the association of a party identifier (MSISDN) with an account at DFSP
- Register identifier in the DFSP systems (in the Account Management System in our environment)
- Manage the registration process with error handling at the “Oracle”



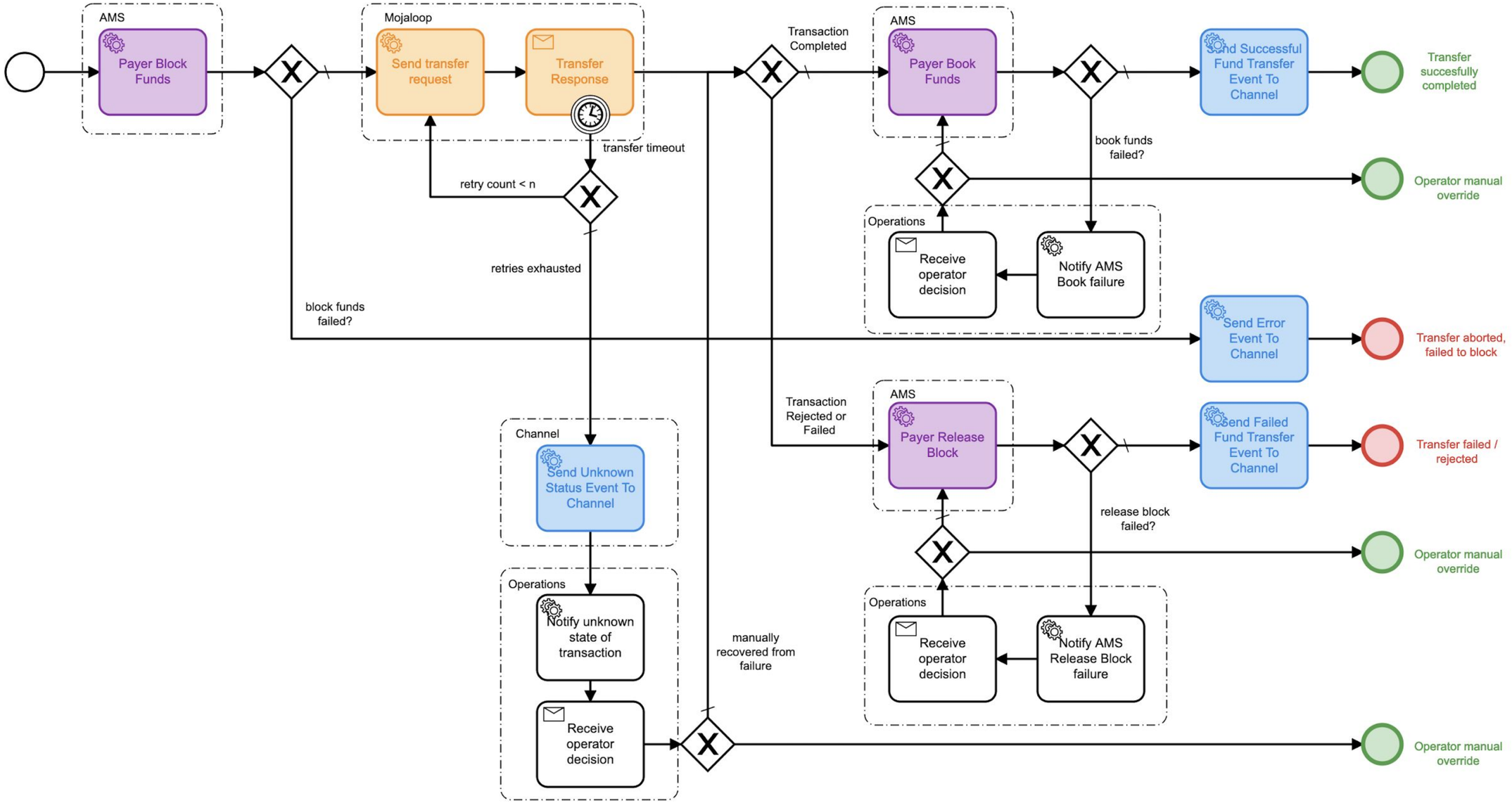
Payee Initiated flows - Request To Pay



Accepting request to pay at Payer



Payer Fund Transfer



Operational Control Center for DFSP actions

Authentication and authorization with privileges for different actions

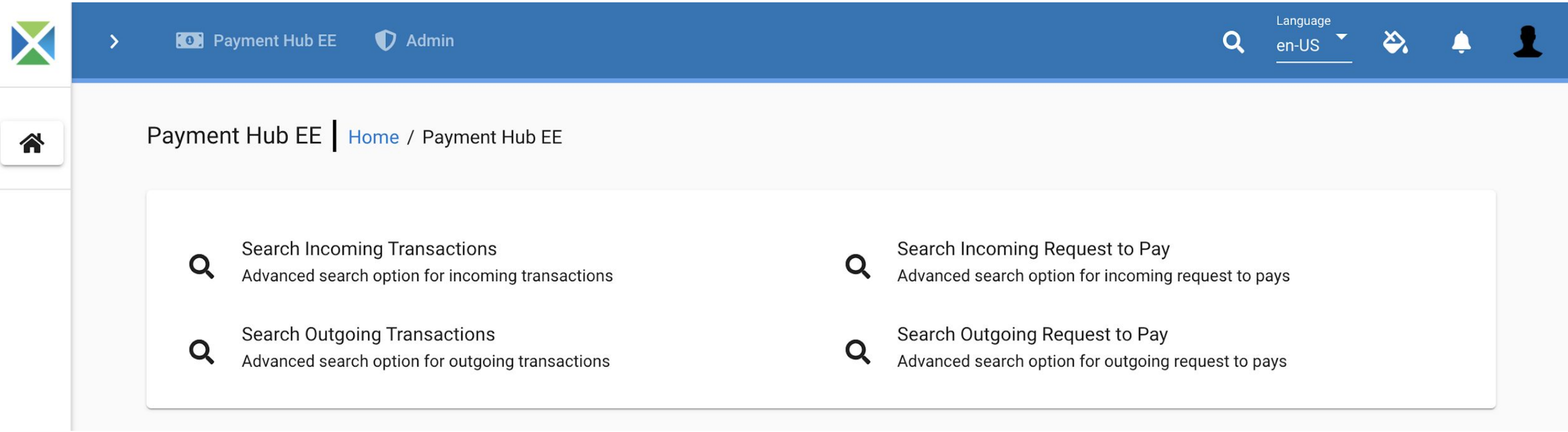
Complete segregation of tenants, separate databases and users

Search and detailed view of transactions

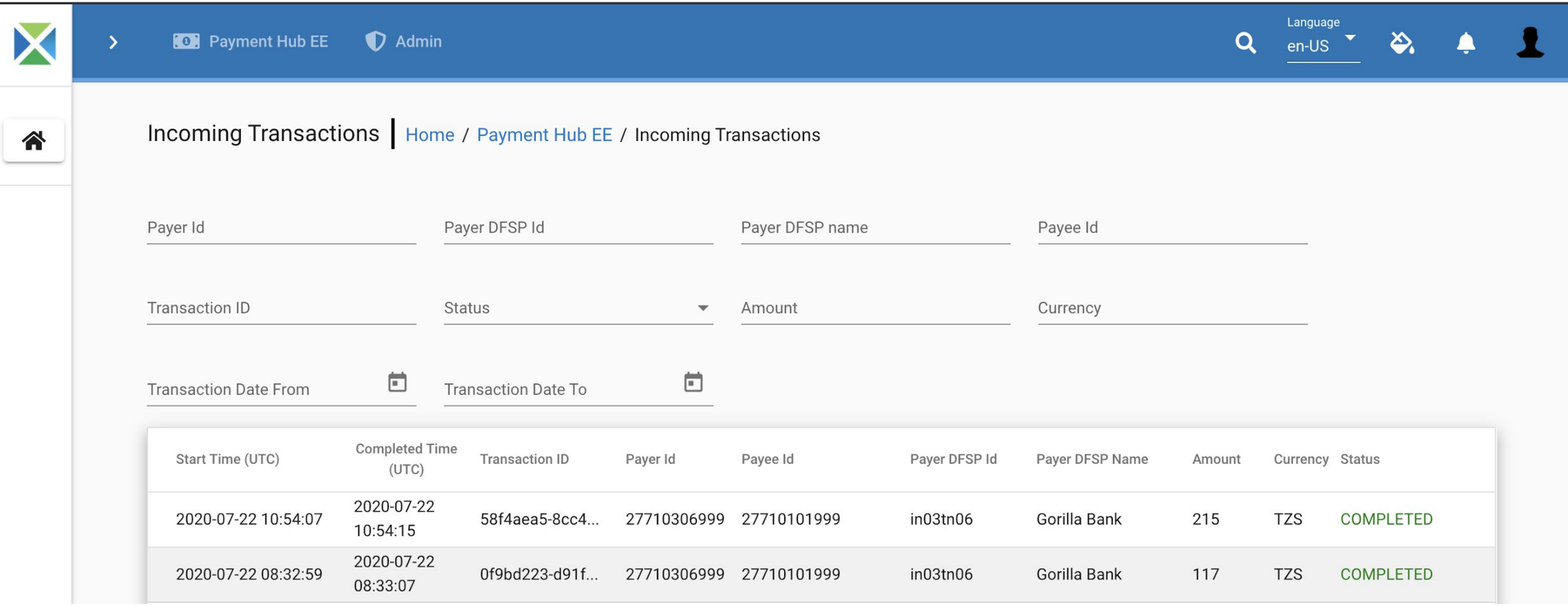
Refund capability for privileged operators on successful incoming transfers

After multiple automated attempts, transactions can be handed over to operations to retry or resolve manually

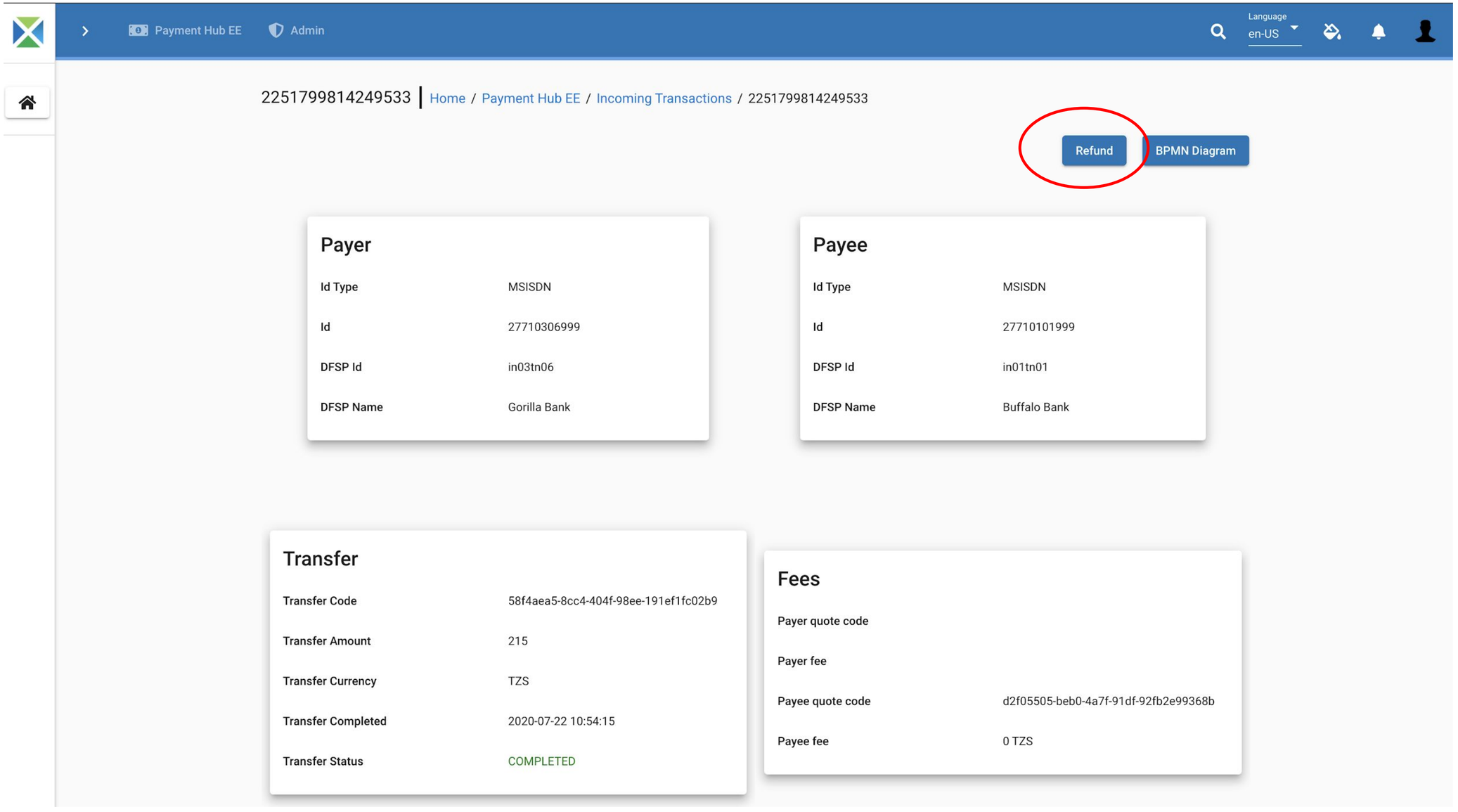
Search capability for the different flows



Incoming Payment at Payee



Incoming transfers with Refund capability



Outgoing Transfer - The Refund