



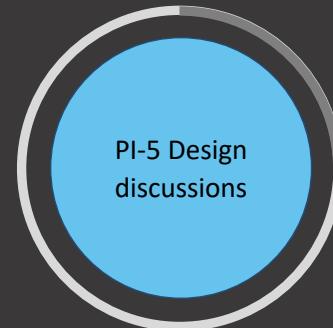
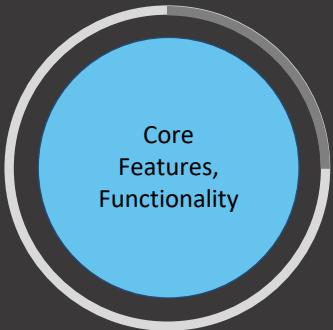
mojaloop

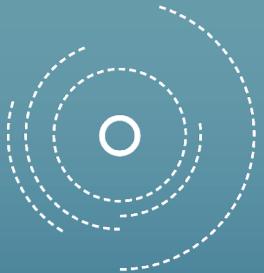
Mojaloop Phase-3 PI6

Supporting Adoption & Deployment

Mojaloop Phase3 PI-6

Supporting Adoption & Deployment





mojaloop

PI-5 Introduction and Summary

Supporting Adoption & Deployment

PI-6 Agenda - 1

1. Community Updates
 - a. Documentation
 - b. Adoption Updates
2. PI-5 Account Lookup Service (ALS) Update
 - a. ALS Design
 - b. ALS implementation

PI-6 Agenda - 2

3. PI-5 Core Features and Functionality updates

- a. Release Mechanism
- b. QA Framework
- c. Integration Tests
- d. Error end-points
- e. Node Upgrade
- f. Bug Fixes
- g. Validation Enhancements
- h. Community Support
- i. Code Enhancements
- j. Metrics Enhancements
- k. Deployment Updates

PI-6 Agenda - 3

4. Topic / Design discussions
 - a. Design Authority Proposal
 - b. Bulk Payments Design
 - c. Settlements Updates and Discussion (Day 2)
 - d. Fraud Workstream and Plans for the next PI (Day 2)
5. PI-6 RoadMap
 - a. Topics (Day 3)
 - b. Priorities (Day 3)

Mojaloop PIs Overview

| Timeline | Summary |
|----------|--|
| Phase-1 | <p>Level One Project</p> <ul style="list-style-type: none">• Reference Implementation• 6 Program Increments (PIs) (2016 - 17) |
| Phase-2 | <p>Road To Productionization: Phase-2 (2018)</p> <ul style="list-style-type: none">• PI – 1 (Feb - April)• PI - 2 (April - June)• PI - 3 (June - August)• PI – 3.5 (September)• PI – 4 (November-December 2018): Performance, Settlements, CEP, QA Framework, Operational Monitoring, Managed backlog for Phase-3 |
| Phase-3 | <p>Supporting Adoption & Deployment – (2019 Jan - June)</p> <ul style="list-style-type: none">• <i>PI-5 (Feb – mid-April): Cross-border/network, Account lookup, Comprehensive QA Framework, Streamlined CI, Release process, Support for error endpoints, Documentation, Node Upgrade, Bug Fixes & Community support, Bulk Transfers Design</i>• PI-6: Merchant Payments, Error-Event handling frameworks, PoC for Bulk payments, Gateway, Fraud & AML, Settlements API |

PI-6 Mojaloop OSS Community

Collaboration

- a. Documentation (Cross-Lake, Fintech-Inversiones, Modusbox)
- b. Quality Assurance (Mowali, Modusbox)
- c. Account Lookup Service (Mowali, Tips, Coil, Modusbox)
- d. Scrum-of-scrums (Wider community)
- e. Design Authority Channel

Switch Functionality – Mojaloop End-points (PI4)

Mojaloop v1.0 – API Specification

Transfers

- [●] POST - Prepare
- [●] PUT - Response
- [●] PUT – Error
- [●] Outgoing
- [●] Incoming
- [●] GET - Query

Parties

- [●] GET - Request
- [●] PUT - Response
- [○] PUT - Error

Quotes

- [●] POST - Request
- [●] PUT - Response
- [●] PUT - Error
- [○] GET - Query

Participants

- [●] POST - Create
- [●] PUT - Response
- [○] POST - Bulk Create
- [○] PUT - Error
- [○] DEL - Delete

Transactions

- [○] PUT - Response
- [○] GET - Query

TransactionRequests

- [○] POST - Request
- [○] PUT - Response
- [○] PUT - Error
- [○] GET - Query

Authorizations

- [○] GET - Request
- [○] PUT - Response
- [○] PUT - Error

BulkTransfers

- [○] POST - Request
- [○] PUT - Response
- [○] PUT - Error
- [○] GET - Query

BulkQuotes

- [○] POST - Request
- [○] PUT - Response
- [○] PUT - Error
- [○] GET - Query

Key

- [●] Fully implemented
- [●] Legacy Code
- [●] Partially implemented
- [●] Not implemented
- [○] Out of Scope for PI4

Switch Functionality – Mojaloop End-points (PI5)

Mojaloop v1.0 – API Specification

Transfers

- [●] POST - Prepare
- [●] PUT - Response
- [●] **PUT – Error**
- [●] Outgoing
- [●] **Incoming**
- [●] **GET - Query**

Parties

- [●] **GET - Request**
- [●] **PUT - Response**
- [●] **PUT - Error**

Quotes

- [●] POST - Request
- [●] PUT - Response
- [●] PUT - Error
- [○] GET - Query

Participants

- [●] **POST - Create**
- [●] **PUT - Response**
- [●] **POST - Bulk Create**
- [●] **PUT - Error**
- [●] **DEL - Delete**

Transactions

- [○] PUT - Response
- [○] GET - Query

TransactionRequests

- [○] POST - Request
- [○] PUT - Response
- [○] PUT - Error
- [○] GET - Query

● Interim:

- subId not supported currently
- no validations applied to “update” operations
- full design for Participants POST – Create is pending

Authorizations

- [○] GET - Request
- [○] PUT - Response
- [○] PUT - Error

BulkTransfers

- [○] POST - Request
- [○] PUT - Response
- [○] PUT - Error
- [○] GET - Query

BulkQuotes

- [○] POST - Request
- [○] PUT - Response
- [○] PUT - Error
- [○] GET - Query

Key

- [●] Fully implemented
- [●] Legacy Code
- [●] Partially implemented
- [●] Not implemented
- [○] Out of Scope for PI5



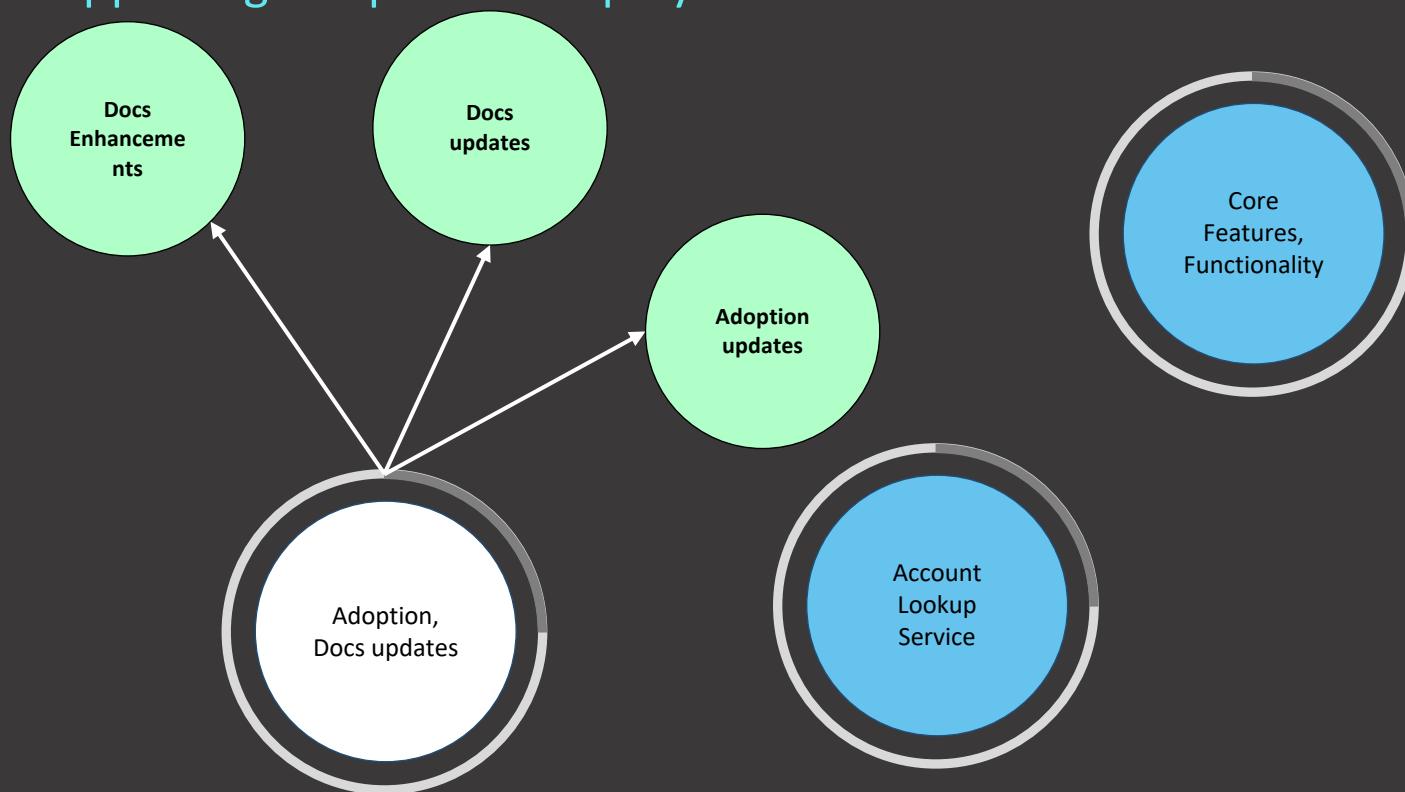
mojaloop

PI-5 Documentation, Adoption Updates

Supporting Adoption & Deployment

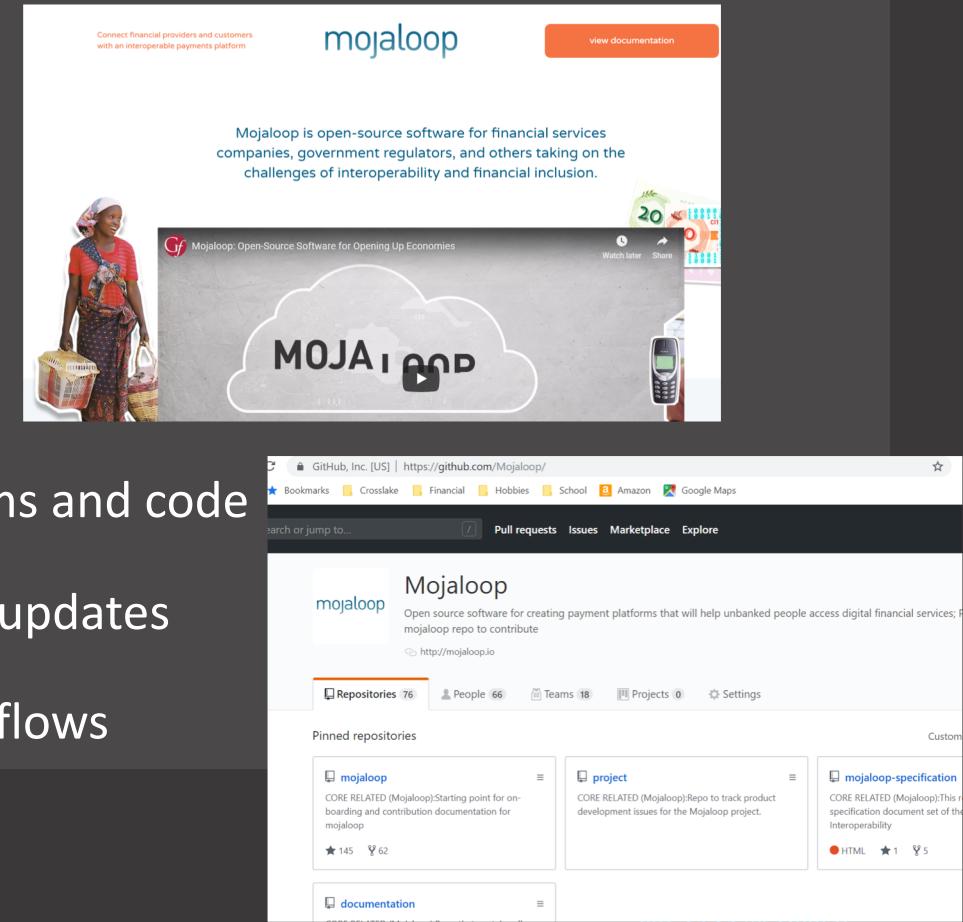
Mojaloop Phase3 PI-6

Supporting Adoption & Deployment



Documentation Overview

- Documentation and Onboarding is critical to any Open Source Project
- Mojaloop.io
 - Provides first entry point into the system
 - Overview of Use Cases – updated this PI
- GitHub
 - Contains project documentation, diagrams and code
 - Open to the public for consumption and updates
 - Challenge for organizing documentation flows





Documentation – GitBooks Overview

What is Gitbooks?

An open-source open documentation framework where teams can document everything from products, to APIs and internal knowledge-bases based on open-standards with community driven plugins.

Why Gitbooks?



Markdown

Lightweight markup language with plain text formatting syntax supporting standard HTML, and CSS.



Embed Generated Content

Embed generated sequence diagrams, openapi/swagger docs, etc.



Search

Find what you are looking for.



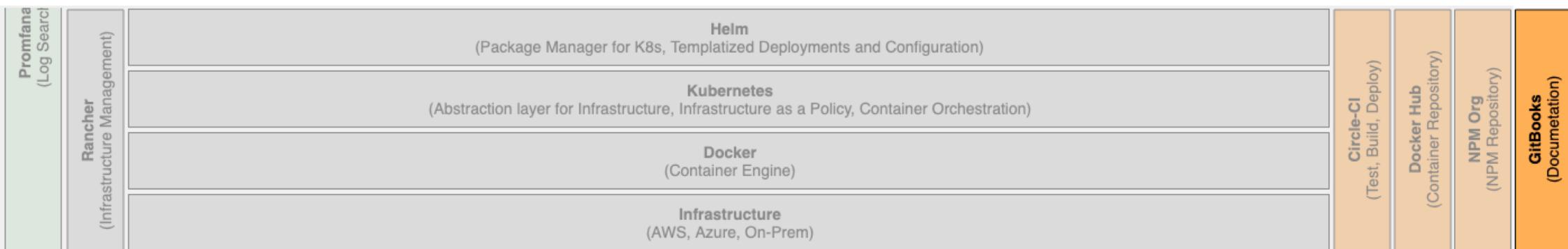
Plugins

Community plugins for generating content (e.g. plantuml, openapi/swagger docs), providing integration to Github, Slack, etc and themes (e.g. ToCs, Navigation, etc)



Cli

Gitbook-cli to build static-content, with support for local testing. Also supports auto-build sense when changes are made locally when testing.



Documentation - Mojaloop

Quickly submit content changes via Pull-Requests by the click of a button

Hosted through gh-pages



Search capability

The screenshot shows the Mojaloop documentation homepage. On the left, there's a sidebar with a search bar labeled "Type to search" and a collapsible navigation menu. The main content area features a heading "Mojaloop Overview" with a detailed description of the project. Below it is a sequence diagram titled "Mojaloop Hub" illustrating the interaction between Payer FSP, Payee FSP, and Central Services. A callout box points to the sequence diagram with the text "PlantUML are auto-generated as part of build process." Another callout box points to the "Central Services" section of the sequence diagram with the text "Editable SVG in Draw.io".

PI-6

April 2019

The screenshot shows a page with an auto-generated table of contents on the right side, listing various sections from "Account Lookup Service" to "ALS Oracle Design". Below the table of contents is a sequence diagram titled "7.1.0. Get Participant Details" under the "Account Lookup Service". The sequence diagram details the interaction between a "Financial Service Provider" (Payer FSP), an "Account Lookup Service (ALS)", and various DAOs (Config DAO, Oracle DAO, ALS Database, ALS Endpoint DAO). A callout box points to the sequence diagram with the text "Auto-generated Table of Contents based on page content headers."

loop. The purpose of this component is to support multiple Look-up registries (Oracles). familiar to the Central-Service API for the Component). The ALS will in all intense rules/config.

Documentation - CI/CD

PRs & Tagged releases trigger CI/CD Process

The screenshot shows the GitHub repository page for 'mojaloop / documentation'. A yellow dashed box highlights the 'Releases' tab, which is currently selected. Below it, the 'v5.5.3 Release' section is shown, containing details about the release, assets (Source code (zip) and Source code (tar.gz)), and a changelog. The changelog lists updates for the v5.5.2 release, including support for gitbook publishing, removed unused methods, added gh-pages branch, and updated .circleci/config.yml.

v5.5.3 Release

mdebarros released this 6 days ago · 7 commits to master since this release

Hotfix Hotfix for PlantUML Sequence Diagrams not being rendered.

Gitbook UML plugin is unable to generate valid SVGs. Changing this to PNG resolves the issue.

Will have to re-look at generating SVGs in future.

Assets 2

Source code (zip)

Source code (tar.gz)

v5.5.2 Release

mdebarros released this 6 days ago · 11 commits to master

- Updated .circleci/config.yml to support gitbook publishing
- Removed unused re-usable methods from .circleci/
- Added gh-pages branch to the ignore list in .circleci
- Added slack announcements
- Added *.jar to gitignore file
- Using standard node-alpine image

Also updated gh-pages branch to support this release:

slack
announcements on successful deployment.

Friday, April 5th

MojaBot APP 1:13 PM documentation - Release v5.5.3: <https://github.com/mojaloop/documentation/releases/tag/v5.5.3>

Workflows » mojaloop » documentation » v5.5.3 (tag) » de0e798b-f12e-4af9-bcb7-9d029df9936d

SUCCEEDED v5.5.3 / build_and_test

Rerun

3 jobs in this workflow

setup 00:46

build 01:36

deploy 00:28

PR's require setup/build to pass successfully. Future enhancements to include UI/Doc tests?

6 days ago

CI/CD Process will Publish and tagged releases to:

GitHub

Announcement to:

slack

Documentation: Wiki (Spanish)

Motivations (started on May 2018):

1. Sparse and deprecated documentation
2. Hard deployment process
3. Confusing concepts (context and translation)
4. New team members onboarding
5. Work logging

Logged in as: Adolfo Rios (arios) [Update Profile](#) [Admin](#) [Log Out](#)

Search

Recent Changes Media Manager Sitemap

Fintech Inversiones - Wiki

Trace: • marco_teórico • api_mojaloop • infra_indice • howto_kubernetes_cluster • repos • resumenes • archivo • start • arquitectura • introducción

introducción

Table of Contents

- Introducción a Mojaloop
 - Fundamentos del proyecto
 - Principios fundamentales del Proyecto Level One
 - Arquitectura de la solución
 - Hub Mojaloop
 - DFSP
 - Banco de Conciliaciones
 - Fines prácticos de la disposición de los códigos fuente
 - Potenciales utilidades para los usuarios

Introducción a Mojaloop

Mojaloop es una plataforma open-source para compañías de servicios financieros y entes reguladores gubernamentales, que afronta los desafíos de interoperabilidad e inclusión financiera. Mojaloop es un proyecto impulsado por la **Fundación de Bill y Melinda Gates (BMGF)**, bajo su programa de Servicios Financieros para los Pobres (*Financial Services for the Poor*). La Fundación demarcó los requerimientos de la plataforma y encargó el desarrollo de las funcionalidades básicas a una serie de empresas privadas como ModusBox, Crosslake Technologies, Ripple y otras.

Con esta plataforma se busca que los clientes puedan efectuar pagos digitales entre sí, independientemente al tipo de servicio o cuenta que utilicen. Por ejemplo, cuentas de bancos, billetes móviles, servicios de giros, etc. A estos últimos se denomina **DFSP (Digital Financial Service Provider)**. Se debe poder utilizar los servicios de la plataforma a través de las tecnologías digitales y móviles básicas, de tal manera que faciliten el acceso a las personas que son marginadas de la dinámica del ecosistema financiero al no contar disponer de cuentas bancarias y al costo alto que representan los procesos de transferencia de dinero.

El stack de Mojaloop es amplio y utiliza tecnologías varias, tales como *Node.js*, *Apache Kafka*, *Java*, *MySQL*, entre otras. La arquitectura es orientada a microservicios empaquetados en *Docker* que pueden ser deployados indistintamente en ambientes on-premise o cloud (AWS, Azure, Google Cloud).

Principales links de interés

| Contenido | URL |
|----------------------------|---|
| Sitio oficial | http://mojaloop.io |
| Documentación del proyecto | https://docs.mojaloop.live/ |
| Repositorio en Github | https://github.com/mojaloop |

Edit

Fundamentos del proyecto

El proyecto Mojaloop inicialmente nació con el nombre de Level One Project, que luego pasó a constituirse en un marco teórico formal de principios deseables para un esquema de pagos interoperables, dando paso a Mojaloop como nombre propio del proyecto y de la marca. Es necesario llevar en cuenta estos principios para comprender ciertas decisiones de diseño y de arquitectura, además de considerar que cualquier eventual

Documentation: Wiki (Spanish)

Structure

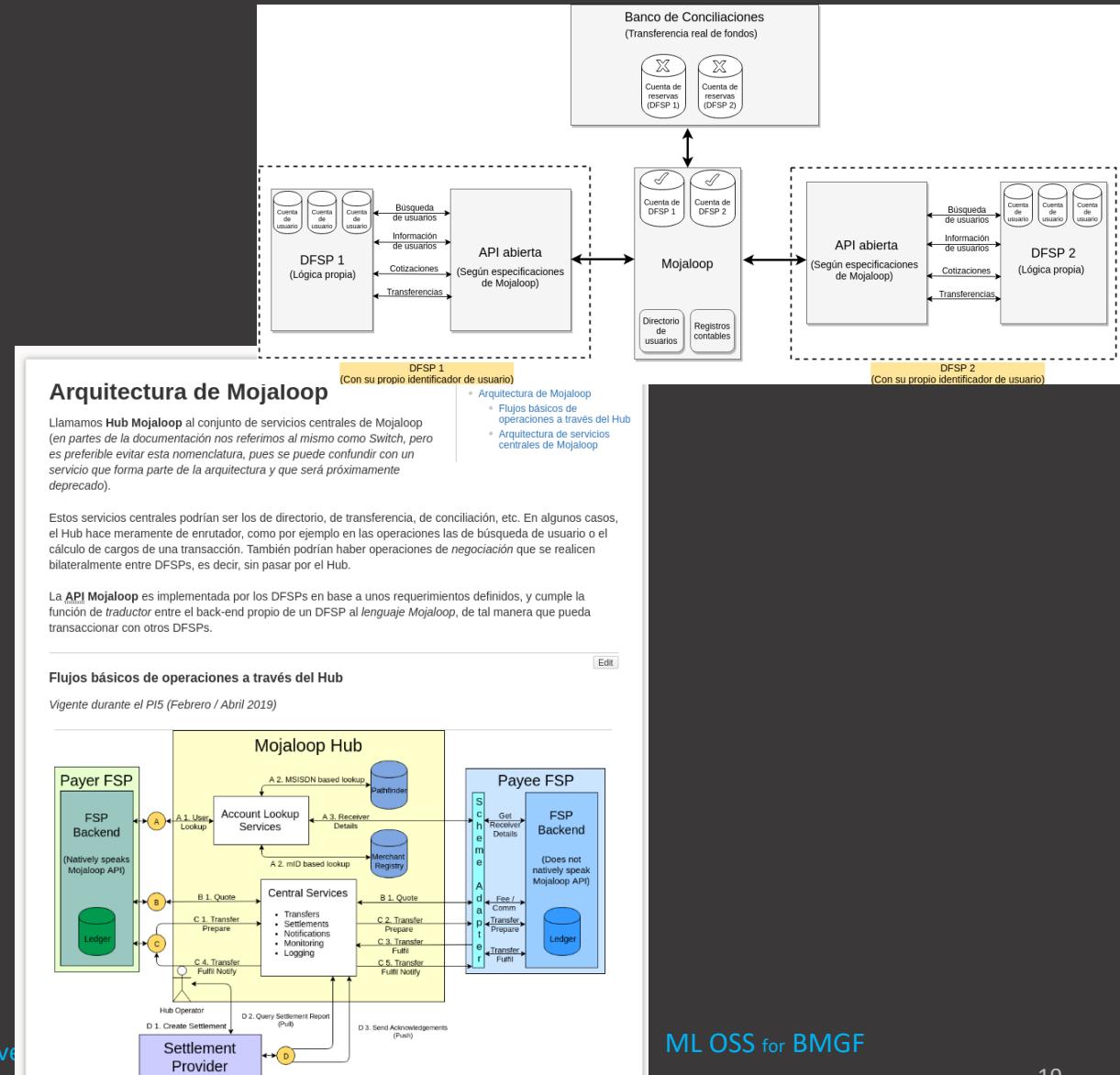
1. Introduction (Onboarding)
2. Infrastructure/DevOps
3. Specific topics
4. Related business topics

1. Start
2. Theoretical framework
 - a. Introduction
 - b. Use cases
 - c. Architecture
 - d. Scheme rules
 - e. Interledger Protocol
 - f. ILP usage
 - g. Glossary of terms
3. Infra / DevOps
4. Mojaloop API
5. Settlements
6. Cross-border & cross-network transactions
7. Documentación interna
 - a. Code repos
 - b. Overall project status
8. Archive

Documentation: Wiki (Spanish)

Onboarding

1. Level One Principles
2. Use cases
3. Architecture (main services + Open API)
4. Basic concepts
 - a. Hub, scheme, DFSP...
 - b. Settlement, position, NDC...
 - c. Interledger, quote, fulfilment...



Documentation: Wiki (Spanish)

Infrastructure/DevOps

1. Kubernetes, Rancher, Docker, Helm
2. Environment setup
 - a. Single-node (Minikube) vs multiple nodes
 - b. Specs (Memory, CPU cores, disk space)
3. Hacks
 - a. K8s, Docker, Helm commands...
 - b. Monitoring

Inicios de 2019

A inicios de 2019 se agrega otro servidor sobre un hipervisor VMware. El objetivo del servidor es contar con recursos suficientes para realización de pruebas de conceptos. Las características del servidor son las siguientes:

| | |
|------------|--------------|
| Procesador | 12 cores |
| HDD | 500 GB |
| RAM | 32 GB |
| OS | CentOS 7 |
| IP | 192.168.0.72 |

— Características de servidor de testing (2019)

Sobre el segundo servidor, también se utiliza virtualización anidada para segmentar los recursos internos disponibles para diferentes objetivos. El hipervisor utilizado para el efecto es QEMU-KVM. El primer servidor que es creado con el fin de disponibilizar una zona DNS para el proyecto y además disponibilizar un repositorio. Este servidor cuenta con:

| | |
|------------|---------------|
| Procesador | 2 cores |
| HDD | 50 GB |
| RAM | 8 GB |
| OS | CentOS 7 |
| IP | 192.168.0.144 |

— Características de servidor DNS/Repositorio

El repositorio Gitlab

Posteriormente, se implementa mediante Docker un servidor Gitlab para almacenar configuraciones dentro del servidor.

VMware VM

KVM VM 01

KVM VM 02

VirtualBox Hypervisor
5.2.26r128414

vboxnet0: 192.168.99.1/24

Minikube Linux

docker0: 172.17.0.1/16

eth0: 10.0.2.15/24

eth1: 192.168.99.100/24

prod.finproyecto
CentOS 7

eth0: 192.168.0.144/24

server01.finproyecto
CentOS 7

eth0: 192.168.0.145/24

KVM Hypervisor
15.3

rancher.finproyecto
CentOS 7

br0: 192.168.0.72/24

Docker Engine CE
18.09.2

Rancher UI Docker Container

Rancher Kubernetes Cluster

Kube-system

MojaLoop

Kubernetes Namespaces

VMware Hypervisor
(Versión XY)

ML OSS for BMGF

Documentation: Wiki (Spanish)

Specific topics

1. Open API
 - a. Lookup, quote, transfer, settlement
 - b. Basic P2P flow + postman collection
2. Settlements
 - a. Theory + models (Glenbrook research)
3. Cross-border
 - a. Proposal
4. Security

Servicios y recursos (endpoints) de la API

| Recurso | Servicio | Descripción |
|----------------------|---|--|
| /participants | Búsqueda de un participante | Determinar en qué FSP está ubicada la contraparte de una transacción financiera. |
| /parties | Obtener información sobre un participante | Obtener información sobre la contraparte de una transacción financiera. |
| /quotes | Cotizar una transacción | Calcular todas las partes de una transacción que modificarán el monto de la transacción (cargos y comisiones). Este recurso se utiliza para cotizar una única transacción, es decir, de un Emisor a un Receptor. |
| /transfers | Realizar una transferencia | Ejecutar propiamente una transacción financiera transfiriendo los fondos electrónicos del Emisor al Receptor, posiblemente a través de sistemas intermediarios. Este recurso se utiliza para realizar una única transacción, es decir, de un Emisor a un Receptor. |
| /transactionRequests | Solicitar una transacción | El Receptor solicita una transferencia al Emisor. El Emisor puede aprobar o rechazar esta transacción. Una aprobación de la petición, iniciará la transacción financiera propiamente. |
| authorizations | Verificar autorización | Solicitar al Emisor las credenciales necesarias cuando el mismo inició la transacción desde algún POS, ATM u otro dispositivo en el sistema del FSP Receptor. |
| transactions | Consultar transacción | Obtener información relacionada a la transacción financiera; por ejemplo, un token creado en una transacción exitosa. |
| bulkQuotes | Cotizar una transacción | Análoga al recurso /quotes, pero utilizada para realizar la operación en volumen, es decir, de un Emisor a varios Receptores. |
| bulkTransfers | Realizar una transferencia | Análoga al recurso /transfers, pero utilizada para realizar la operación en volumen, es decir, de un Emisor a varios Receptores. |

Documentation: Spanish Wiki (Spanish)

Team effort

1. Challenges faced were shared on the PI4 meeting
2. Documentation team (discussion on structure, topics, roles)
3. Ongoing English translation

Results from the OSS Team effort

1. Deployment guide (PI4)
2. Gitbook (PI5)

Available for the community: <https://wiki.fintechinversiones.com.py>

Where is Mojaloop being Deployed?

- **Tanzania**: A project sponsored by the [Central Bank](#). This is a merchant and P2P payments solution between banks and mobile money operators, enabling a massive growth in payments beyond what's been achieved on bilateral system integrations.
- **Mowali** : A pan-African payment solution involving Orange and [MTN](#) but also anticipating other players.
- **Others** that are confidential to those companies.



mojaloop

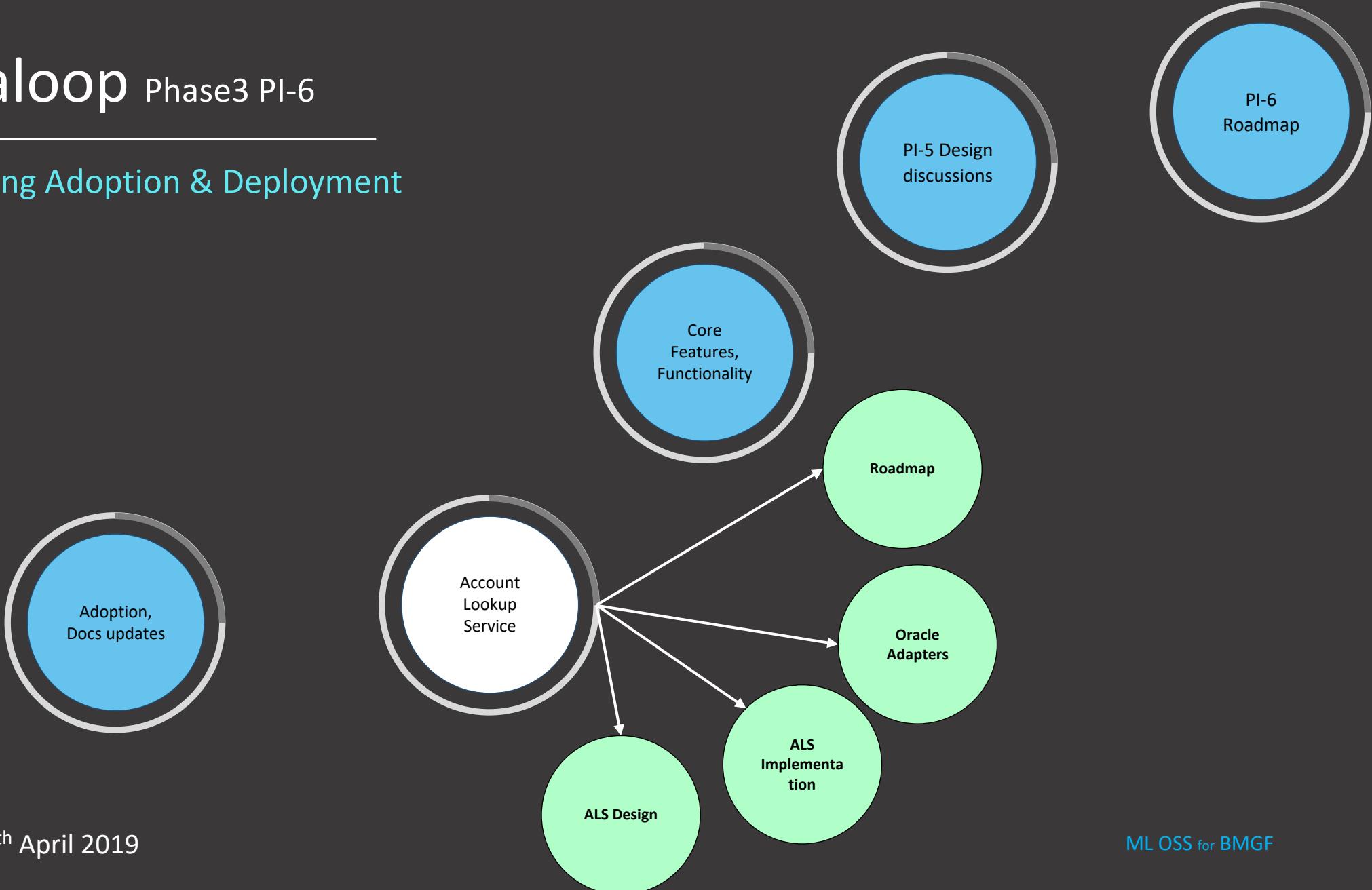
PI-5 Account Lookup Service

Supporting Adoption & Deployment

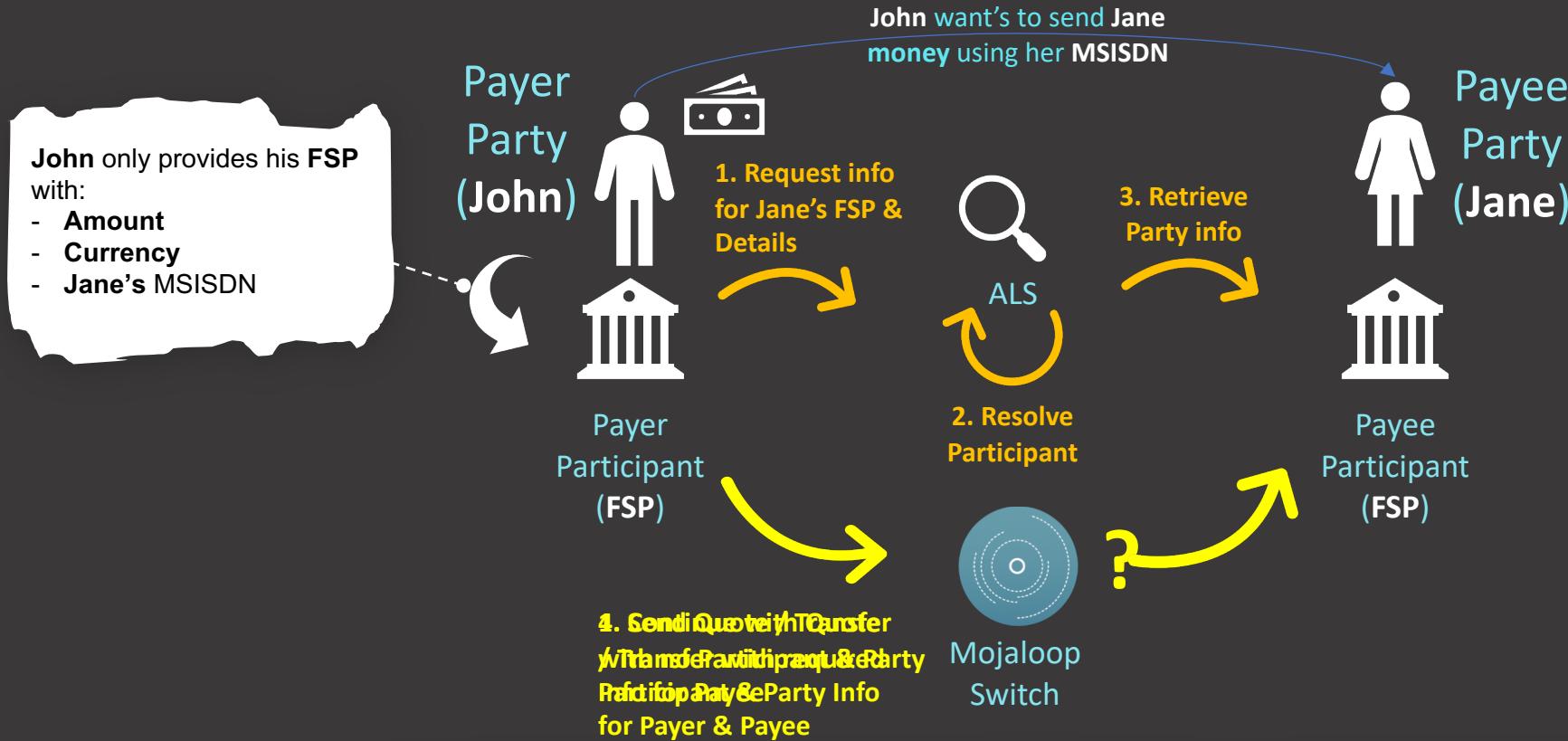


Mojaloop Phase3 PI-6

Supporting Adoption & Deployment



ALS: Why do we need the ALS (Account Lookup Service)?



API Definition

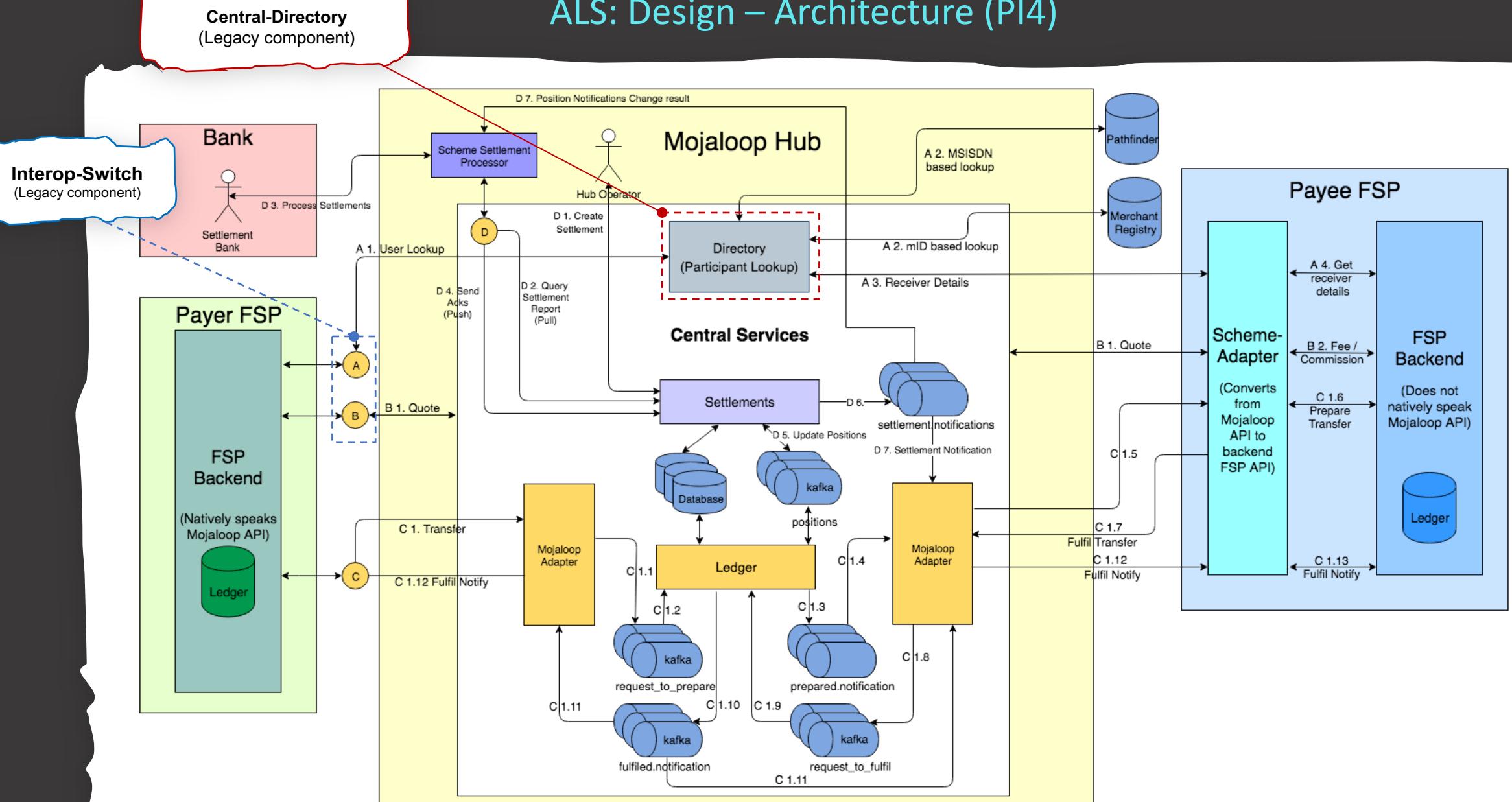
Open API for FSP Interoperability Specification

6.1 High Level API Services

On a high level, the API can be used to perform the following actions:

- **Lookup Participant Information** – Find out in which FSP the counterparty in a financial transaction is located.
 - Use the services provided by the API resource [/participants](#).
- **Lookup Party Information** – Get information about the counterparty in a financial transaction.
 - Use the services provided by the API resource [/parties](#).

ALS: Design – Architecture (PI4)

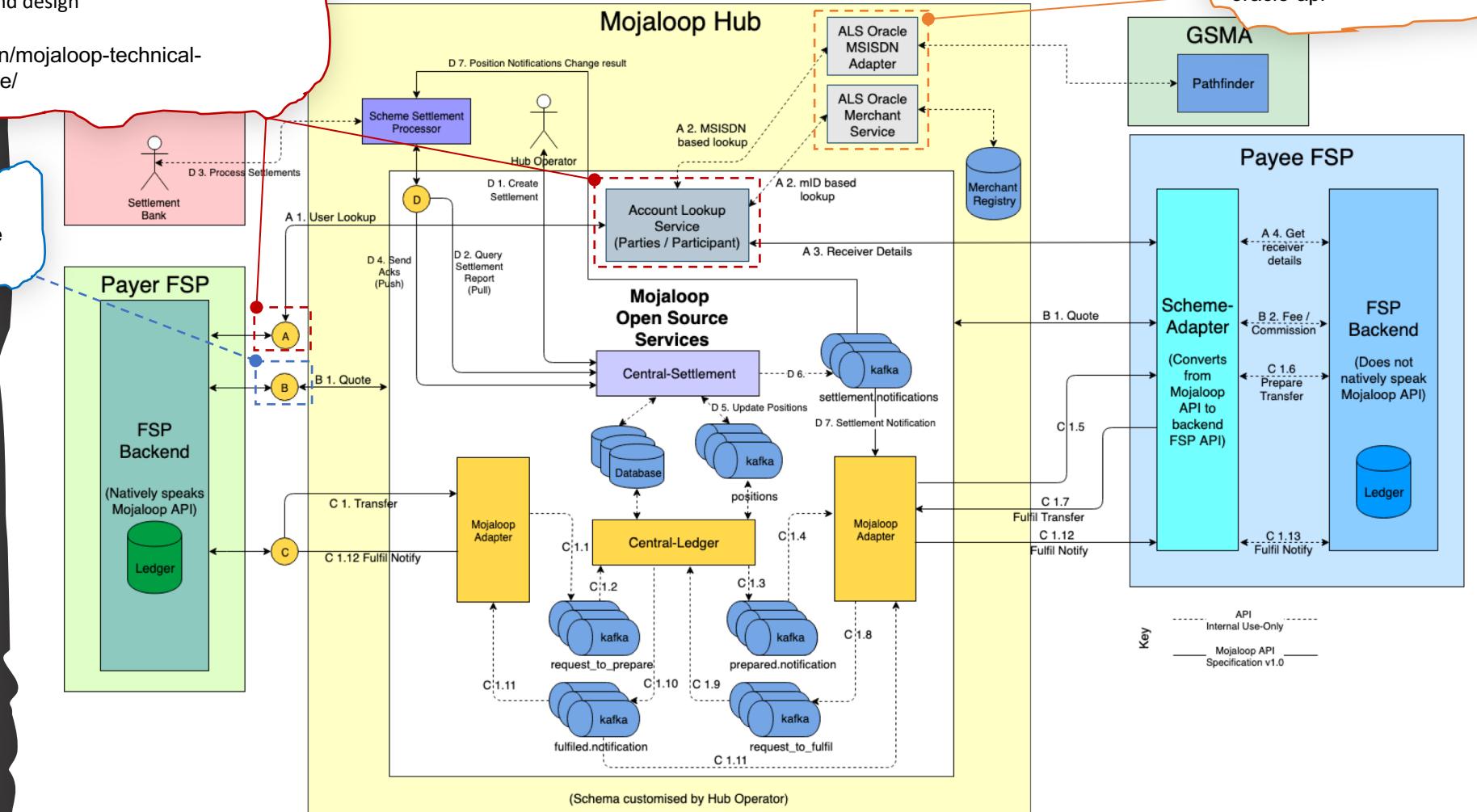


ALS: Design – Architecture (Current PI5)

- Deprecated Central-Directory
- Deprecated Interop-Switch for Participants/Parties
- **Account Lookup Service (ALS) design and implementation**
- ALS code-base contributed from **Mowali**.
- **Re-factored OSS standards and design**

<http://mojaloop.io/documentation/mojaloop-technical-overview/account-lookup-service/>

Legacy Interop-Switch
continues to handle Quote operations



Oracle Adapter/Service:

- Standard interface contract (openapi/swagger)

<http://mojaloop.io/documentation/api/#als-oracle-api>

ALS Mojaloop API operations

Parties

- [●] GET - Request
- [●] PUT - Response
- [●] PUT - Error
- [●] SubId support

Participants

- [●] GET - Request
- [●] POST - Create
- [●] POST - Create (Bulk)
- [●] PUT - Response
- [●] PUT - Error
- [●] DEL - Delete
- [●] SubId support

ALS Admin API operations

Oracles

- [●] POST – Create
- [●] GET - Request
- [●] PUT - Update
- [●] Delete - Remove

ALS: Design - Overview

Key

- [●] Fully implemented
- [●] Legacy Code
- [●] Partially implemented
- [●] Not implemented
- [○] Out of Scope

Design Considerations

Account Lookup Service (ALS)

- Pluggable Architecture to support multiple Participant Registries
- Scalable Architecture
- Future considerations:
 - Support running a single ALS with multiple Switches
 - Leverage on routing mechanisms defined by Cross-Border/Network design for Switch routing
- Implements Participants operations for Mojaloop API specifications v1.0
- Implements Parties operations for Mojaloop API specifications v1.0

Oracle Service/Adapter

- Standard interface/contract based on Mojaloop API specifications v1.0
- Support resolution of FSPIOP-Destination for FSPs via the ALS
- Synchronous API that follows traditional REST Paradigms
- Accelerator (Template) to expedite development
- Community Contributions by TIPS Team:
 - ALS Oracle Template - <https://github.com/mojaloop/als-oracle-template>

● Interim:

- no validations applied to “update” operations
- full design for Participants POST – Create is pending

Oracle API operations

Participants

- [○] GET - Request
- [○] POST – Create
- [○] PUT - Update
- [○] Delete - Remove

ALS: Design – Participants

API Mojaloop Specifications:

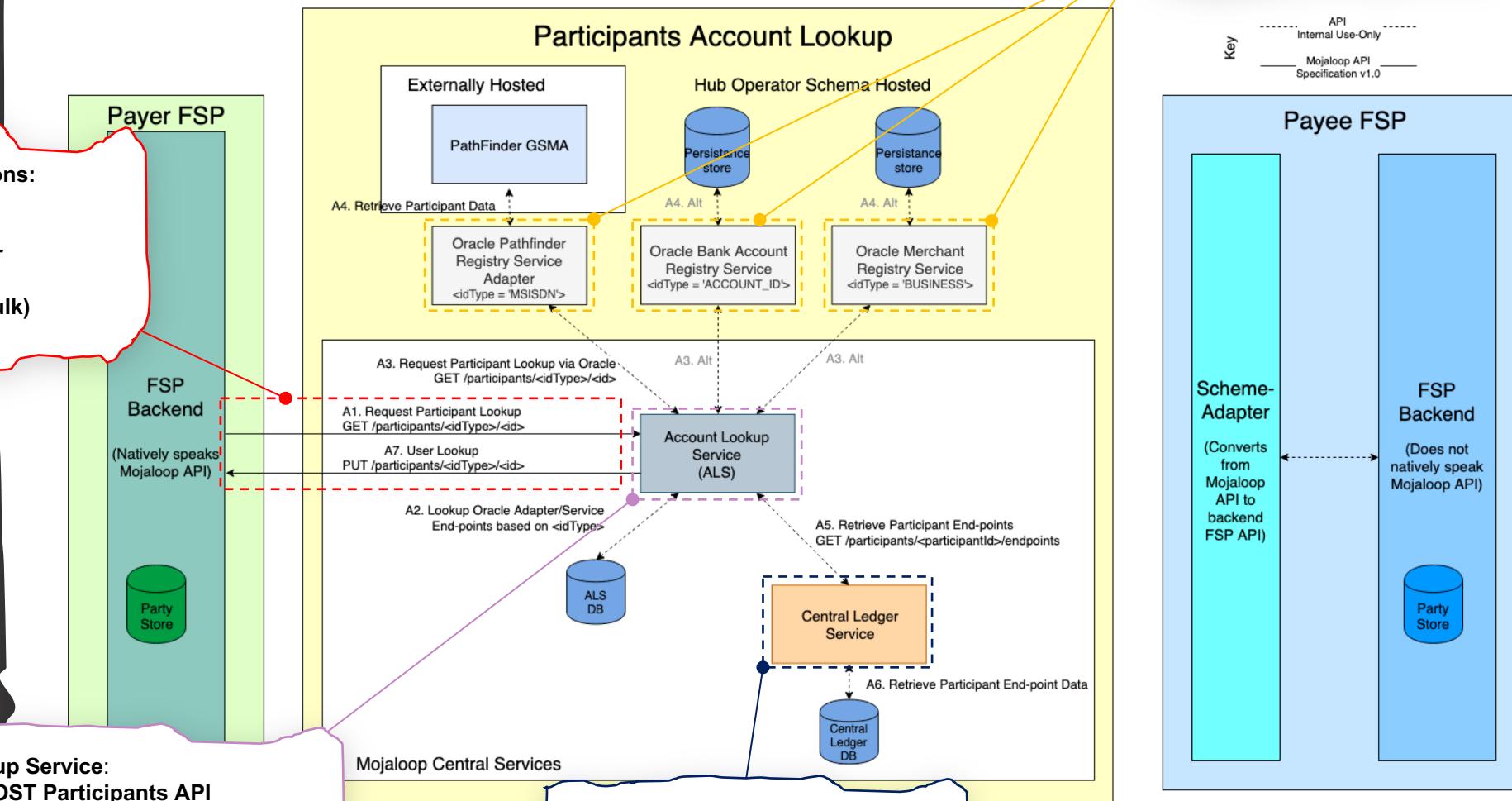
- GET /Participants
- PUT /Participants
- PUT /Participants/error
- POST /Participants
- POST /Participants (Bulk)
- DEL /Participants

Account Lookup Service:

- GET/PUT/POST Participants API
- Routing to Oracles
- Callback resolution for FSP responses

Oracle Adapter/Service:

- Registries to resolve Participant requests via an easy pluggable sync API.



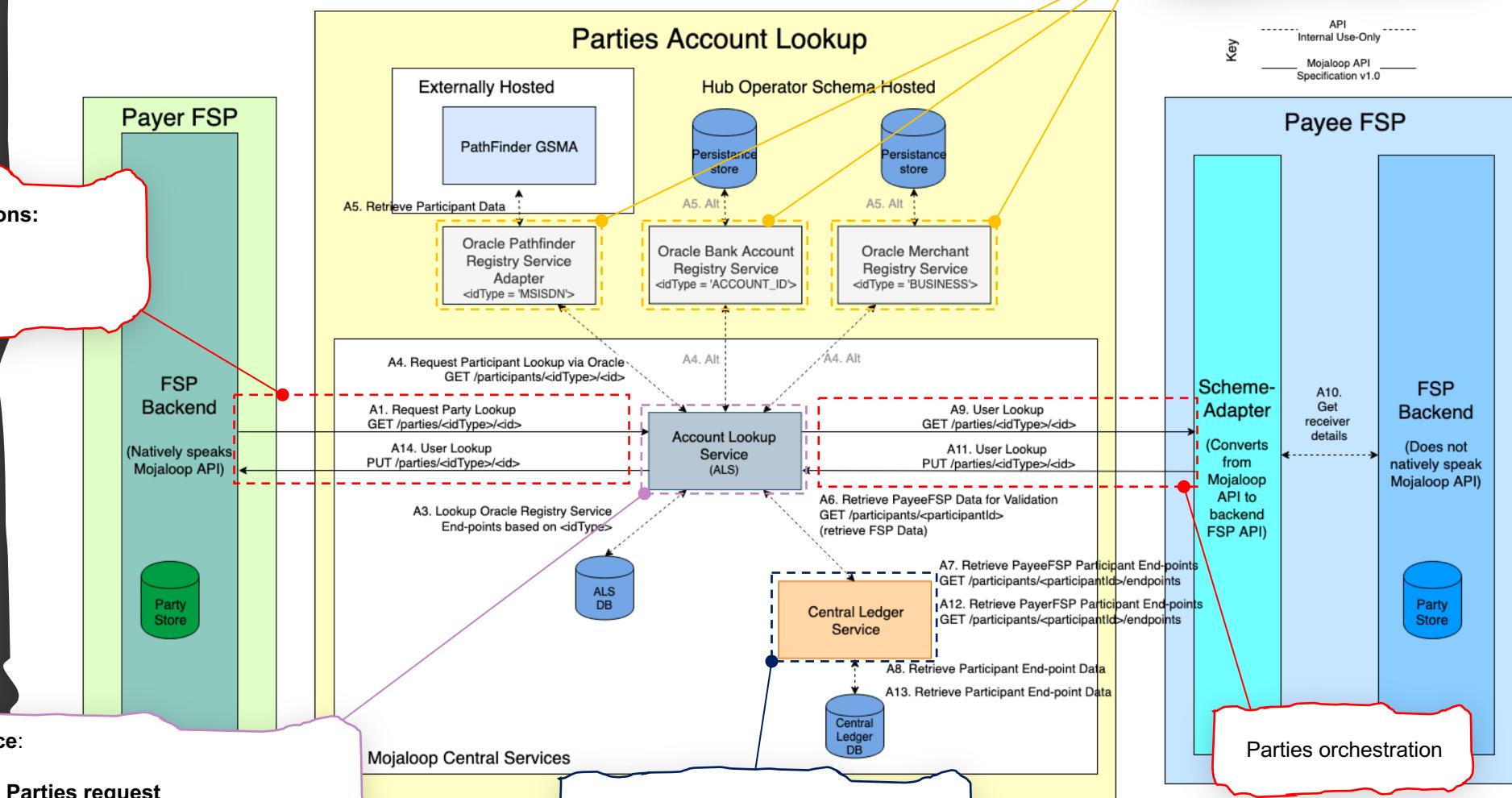
ALS: Design – Parties

Oracle Adapter/Service:
 - **Registries to resolve Participant requests** via an easy pluggable sync API.

API Mojaloop Specifications:
 - GET /Parties
 - PUT /Parties
 - PUT /Parties /error

Account Lookup Service:
 - GET/PUT Parties
 - Orchestration of the Parties request
 - routing to Oracles
 - Callback resolution for FSP responses.

Operational Admin API
 - resolve FSP Request End-points
 - resolve FSP Callback End-points



ALS: Roadmap

Account Lookup Service (ALS)

- Functional
 - Support for SubId on [Participants](#) operations
 - Support for SubId on [Parties](#) operations
 - Support to run [ALS](#) as a Central service with [multiple Switches](#)
 - Leverage routing mechanisms defined by Cross-Border/Network design for Switch routing
- Non-Functional
 - Expand on Unit [Test coverage](#)
 - [Integration](#) tests
 - [QA Framework](#)
 - [Performance](#) Tests

Oracle Service/Adapter

- Registry for [MSISDN](#) via [Pathfinder](#) (UDP) leveraging existing Mock-Pathfinder implementation
- Enhance [Oracle Template Accelerator](#) to provide functionality to [auto-generate a base project](#)

Simulator

- Enhance to provide [Participants simulation](#)
- Enhance to provide [Parties simulation](#)

ALS: Bank Account Oracle

Bank Account Oracle (BAO)

- Functional
 - Identify Bank Participant
 - Uses standardized account numbers
 - Account Number format (14 digits)
 - 3 digits: Participant Identifier
 - 1 digit: Currency Identifier
 - 10 digits: Party Account Identifier
 - Participant registry is maintained (based on the 3 digits participant identifier)
 - Support for one currency (TZS)

Operations Supported

Participants

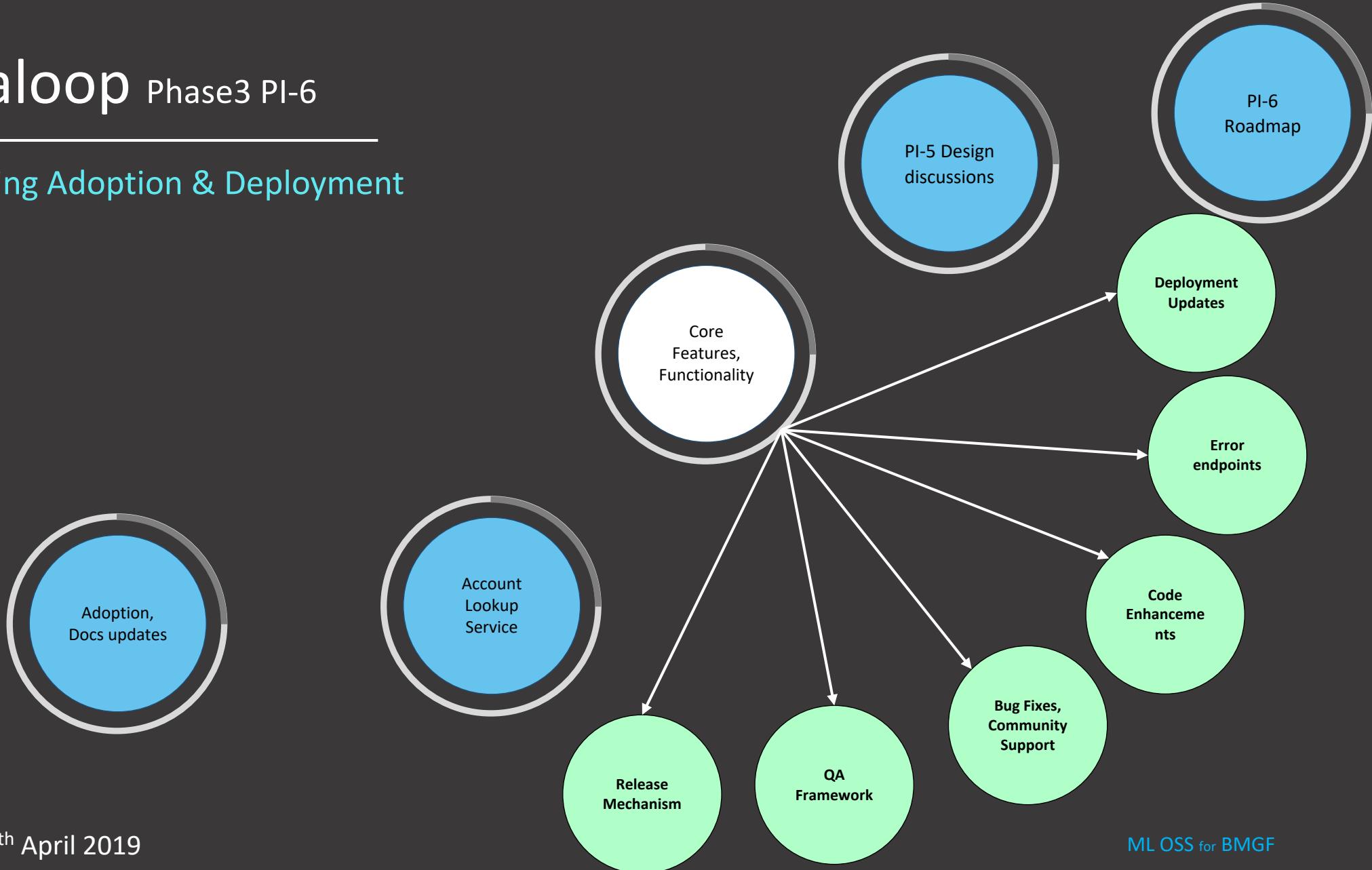
- [●] GET - Request
- [●] POST - Create
- [●] POST - Create (Bulk)
- [●] PUT - Update
- [●] DELETE - Delete

ALS: Implementation

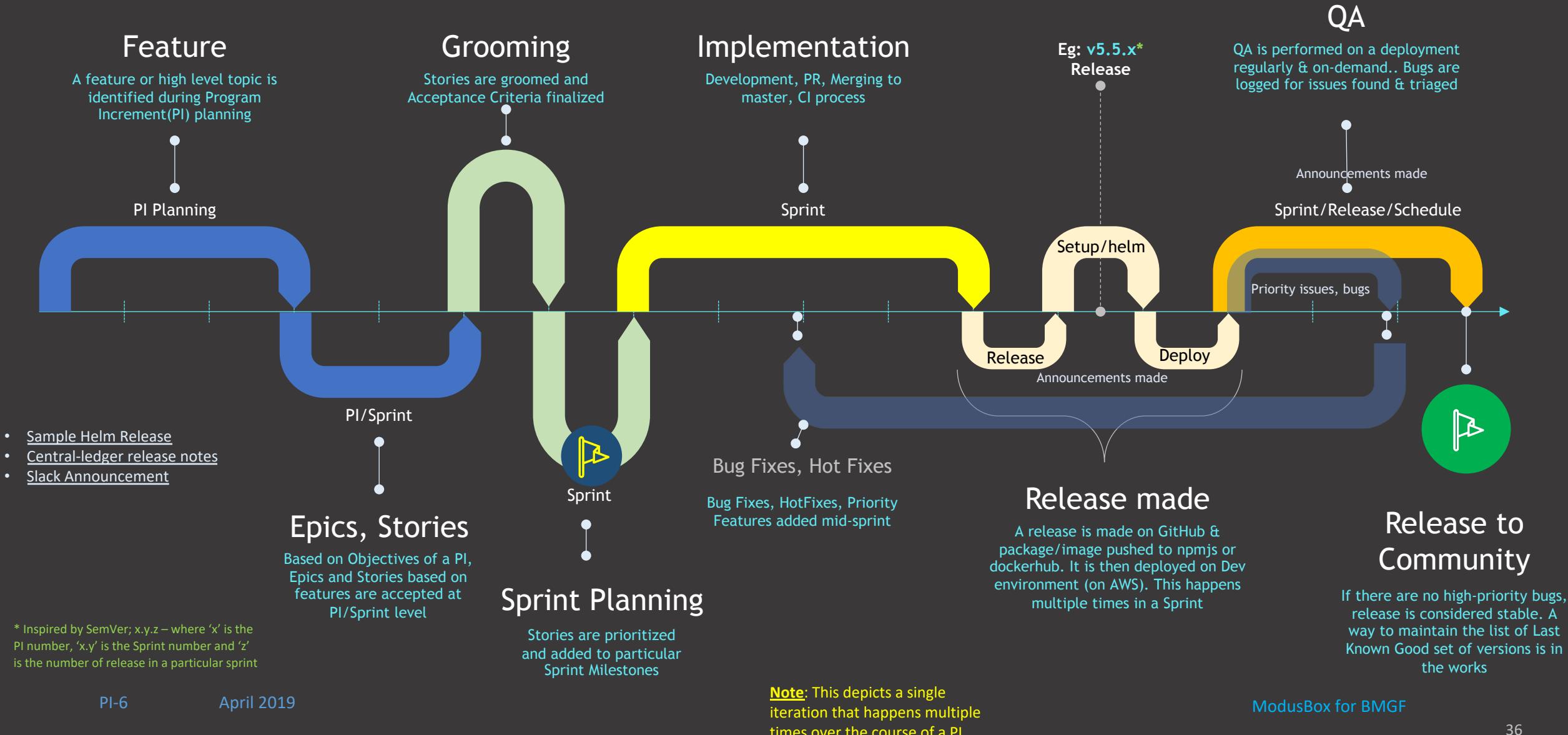
Demo

Mojaloop Phase3 PI-6

Supporting Adoption & Deployment



ML OSS: Release Mechanism



QA Framework: Regression Testing

is not

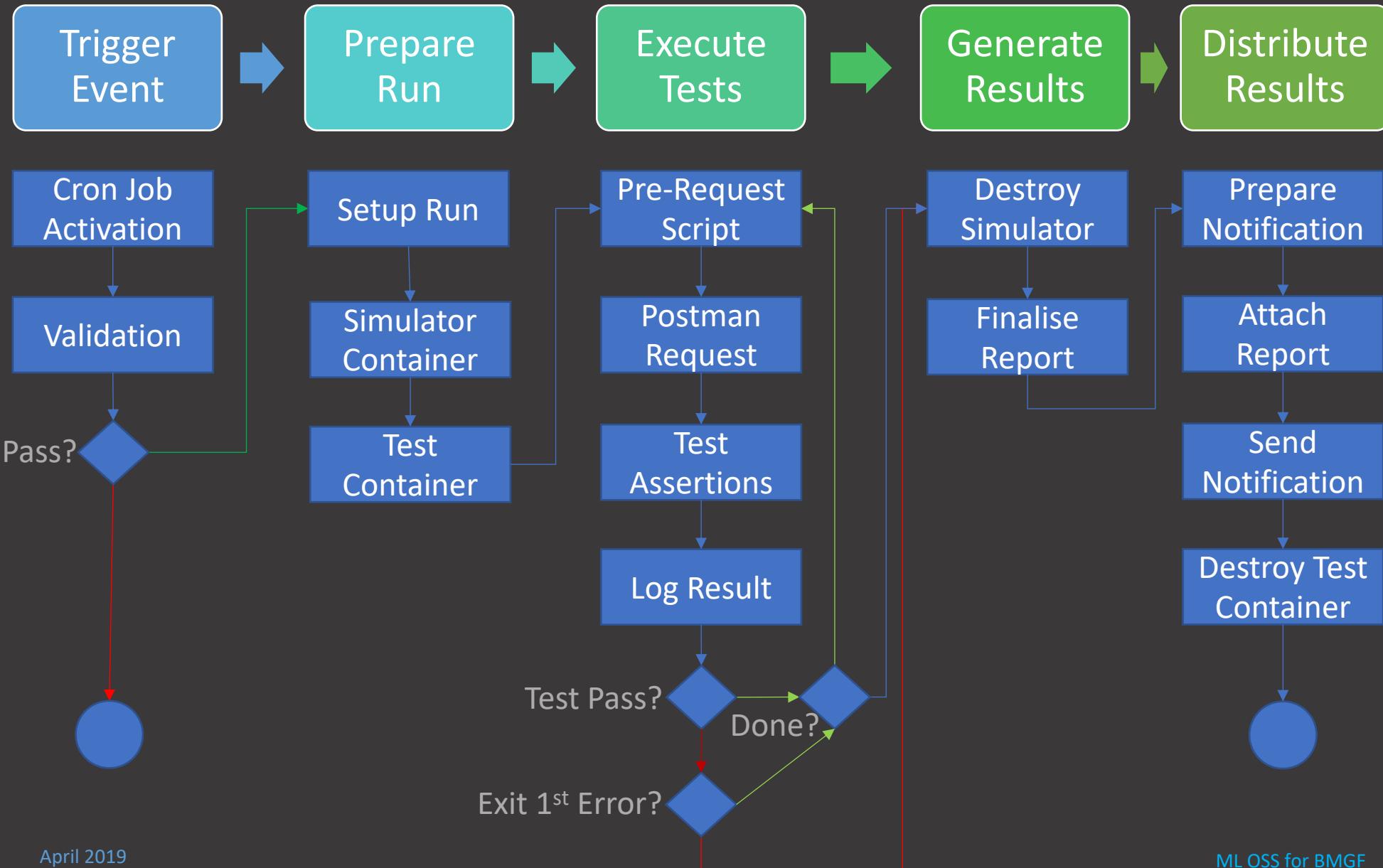
- 1) Unit testing
- 2) Coverage Testing
- 3) Integration Testing
- 4) Specific Functionality (use case) testing

QA Framework: Regression Testing

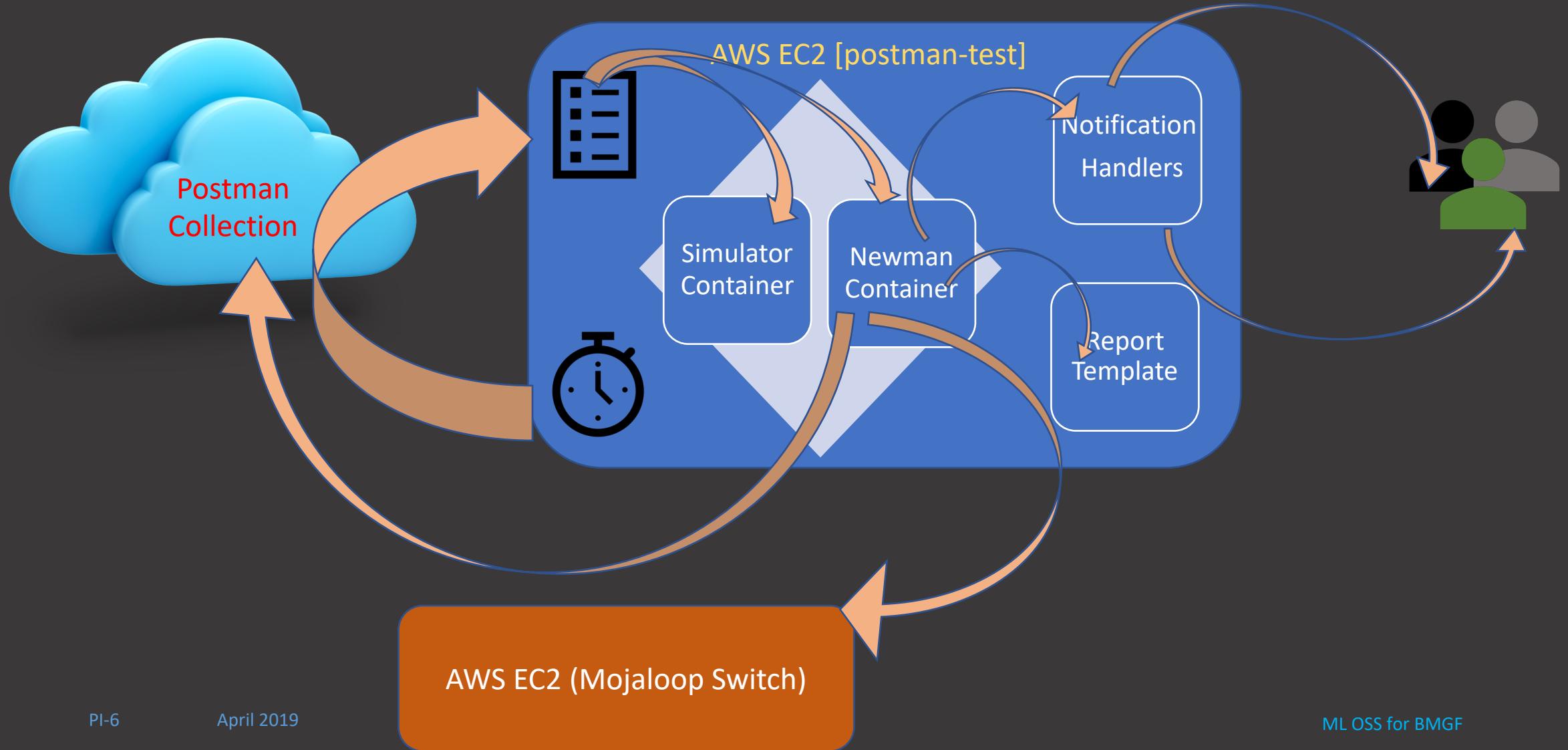
is typically characterized by

- 1) Ensuring a specified deployed environment performs and produces consistent results
- 2) Manual or automatic triggering by linking it with an event
- 3) Execute as many times as required – idempotent
- 4) Automatically sends out alerts in case of any failure or unexpected result
- 5) Communicated results can be used to pinpoint anomalies
- 6) Executes the entire range of functionality of the system under test (FR / NFR)

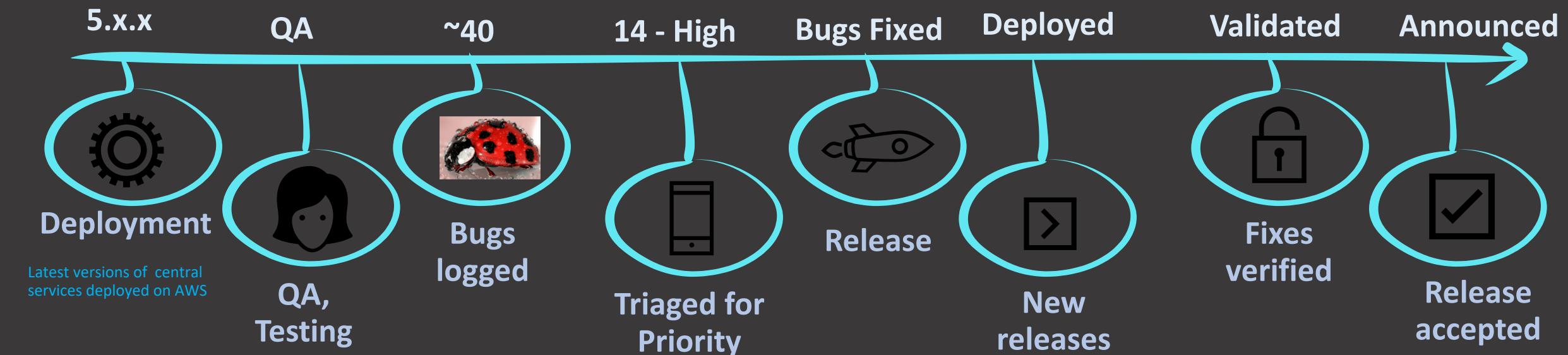
Regression Testing – Instance Life-Cycle



Regression Testing Interaction



Bug Fixes, Community Support



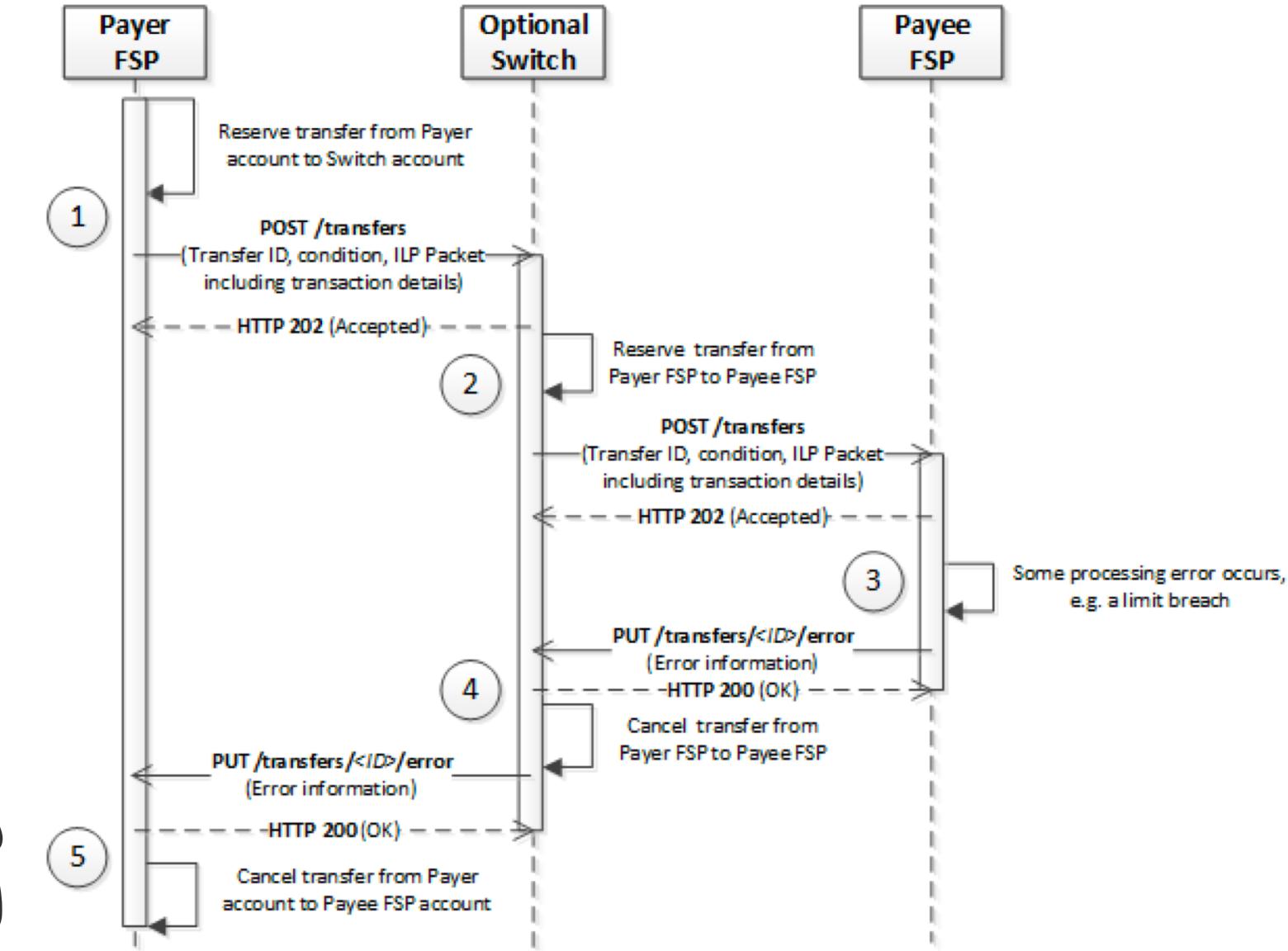
Community Support

1. Slack *#general* channel
2. Support to current adopters, implementers & contributors [Bug Fixes, Priority Features]
3. Documentation updates
4. Addressing Specification questions
5. Addressing Deployment questions
6. Adding of FAQs Section

Error end-points on the Switch

1. Generic handling of error callbacks on /transfers
2. Scenarios
 1. Payee sending PUT /error callback for a transfer
 2. Using 'ABORTED' as transferState in a fulfilment
3. Key design decisions
4. Clarify usage of 'ABORTED' state by Payee in a Fulfil request
5. Demo

PI-5: Handling of error callback for transfers



Integration tests

1. PI-4 Central-Ledger integration test processing time: > 10m
 1. The wait for callback result implemented via a While – Timeout loop
2. PI-5 Central-Ledger integration test processing time: < 5m
 1. Implemented `async-retry` library to handle the wait for callback result which allows for:
 1. Retry count limit
 2. Min timeout
 3. Max timeout

Future enhancements:

- Apply similar methodology to Integration Tests on ML-API-Adapter

Validation on incoming requests

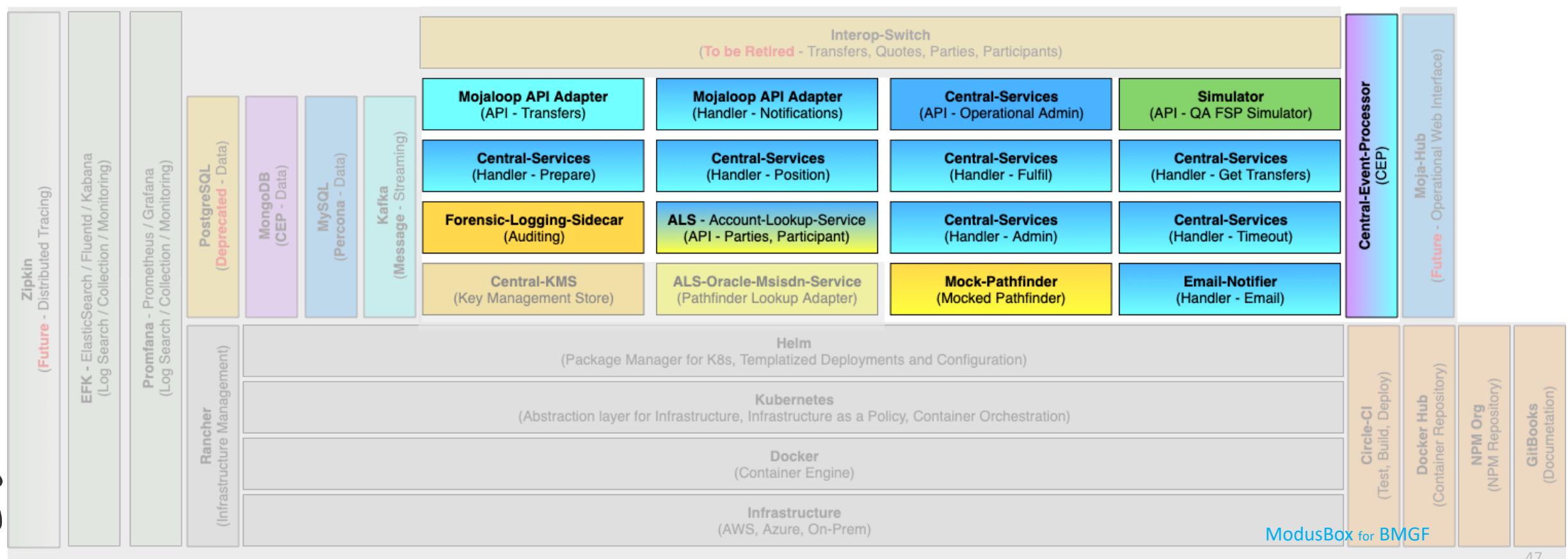
1. FSPIOP-Source validation
 1. Validate source FSP exists and is valid
 2. Validate Source FSP is a participant of an existing Transfers (GET – Request / PUT - Callback)
 3. Validate Source FPS is the Party owner for Participant Delete operations
2. FSPIOP-Destination
 1. Validate Destination FSP exists and is valid
 2. Validate Destination FSP is a Participant of an the Transfers (POST - Create/ PUT - Callback)

Node upgrade

1. All **Node Central Components** have been upgraded from **Node v8.x** to the latest **LTS v10.x**
 1. **Security** *fixes
 2. **Performance** *improvements ([http](http://http://nodejs.org/en/blog/release), event-loop)

* <https://nodejs.org/en/blog/release>

* <https://medium.com/sharenowtech/node-js-10-is-the-lts-the-enterprise-will-love-2395372a80c3>



Metrics enhancements

1. Standardized metric naming conventions enforced through-out code-base on central-ledger and ml-api-adapters: moja_<process>_<sub-process>
2. Labels were added to identify the specific processing component
3. Example metrics:
 1. moja_transfer_prepare (labels: serviceName="central-handler-prepare" or "ml_service")
 2. moja_transfer_position (labels: serviceName="central-handler-position")
 3. moja_transfer_fulfil (labels: serviceName="central-handler-fulfil" or "ml_service")
 4. moja_notification_event (labels: serviceName="ml-handler-notification")

Improved Code Standardization, Quality & Maintenance

1. Hard-coded values were re-factored to use enumerations for ml-api-adapter & central-ledger (future story to create an enum framework across micro-service components to fix duplication of enums across components).
2. Improvements on routing and call-back handling between the ml-api-adapter and central-ledger:
 1. Standard Header transformer for ml-api (specific to mojaloop transport):
 1. Remove unwanted headers (e.g. FSPIOP-Signature when FSPIOP-Source="switch")
 2. Set correct headers depending on scenario (e.g. FSPIOP-Method: POST, PUT, etc for the correct use-case prepare, fulfil, etc)
 2. Destination/Source are set by the central-ledger (except for the additional call-back scenario for fulfils which is handled by the ml-api - in future this will be optional config)

Deployment Updates

1. Externalized [configuration files](#) into a template file to improve maintenance
2. Cleaned up all [configuration files](#) by removing [duplicate/unused](#) values
3. [Upgraded](#) dependent charts for:
 1. [Kafka](#) (upgraded from v1.0.1 to v2.1.0 – improvements: memory, resilience, security, DNS)
 2. [Zookeeper](#) (fixed scaling bugs with deployments)
 3. [Percona-Xtra-db Mysql](#) (fixed issues with metrics, nodeSelector & tolerations)
4. Added new Helm scripts for :
 1. [Documentation](#)
 2. [Account-lookup-service](#)
5. Setup a new [DEV1](#) environment behind a VPC (requiring VPN access):
 1. operational management plane ([Rancher](#))
 1. Now [Highly Available](#)
 2. [Scalable](#) to support future clusters/lab environments
 3. [Certificates](#) are [auto-provisioned](#) & [auto-renewed](#) by cert-manager



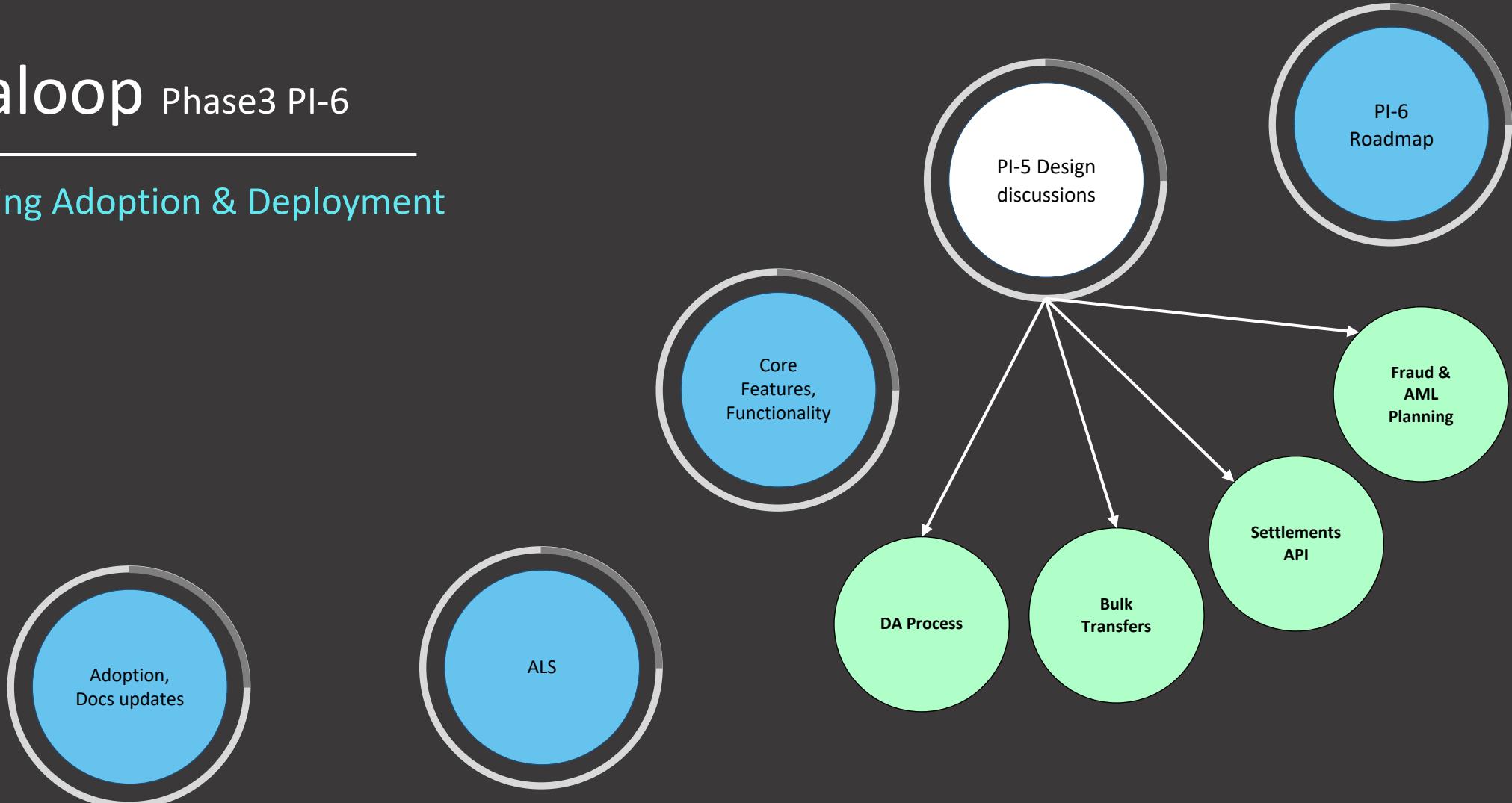
mojaloop

PI-5 Design discussions

Supporting Adoption & Deployment

Mojaloop Phase3 PI-6

Supporting Adoption & Deployment



Design Authority

Aims to

- 1) Ensure a uniform Architecture
- 2) Create a channel for bringing ideas to Design and Verification
- 3) Define Technical Strategies
- 4) Verify Architectural Standards
- 5) Drive Design Methodologies

Design Authority

Implementation

- 1) Have representation from OSS Core Implementer
- 2) Include representatives from commercial implementers and adopters (similar to scrum-of-scrums)
- 3) Should ideally have a DA/Design team for each Implementation Project
- 4) Regular scheduled DA meetings as well as Ad-Hoc high priority design decisions
- 5) Ideally meet in zoom meetings but in rare cases, verification and voting via email or Slack
- 6) Publicly visible Decision Log with clear life-cycle management of decisions
- 7) Invite Public and Community participation via easy submission

Proposal for initial membership

Representation

- 1) BMGF
- 2) ML Core OSS team
- 3) Crosslake Technologies
- 4) Coil
- 5) Mowali
- 6) BoT TIPS
- 7) CCB



mojaloop

Designing Bulk Transfers

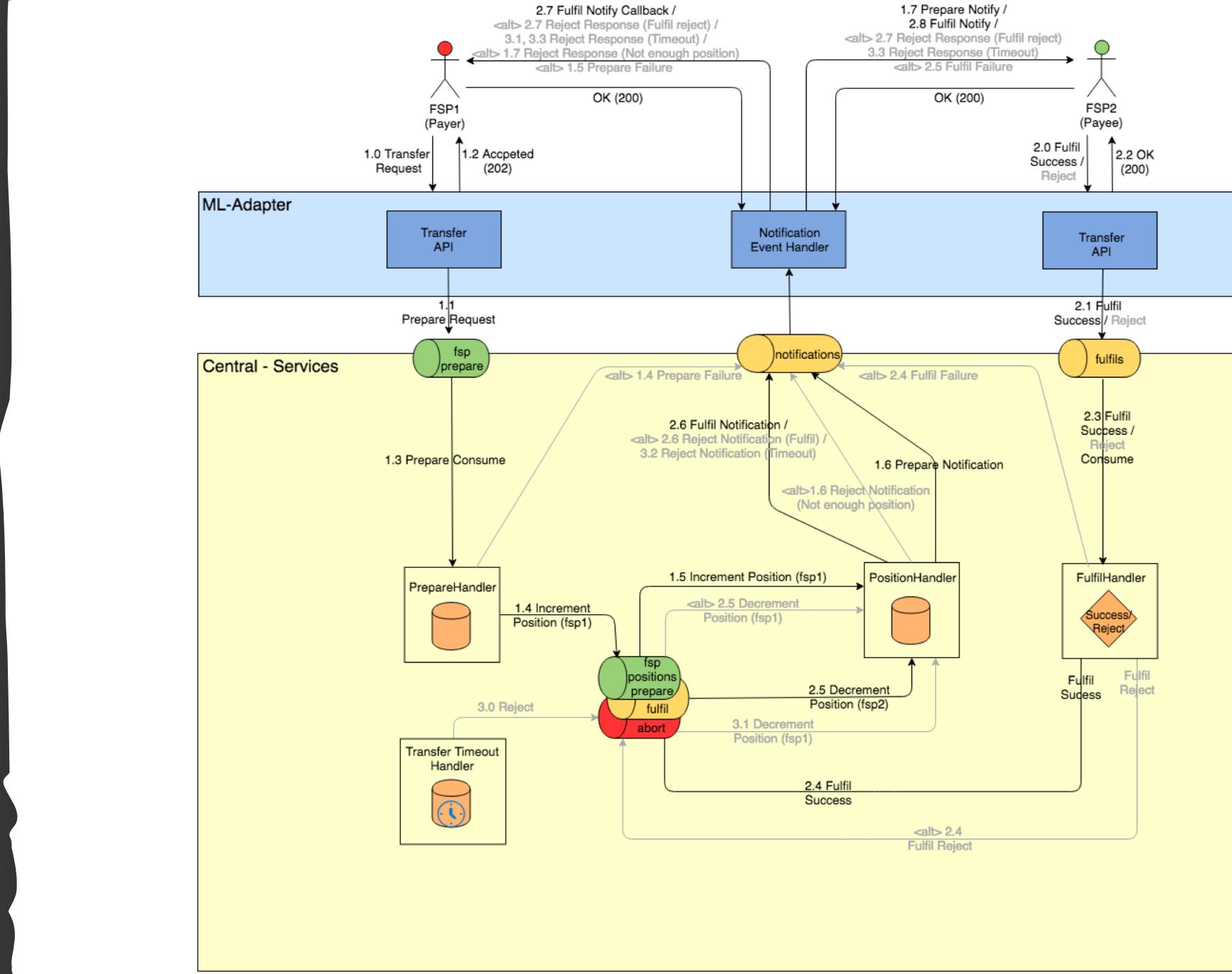
Supporting Adoption & Deployment

PI-5: Bulk Transfers Design

1. Review Individual transfers (Example P2P)
2. Discuss Bulk Transfers Sequence from the Specification
3. Considerations during Design
4. High Level architecture diagram
5. Design items to consider
6. Roadmap

PI-5: Individual Transfers High Level Architecture

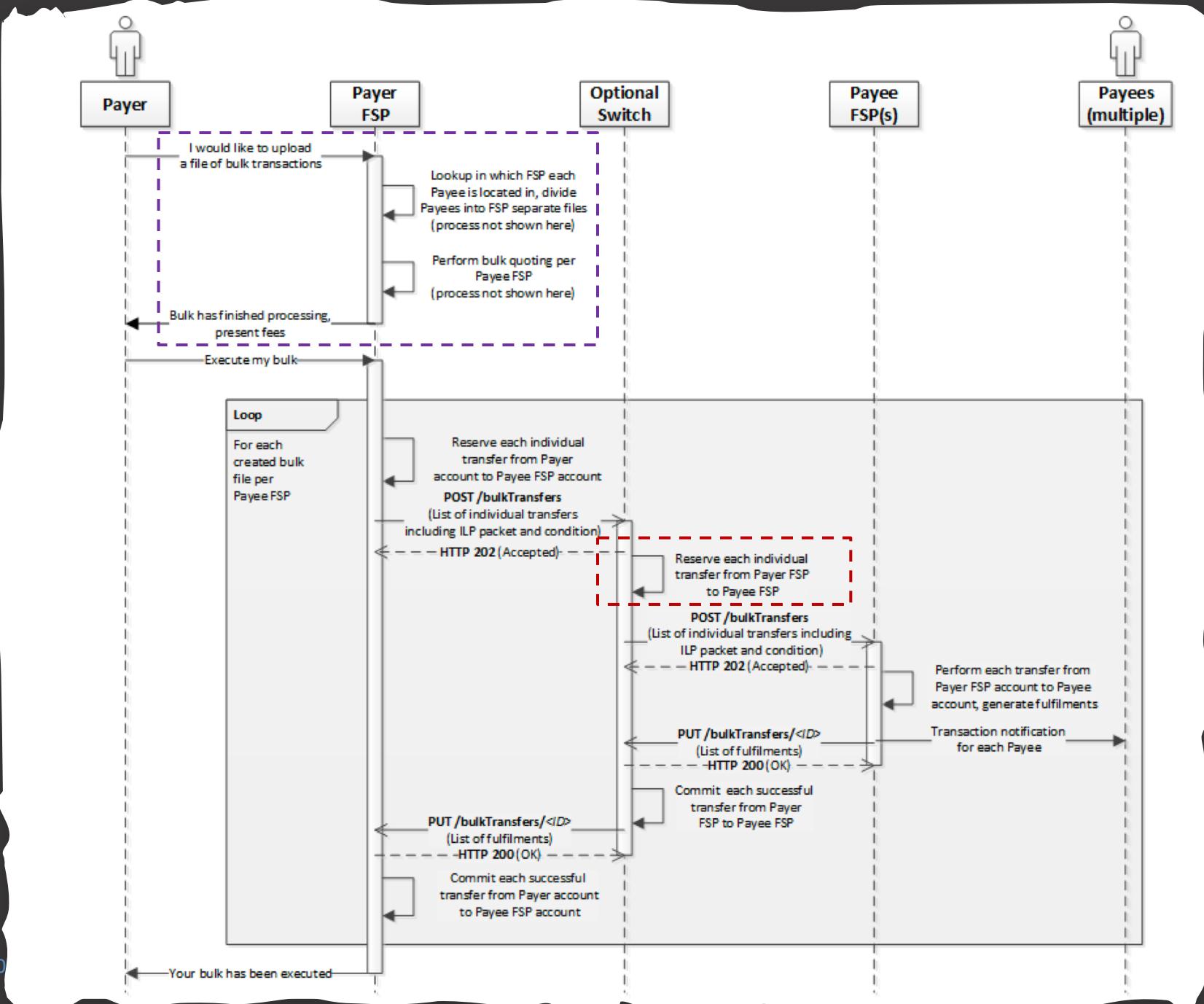
PI-6



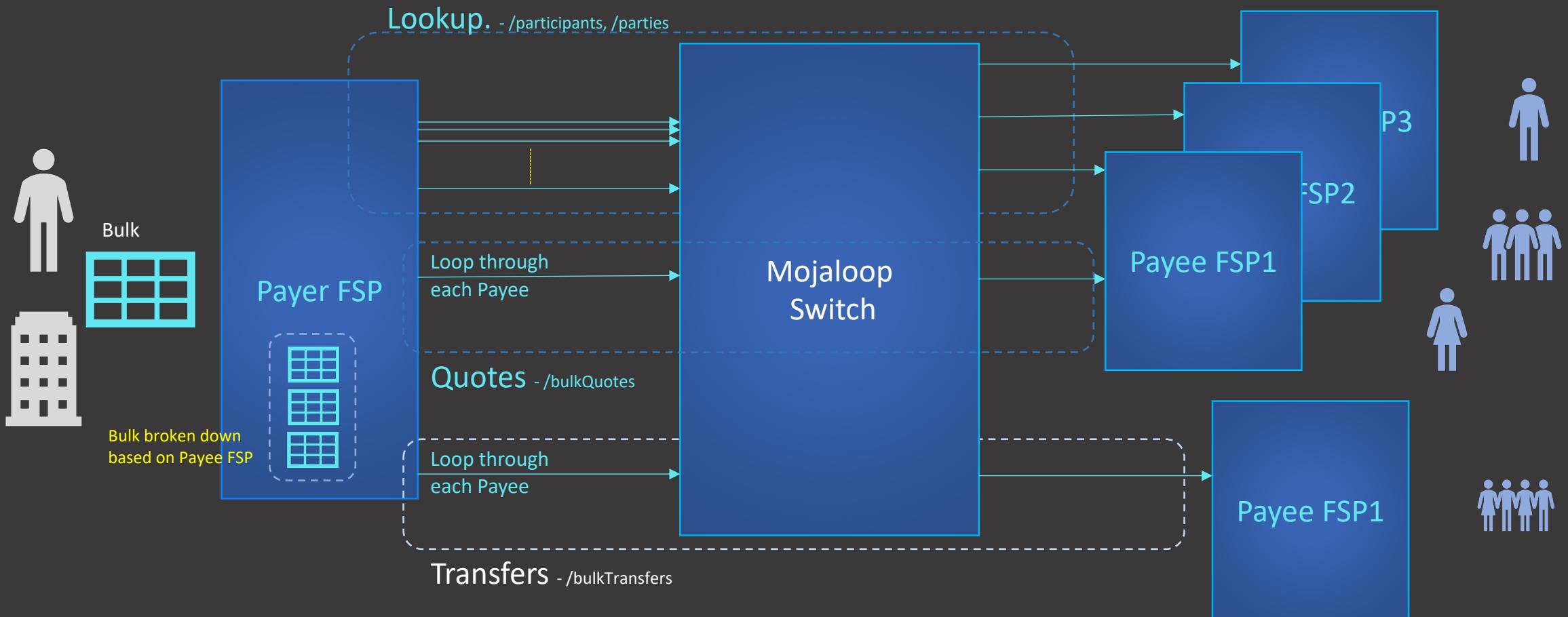
PI-5: Bulk Transfers Sequence from the Spec

PI-6

April 20



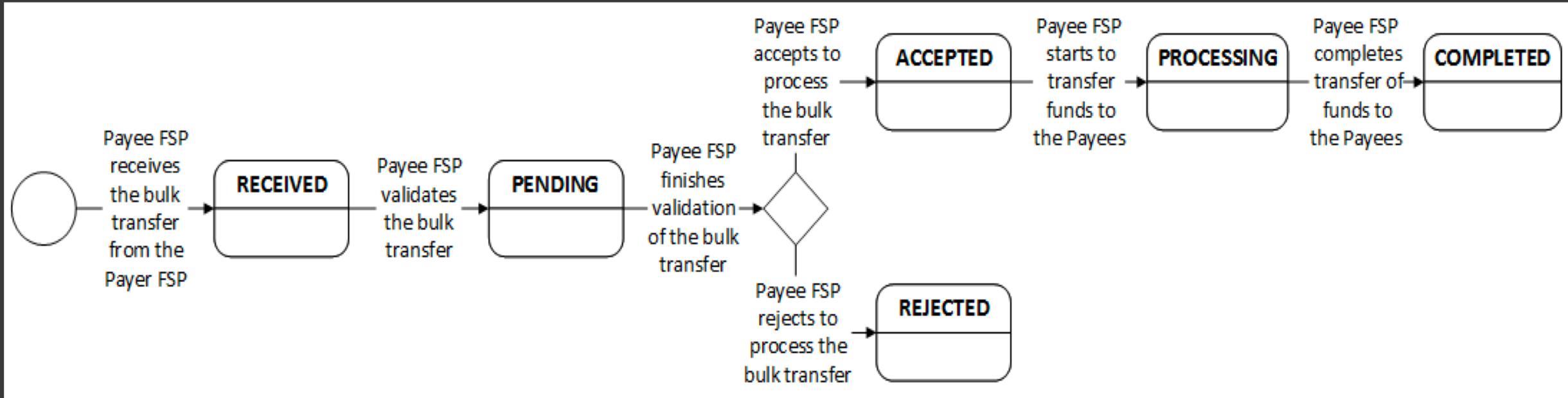
PI-5 Bulk Transfers: Break-down



PI-5: Bulk Transfers – Notes, Changes

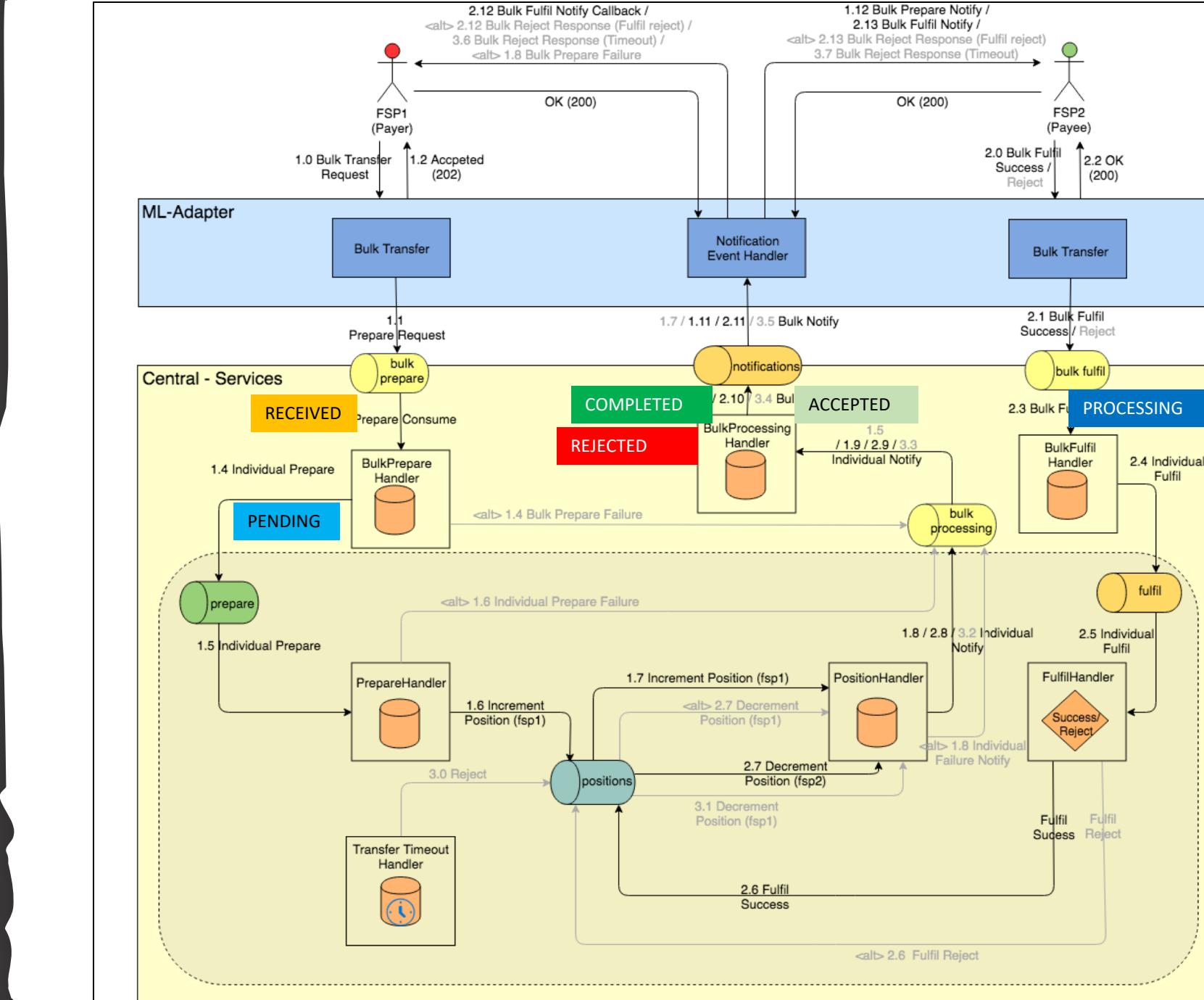
1. Switch to validate bulk, break down into individual transfers
2. Send out Bulk Prepare to Payee only for reserved transfers
3. Send a single aggregated notification (PUT callback) to payer
4. Address considerations for GET calls, validation
5. Validation failures on bulk - **Rejected**
6. Even a single individual transfer is committed - **Completed**

PI-5: Bulk Transfers States



PI-5: Bulk Transfers High Level Architecture

PI-6



PI-5 Bulk Transfers: Design Considerations

1. Specification [Changes](#) to allow Switch capabilities
2. Impact on Settlements, changes needed to support G2P - [Discuss](#)
3. Time-outs considerations
4. [Size](#) considerations of bulkTransfers messages
5. Headers, Signature/Encryption, Security considerations
6. Ensure correct behavior for GET on /bulkTransfers/{ID} and changes proposed
7. Questions

Open PRs for Design:

<https://github.com/mojaloop/docs/pull/143>

<https://github.com/mojaloop/documentation/pull/29>

<https://github.com/mojaloop/central-ledger/pull/267>

PI-5 Bulk Transfers: Roadmap

1. Incorporate Feedback from discussions
2. Finalize design for PoC
3. Implement PoC
4. Submit changes to the API to the CCB
5. After resolution, complete implementation



mojaloop

Settling liabilities in Mojaloop systems

Overview and discussion

How are liabilities incurred?

1. When funds are transferred, the payee DFSP actually credits the beneficiary.
2. Those funds can be cashed out, or converted into any other form at the discretion of the beneficiary...
3. But the payee DFSP hasn't yet received anything more than the promise of payment from the payer DFSP.
4. The promise to pay is the payer DFSP's *liability* to the payee DFSP

The motto of settlements

“Jack shall have Jill;
Nought shall go ill;
The man shall have his mare again,
and all shall be well.”

William Shakespeare

A Midsummer Night's Dream

III, ii, 490-493

Liability and confidence

1. How can a participant be confident that other participants in the scheme will make good on their liabilities?
2. One way is to ensure that no participant can incur liabilities that they are unable to settle.
3. All liabilities are backed by actual funds in a Settlement Account; and perhaps additionally by collateral or credit arrangements at the Settlement Bank
4. Participants are not allowed to transact if their liabilities aren't covered by one of these methods
5. Which implies that, for each transaction, the current total liability of the payer DFSP is calculated and checked against its ability to settle its liabilities
6. A scheme's assessment of a participant's ability to settle its liabilities is defined as the participant's *Net Debit Cap*.
7. *But we're still in the land of promises...*

Net Debit Cap: Manual vs. Automated

1. The Net Debit Cap (NDC) prevents participants incurring liabilities that they may not be able to settle.
2. Its base position is the participant's balance in their Settlement Account.
3. The base position may be varied
4. Positively, for instance to encourage participants or to express trust...
5. Negatively, for instance to allow for account unavailability
6. Ideally, the scheme should be able to get direct information about the participant's balance in their Settlement Account, and use this to set the participant's NDC automatically.
7. In any case, a scheme needs to be able to set a permitted variance margin per participant, or for the scheme overall.
8. If the NDC is to be reliable, the Scheme must be able to approve the removal of funds from the Settlement Account and prevent situations where funds are removed from the Settlement Account between the time the liability is incurred and the time it is settled.

Settlement is the process of making those promises real

1. Which means: each participant should be able to possess the funds they're owed and should be required to disburse the funds they owe.
2. This discussion is about how that happens
3. Next, we look at the sorts of choices that need to be made.

Types of Settlement: Bilateral or multilateral settlements

Bilateral settlements

1. In bilateral settlements, each participant settles separately with every other participant with which it has participated in a transfer.
2. This can be done on either a net or a gross basis
3. That's a lot of settlements...
4. But it *does* mean that settlement is actually occurring between the participants who incurred the liabilities

Types of Settlement: Bilateral or multilateral settlements

Multilateral settlements

1. In multilateral settlements, each participant settles once for all the transfers it has participated in, no matter who the counterparty was.
2. That's a lot fewer settlements
3. But it means that settlement is *actually* occurring between the participant and the scheme, not between the participants who actually contracted the liabilities.
4. In multilateral settlements, the scheme is guaranteeing the settlement of liabilities
5. This is (by far) the most common type of settlement in implemented schemes

Types of Settlement: where is settlement made?

1. Settlement is made using a physical bank account
2. The types of bank account which might be involved are..

Types of settlement bank account

1. Settlement bank accounts can be hosted by
 - a. A central bank
 - b. A commercial bank
2. Settlement bank accounts can be
 - a. Maintained separately for each participant
 - b. Grouped together, either
 - i. Where all settlements take place in a single account
 - ii. Where some participants maintain accounts on behalf of (themselves and) others.
 - A. Where this occurs, it is good practice for the scheme to ensure that the positions of all participants who use the shared account can be treated separately
 - c. Or a mixture of the two...
3. Settlement bank accounts can be
 - a. Used only for the purposes of settlement
 - b. Used for other purposes as well
 - i. In which case, control of the accounts for the purposes of ensuring that settlements can always be covered becomes problematic
4. What is *technically* possible may be limited by legislation or regulation in particular jurisdictions

How do we record the list of promises that were made?

1. The Scheme maintains an internal Position Ledger for each participant (and for each currency in which a participant transacts, where more than one currency is supported.)
2. Each funds transfer in which the participant is involved is recorded in the ledger
 - a. If the participant is the payer, the ledger is debited.
 - b. If the participant is the payee, the ledger is credited.
3. It's important to remember: whether the settlement model is net or gross, this is a question of liquidity
 - a. The scheme needs to know when a ledger entry has been marked for settlement.
 - b. The scheme needs to know when the participant's Settlement Account has been adjusted to settle that ledger entry.

Types of Settlement: Net Settlement

1. *How it works:* at a specified point, the net position of every Position Ledger over the *net settlement period* (i.e. since the last time a net settlement was completed) is calculated.
2. This net position excludes any transfers which have been requested but not yet completed.
 - a. When a transfer is requested, the amount of the transfer is reserved in the payer's ledger, but nothing is yet recorded in the payee's ledger.
 - b. So the ledger position is asymmetric with respect to incomplete transactions.

Net Settlement (continued)

1. How it works: at a specified point, the net position of every Position Ledger over the *net settlement period* (i.e. since the last time a net settlement was completed) is calculated.
2. If the net amount shows more debits than credits:
 - a. The participant owes money to the scheme
 - b. The Scheme deducts funds from the pre-funded total in the participant's Settlement Account.
 - c. It is the participant's responsibility to transfer funds into the Settlement Account to restore their pre-funded total and allow them to continue operating
3. If the net amount shows more credits than debits:
 - a. The scheme owes money to the participant
 - b. The funds are added to their Settlement Account's pre-funded value
 - c. The participant may withdraw some or all of the funds from their Settlement Account, subject to the consent of the Scheme

Types of Settlement: Gross Settlement

1. *How it works:* the Settlement Accounts for each participant in a transfer are adjusted for each transaction that is completed, as it is completed
2. Settlement amounts and pre-funded values are always up to date.
3. This works very efficiently for a single pooled account, but potentially requires an account transfer mechanism which can support large numbers of transactions if a single pooled account is not used.
4. This is how RTGS works for bank reserve accounts
 - a. But RTGS is designed for high-value, low-volume transactions
 - b. This model may not be appropriate for the volumes of transactions we have in mind.
5. The single pooled account is analogous to the way in which Mobile Money systems work successfully today:
 - a. The pooled account is equivalent to the Control Account in a Mobile Money system: it represents the total amount of funds in the system.
 - b. The Scheme, like the MMS, records the position of individual participants in relation to the total account
 - c. Participants can add funds to the account as required, in the same way as participants in an MMS can add funds to their e-money accounts by adding funds to the Control Account.

Net vs. Gross Settlement: Pros and Cons

1. Accounts can be either pooled or individual
 - a. Pooled accounts can cover a combination of participants, while other participants have their own accounts;
 - b. Or there may be a single account which pools the funds of all participants
2. Individual accounts can be either dedicated or commingled
 - a. Commingled accounts are accounts which are, or can be, used for purposes other than settlement.
 - b. Where an account can be used for purposes other than settlement:
 - c. it is not possible to rely on the funds in it for the purposes of assessing the liabilities which a participant can guarantee.
 - d. The Net Debit Cap for the Participant(s) who use this account must therefore be set manually by the Scheme

This gives us the following settlement type matrix

| Problems and constraints in various settlement models | | | |
|---|---|--|---|
| Settlement Model | Individual Participant Settlement Bank Account, Multi-Purpose | Individual Participant Settlement Bank Account, Scheme-Dedicated | Pooled Settlement Bank Account (all Participants), Scheme-Dedicated |
| Net Settlement | | | |
| Gross Settlement | | | |

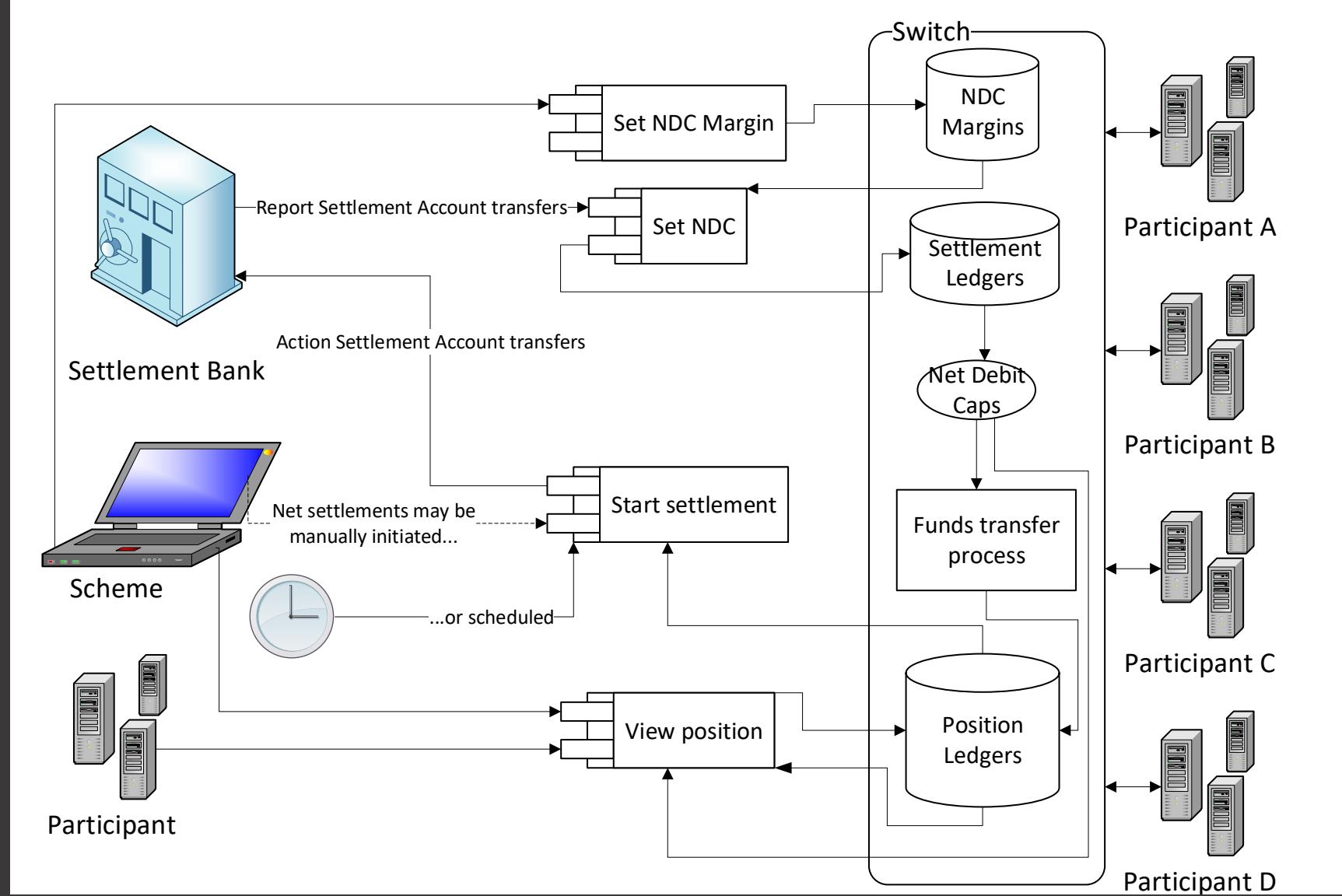
Net vs. Gross Settlement: What are the potential problems?

1. All commingled accounts prevent the automation of the Net Debit Cap
2. All gross settlement to individual accounts has the RTGS volume limitation problem
 1. RTGS systems are designed for high-value, low-volume transactions
 2. Our model is low-value, high-volume...
3. All net settlement to individual accounts has the 24x7 problem
 1. Net settlements will typically require larger adjustments of funds...
 2. Systems may not be available when required
4. All net settlement has the reconciliation problem
 1. Because settlement takes place in large batches, participants can only reconcile after the fact.
 2. Whereas gross settlement allows participants to check their expected positions in real time.

This gives us the following settlement type matrix

| Problems and constraints in various settlement models | | | |
|---|--|---|--|
| Settlement Model | Individual Participant Settlement Bank Account, Multi-Purpose | Individual Participant Settlement Bank Account, Scheme-Dedicated | Pooled Settlement Bank Account (all Participants), Scheme-Dedicated |
| Net Settlement | <ul style="list-style-type: none">• No automated NDC calculation• 24x7 problem• Reconciliation problem | <ul style="list-style-type: none">• 24x7 problem• Reconciliation problem | Why would you? |
| Gross Settlement | <ul style="list-style-type: none">• No automated NDC calculation• High volume constraint | <ul style="list-style-type: none">• High volume constraint• 24x7 problem |  |

Suggested outline API Schematic for settlements





mojaloop



The floor is yours...



mojaloop

Fraud & AML

Supporting Adoption & Deployment



Requirements now available for feedback

- Document now available here
- Key components Identified
 1. Risk Scoring of Transfers
 2. Process Modelling
 3. User Blacklisting
 4. Data Repository
 5. Transaction Tracing
 6. Key Event Monitoring
 7. Management Reporting
 8. Suspicious Activity Reporting
 9. Case Management
 10. External Data Capture

Key Tasks we need to address

1. We need to introduce some new processes to assist in the sharing of typologies
 - a. A common framework for creation
 - b. Management of who sees the fraud type
 - c. Management of who sees the resolution
2. Whilst maintaining an Open Source approach

Cressey's Fraud Triangle



Gated Milestone Approach

1. Adopted from the Level One Project

*Fraud Scenarios in the Mobile
Money Ecosystem*

2. Will assist in the identification of weak points
3. Will be used when analyzing the Typology

Prospecting

Identification of accounts where money can be stolen from

1

Norming

Creation of the Fraud Execution Process – but not limited to

- Removal of funds from identified Account
- Parking of the funds
- Liquidation of funds

2

Transacting

Fraud executed and Financial Transition is performed

3

Liquidation

Money is liquidated into cash or services

4

Assessment of the Typology

STRIDE – used for Security threats

1. Spoofing Identity
2. Tampering with Data
3. Repudiation
4. Information Disclosure
5. Denial of Service

DREAD – used for assessing threat risk Level

1. Damage Potential
2. Reproducibility
3. Exploitability
4. Affected Users
5. Discoverability

Sample Typology – that would impact settlement

Spoofing Emails of the Bank Mandate Holder

1. Transfer account for Funds Out changed via email notification
2. When DFSP Requests Excess Funds to be transferred back, they are sent to the new account – and rapidly moved on

Stride Classification

1. Spoofing
2. Denial of Service

Stage of Fraud Assistance

1. Financial Transaction

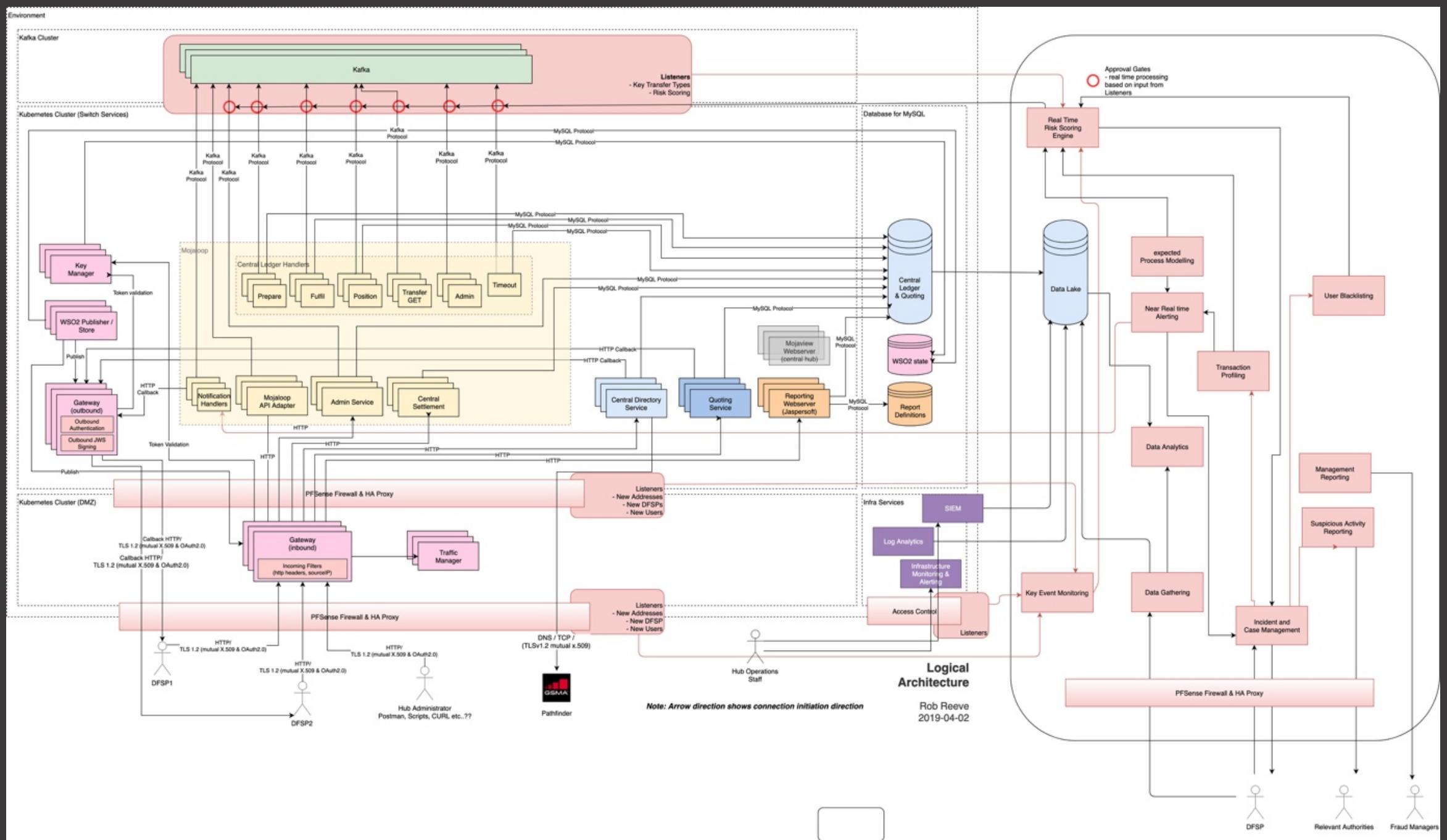
| DREAD SCORE | 5.6 |
|------------------|-----|
| Damage Potential | 10 |
| Reproducibility | 10 |
| Exploitability | 5 |
| Affected Users | 1 |
| Discoverability | 2 |



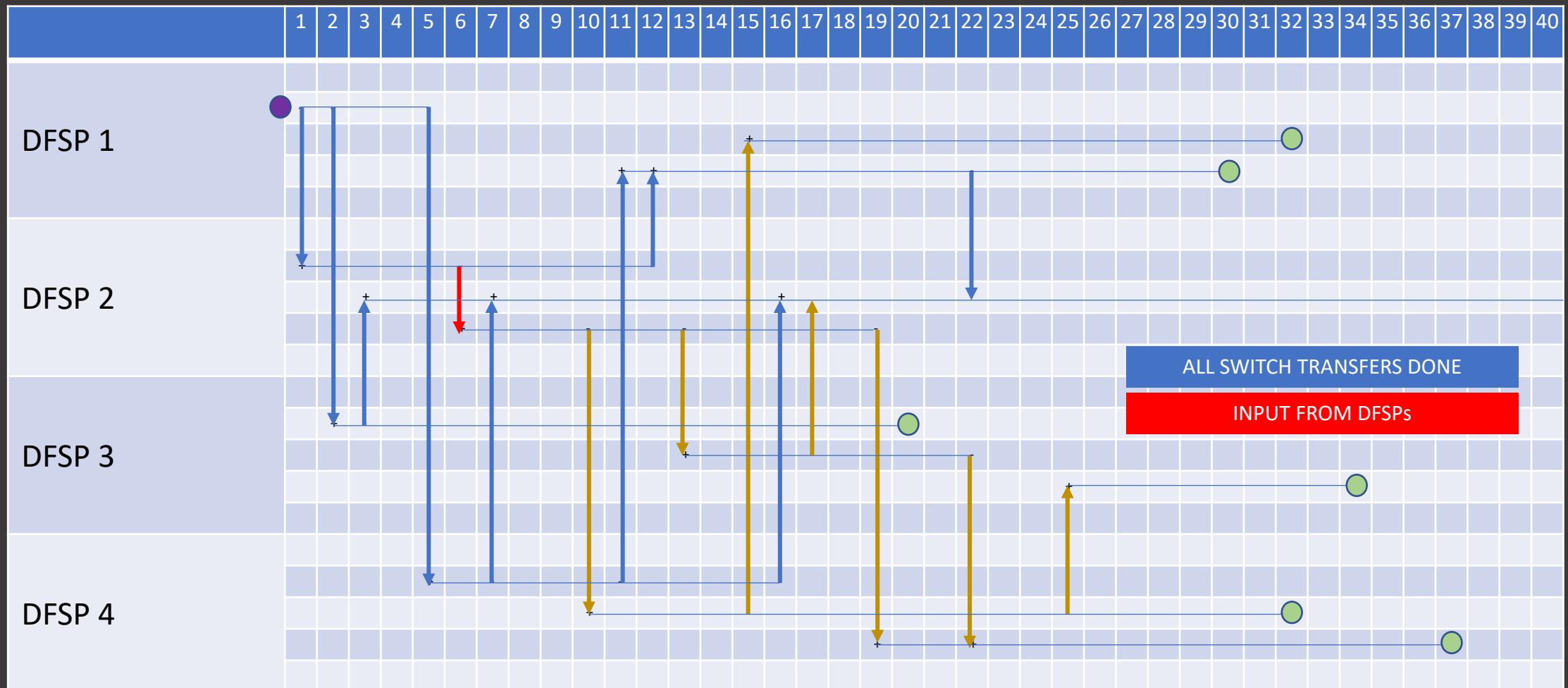
Sharing of Typologies

1. During Assessment
 - a. We risk sharing new approaches in Fraud with the criminals who we are looking to stop – before a remediation has been implemented
2. Post implementation
 - a. We need to prove the development works – before the Typology is released
3. But there are examples of this process working well already for Fintechs

Architecture



Money Tracing as an Example

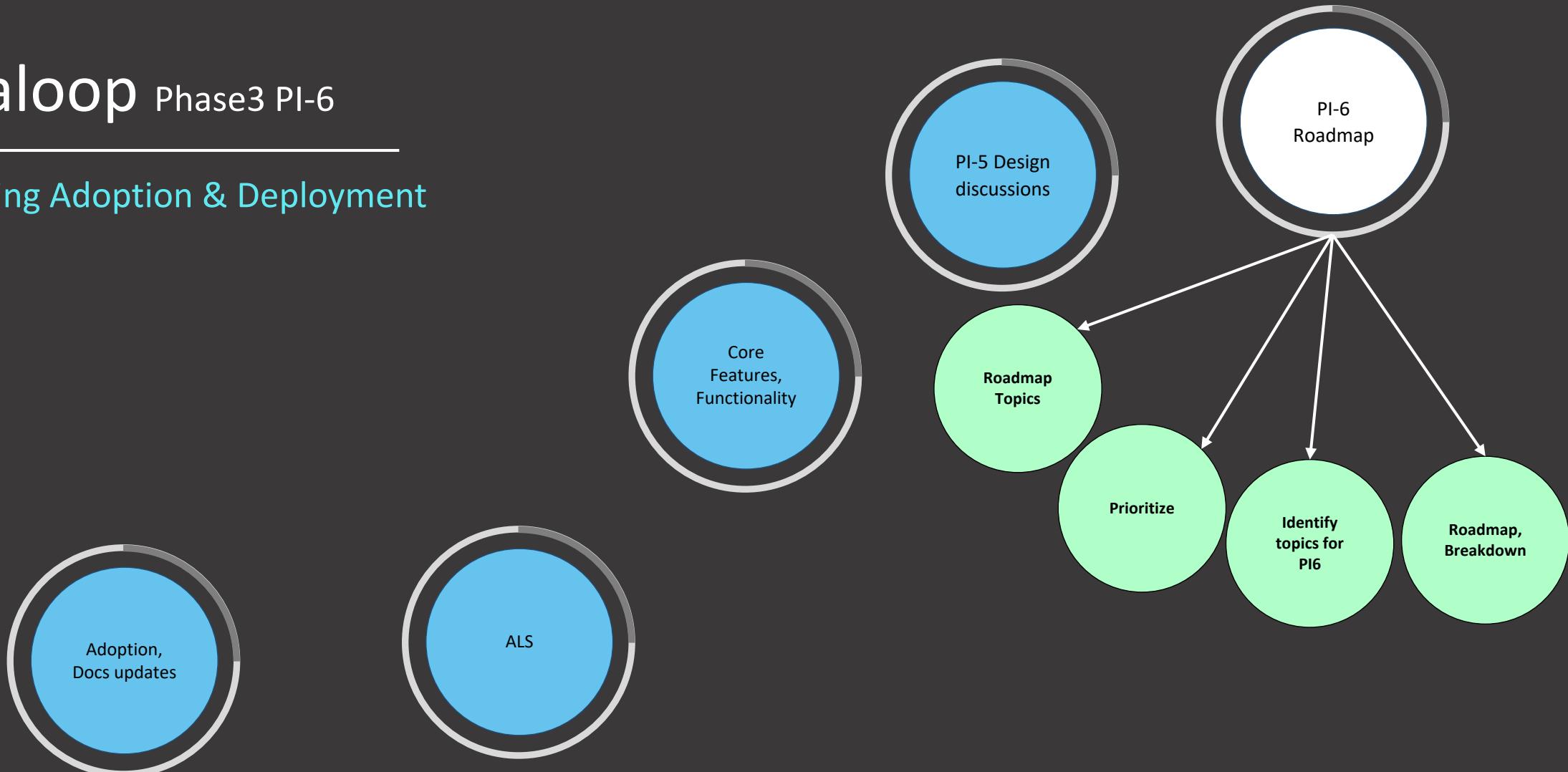


Working Assumptions on implementation

1. Case and Incident Management will not be addressed – most organizations already have an existing platform
 - a. But Open Source solutions should be suggested for those who do not
2. Machine Learning and Modelling will be introduced – however data would be required before any real value would be seen
 - a. This moves into the realm of Data Scientists
3. This will be revisited in the PI Planning

Mojaloop Phase3 PI-6

Supporting Adoption & Deployment



Switch Functionality – Mojaloop End-points (Phase-3)

Mojaloop v1.0 – API Specification

Transfers:

- [●] POST - Prepare
- [●] PUT - Response
- [●] PUT – Error
- [●] Outgoing
- [●] Incoming
- [●] GET - Query

Parties:

- [●] GET - Request
- [●] PUT - Response
- [●] PUT - Error

Quotes:

- [●] POST - Request
- [●] PUT - Response
- [●] PUT - Error
- [●] GET - Query

Participants:

- [●] POST - Create
- [●] PUT - Response
- [○] POST - Bulk Create
- [●] PUT - Error
- [○] DEL - Delete

Transactions:

- [○] PUT - Response
- [○] GET - Query

TransactionRequests:

- [●] POST - Request
- [●] PUT - Response
- [●] PUT - Error
- [●] GET - Query

Authorizations:

- [○] GET - Request
- [○] PUT - Response
- [○] PUT - Error

BulkTransfers:

- [●] POST - Request
- [●] PUT - Response
- [●] PUT - Error
- [●] GET - Query

BulkQuotes:

- [○] POST - Request
- [○] PUT - Response
- [○] PUT - Error
- [○] GET - Query

Key

- [●] Fully implemented
- [●] Legacy Code
- [●] Partially implemented
- [●] Not implemented
- [○] Out of Scope for PI6

PI-6 Roadmap: Continuation of ongoing features

1. Cross network/border
2. Account Lookup Service
3. Bulk Payments
4. Documentation
5. QA, Testing

PI-6 Roadmap: Error Handling Framework

PI-6 Roadmap: Forensic Logging



PI-6 Roadmap: Fraud & AML

PI-6 Roadmap: Merchant Payments

PI-6 Roadmap: Quotes Service



PI-6 Roadmap: Workshops

Release Mechanism: Sample Helm Release

The screenshot shows the GitHub releases page for the repository `mojaloop/helm`. The page displays the `v5.5.2 Release`, which was released by `elnyry` 9 days ago. The release notes mention upgrading to helm v5.5.2. The application versions supported are central-ledger v5.5.2, ml-api-adapter v5.5.0, central-settlement v5.5.0, central-event-processor v5.3.0, and email-notifier v5.3.0. The release also links to the application release notes for ml-api-adapter, central-ledger, central-settlement, email-notifier, and central-event-processor.

Latest release
v5.5.2
64d88a6
Verified

v5.5.2 Release

elnyry released this 9 days ago

Helm release changes:

- Upgrading to helm v5.5.2 to be able to use central-ledger v5.5.2 release and v5.5.0 of documentation (#165)

Application versions:

Application versions that are supported for this update:

- central-ledger: v5.5.2
- ml-api-adapter: v5.5.0
- central-settlement: v5.5.0
- central-event-processor: v5.3.0
- email-notifier: v5.3.0

Application release notes:

- ml-api-adapter - <https://github.com/mojaloop/ml-api-adapter/releases/tag/v5.5.0>
- central-ledger - <https://github.com/mojaloop/central-ledger/releases/tag/v5.5.2>
- central-settlement- <https://github.com/mojaloop/central-settlement/releases/tag/v5.5.0>
- email-notifier - <https://github.com/mojaloop/email-notifier/releases/tag/v5.3.0>
- central-event-processor - <https://github.com/mojaloop/central-event-processor/releases/tag/v5.3.0>

Assets 2

- Source code (zip)
- Source code (tar.gz)

Go Back

v5.4.0

122a9ed

Verified

v5.4.0 Release

mdebarros released this 25 days ago · 5 commits to master since this release

[Edit](#)

Bug fix changes:

- Fixes below cater for the following bug [mojaloop/project#687](#)
- POST /transfers when fulfil transferState='ABORTED' now returns the correct payload
- GET /transfers when fulfil transferState='ABORTED' now returns the correct payload
- Minor bug in the Participant domain was not returning the headers as the message contained `header` instead of `headers` key
- GET /transfers was not returning the correct error when a transfer ID was not found. This now returns `ERROR 3208`.
- Central-switch designation has been changed to `switch` which is aligned to the Mojaloop specification

Maintenance changes:

- Added a new Util.clone() method and updated most (if not all) clone methods to use this common operation
- Moved toFulfil transform method from the Transfer Handlers to transformer.js
- Added unit tests for the toFulfil transformer method to ensure code-coverage is maintained
- Added new ENUM for TransferStateEnum which is required for logical comparison within the Transfer Handlers
- Updated all Transfer Transformers to return only fields that have values, and leave out any undefined/null key-values from the output transformation result in transformer.js
- Bumped version to 5.4.0
- Cleaned up config by removing kafka configs for reject and abort as their respective handlers do not exist.

▼ Assets 2

[Source code \(zip\)](#)

[Source code \(tar.gz\)](#)

[Go Back](#)

Release Mechanism: Slack Announcements

#announcements
1 star | 35 messages | 0 files | Add a topic

Thursday, April 11th

10:55 PM MojaBot APP central-event-processor - Release v5.5.0: <https://github.com/mojaloop/central-event-processor/releases/tag/v5.5.0>
email-notifier - Release v5.5.0: <https://github.com/mojaloop/email-notifier/releases/tag/v5.5.0>

11:43 PM MojaBot APP ml-api-adapter - Release v5.5.1: <https://github.com/mojaloop/ml-api-adapter/releases/tag/v5.5.1>

Friday, April 12th

4:30 AM MojaBot APP QA-Regression Report for https://raw.githubusercontent.com/mojaloop/postman/master/Golden_Path.postman_collection.json on https://raw.githubusercontent.com/mojaloop/postman/master/environments/Mojaloop-DEV0.postman_environment.json at 2019-04-11 23:00:21 - FAILED
[View QA and Regression test res...](#)
 1 reply 1 day ago
 1 reply 1 day ago

9:48 AM Sam replied to a thread: QA-Regression Report for https://raw.githubusercontent.com/mojaloop/postman/master/Golden_Path.postman_collection.json on https://raw.githubusercontent.com/mojaloop/postman/master/environments/Mojaloop-DEV0.postman_environment.json at 2019-04-11 23:00:21 - FAILED
Did not run the FSP setup on this one, so this can be ignored.. Should be rerun when possible (I'm going to redeploy with a newer version of ml-api-adapter and then run the FSP setup)
 1 

9:58 AM MojaBot APP ml-api-adapter - Release v5.5.2: <https://github.com/mojaloop/ml-api-adapter/releases/tag/v5.5.2>

6:46 PM MojaBot APP QA-Regression Report for https://raw.githubusercontent.com/mojaloop/postman/master/Golden_Path.postman_collection.json on https://raw.githubusercontent.com/mojaloop/postman/master/environments/Mojaloop-DEV0.postman_environment.json at 2019-04-12 13:16:56 - PASSED
[View QA and Regression test res...](#)

7:21 PM Sridevi Miriyala hi @Nico Duvenage ..after the new deployment, did the test run without any changes in Postman scripts?
which test is that?

7:24 PM Nico Duvenage Let me send it to you on your channel.

7:24 PM Sridevi Miriyala k thx

Yesterday

4:36 AM MojaBot APP QA-Regression Report for https://raw.githubusercontent.com/mojaloop/postman/master/Golden_Path.postman_collection.json on https://raw.githubusercontent.com/mojaloop/postman/master/environments/Mojaloop-DEV0.postman_environment.json at 2019-04-12 23:06:29 - PASSED
[View QA and Regression test res...](#)
 2

Today

12:04 AM MojaBot APP QA-Regression Report for https://raw.githubusercontent.com/mojaloop/postman/master/Golden_Path.postman_collection.json on https://raw.githubusercontent.com/mojaloop/postman/master/environments/Mojaloop-DEV0.postman_environment.json at 2019-04-13 18:34:40 - FAILED
[View QA and Regression test res...](#)

12:20 AM MojaBot APP QA-Regression Report for https://raw.githubusercontent.com/mojaloop/postman/master/Golden_Path.postman_collection.json on https://raw.githubusercontent.com/mojaloop/postman/master/environments/Mojaloop-DEV0.postman_environment.json at 2019-04-13 18:50:36 - PASSED
[View QA and Regression test res...](#)
