



mojaloop

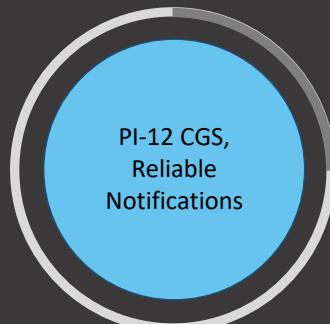
# Mojaloop PI-13

---

Phase-5 One Loop for All: Mojaloop in Motion

# Mojaloop PI-13

Phase-5 One Loop for All: Mojaloop in Motion



# PI-12 Core Report-out: Agenda

1. PI-12 Overview of changes and progress
2. OSS Community, Collaboration Updates
3. PI-12 Focus Areas



mojaloop

# ML Core Overview PI-12

---

Features, Improvements & Community Support

# Mojaloop PIs Overview

Timeline	Summary
<b>Phase-1</b> (2016 - 17)	<p><b>Level One Project</b></p> <ul style="list-style-type: none"> <li>• Reference Implementation</li> <li>• 6 Program Increments (PIs)</li> </ul>
<b>Phase-2</b> (2018)	<p><b>Road To Productionization</b></p> <ul style="list-style-type: none"> <li>• 1 – 4 Program Increments</li> </ul>
<b>Phase-3</b> (2019 Jan - Dec)	<p><b>Supporting Adoption &amp; Deployment</b></p> <ul style="list-style-type: none"> <li>• PI-5 (Feb – April): Account lookup, QA Framework, Streamlined CI, Release process, Error endpoints, Documentation, Node Upgrade, Bug Fixes &amp; Community support, Bulk Transfers Design</li> <li>• PI-6 Event handling framework, Bulk Transfers PoC, API Gateway, OSS Settlements API, Quoting Service, ALS</li> <li>• PI-7 Event &amp; Error Handling framework, Packaging, OSS Settlements, Performance testing capabilities, QA</li> <li>• PI-8 Consolidation, Performance, Community Support</li> </ul>
<b>Phase-4</b> (2020 Jan - Dec )	<p><b>Going Live!</b></p> <ul style="list-style-type: none"> <li>• PI-9: Performance Testing &amp; Improvements, Merchant Request to Pay, Operational Monitoring, Testing toolkit, Settlement v2</li> <li>• PI-10: Performance PoC, Standardizing Bulk Transfers, Testing toolkit, Settlement v2, Adopting FSPIOP API v1.1, Versioning</li> <li>• PI-11: Performance PoC, Standardizing designs, code &amp; QA, Adopting FSPIOP API v1.1, Versioning, PISP support, Code quality</li> <li>• <b>PI-12: Hardening, Packaging, Releases and prep for Phase-5 (October 2020 – January 2021)</b></li> </ul>
<b>Phase-5</b> (2021 Jan - Dec )	<p><b>One Loop for All: Mojaloop in Motion!</b></p> <ul style="list-style-type: none"> <li>• <b>PI-13: Removing friction, Understanding &amp; addressing (feature) gaps, Cost reduction, Increasing Trust &amp; confidence in Mojaloop (February 2021 – April 2021)</b></li> </ul>

# Switch Functionality – Mojaloop (Phase-4 PI-12)

## Payer-Initiated Transaction\*

- [●] P2P Transfers
- [●] Prepares, Fulfils, Query
- [●] Rejections, Timeouts
- [●] Error Endpoints

## Payee-Initiated Transaction\*

- [●] MIMP Transfers
- [●] Transaction Requests
- [●] Prepares, Fulfils, Query
- [●] Rejections, Timeouts
- [●] Error Endpoints

## Mojaloop: Focus Use-cases

1. P2P
2. Merchant ‘Request to Pay’
3. Bulk Payments
4. **PISP payments**

## Bulk Payments\*

- [●] Bulk Transfers
- [●] Prepares, Fulfils, Query
- [●] Rejections, Timeouts
- [●] Error Endpoints

## PISP Payments\*

- [●] PISP
- [●] Transfers
- [●] Account Linking
- [●] Un-linking
- [●] Error scenarios

### Key

- [●] Fully implemented
- [●] PoC / Initial Version
- [●] Partially implemented
- [●] Not implemented
- [○] Out of Scope

## Mojaloop FSPIOP API v1.1 – Use-cases

### Payer-Initiated Transaction

- [●] P2P Transfers
- [●] Prepares, Fulfils
- [●] Rejections, Timeouts
- [●] Error Endpoints
- [●] Customer-Initiated Merchant Payment
- [●] Customer-Initiated Cash-out - Receive Amount
- [●] Customer-Initiated Cash-out - Send Amount
- [●] ATM-Initiated Cash-out
- [●] Refund

### Bulk Transactions

- [●] Bulk Payments

### Payee-Initiated Transaction

- [●] Merchant-Initiated Merchant Payment
- [●] Agent-Initiated Cash-out
- [●] Agent-Initiated Cash-In – Send Amount
- [●] Agent-Initiated Cash-In – Receive Amount

### Payee-Initiated Transaction using OTP

- [●] Merchant-Initiated Merchant Payment Authorized on POS
- [●] Agent-Initiated Cash-out Authorized on POS

#### Key

- [●] Fully implemented
- [●] Supported, not tested
- [●] Proof of Concept
- [●] Not implemented
- [○] Out of Scope

# Switch Functionality: Mojaloop End-points (PI-11 → PI-12)

## Mojaloop v1.1 – FSPIO API Specification

### Transfers\*

- [●] POST - Prepare
- [●] PUT - Response
- [●] PUT - Error
- [●] PATCH - Notification
- [●] 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 (v1.1)
- [●] PoC / Initial Version
- [●] Partially implemented
- [●] Not implemented
- [○] Future Roadmap

## Operational – Use Cases

### Participants

- [●] Manage Participants
- [●] Create Initial Value
- [●] Query
- [●] Update
- [●] Manage Participant Limits
- [●] Create Initial Value
- [●] Query
- [●] Update
- [●] Manage Callback URLs
- [●] Create Initial Value
- [●] Query
- [●] Update

### Settlement features\*

- [●→●] Ledger account type
- [●→●] Create
- [●→●] Query
- [●→●] Settlement models
- [●→●] Configure
- [●→●] Query

### Oracles (ALS)

- [●] Manage Oracles
- [●] Create
- [●] Query
- [●] Update
- [●] Delete

### Monitoring, Tracing

#### Tracing

- [●] Transfers
- [●] Quotes
- [●] ALS

#### Metrics

- [●] Transfers
- [●] Quotes
- [●] ALS

#### Key

- [●] Fully implemented
- [●] PoC / Initial Version
- [●] Partially implemented
- [●] Not implemented
- [○] Roadmap

## Settlements v1.0 [Deferred, Multilateral Net]

- [●] Open, close Settlement Windows
- [●] Query Settlement Windows
- [●] Query Settlement Report
- [●] Create/Trigger Settlement with Windows
- [●] Process successful Settlement Acknowledgements
- [●] Reconcile Positions based on successful Settlements
- [●] Process failed Settlement Acknowledgements

## Positions

- [●] Query Positions
- [●] Manage Positions
- [●] Create Initial Value
- [●] Query
- [●] Update

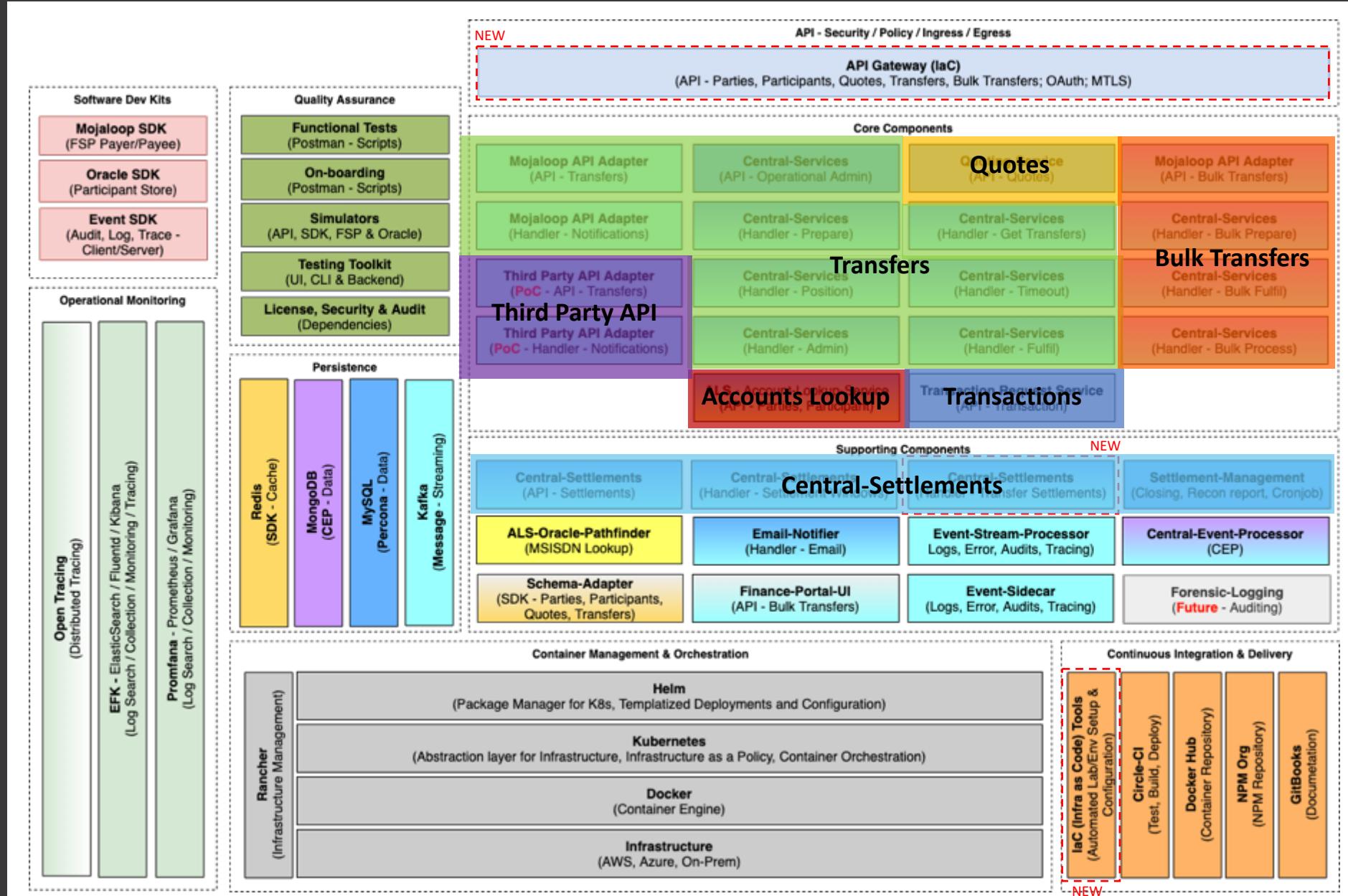
## Settlements v2

- [●] Settlement models
- [●] Designs
- [●] Settlement by Currency
- [●->●] Interchange Fees
- [●->●] Continuous Gross Settlement

### Key

- [●] Fully implemented
- [●] PoC / Initial Version
- [●] Partially implemented
- [●] Not implemented
- [○] Roadmap

# PI-12: Component Architecture





mojaloop

## ML Community Updates

---

Features, Improvements & Community Support

# PI-12 Overview: ML OSS Community

## Contributions, Collaboration

1. Settlement v2 (CGS, Interchange Fees)
2. Reliable notifications, cascaded time-outs
3. Quality Assurance (bugs, test coverage)
4. Architecture / Performance PoC and follow-up
5. Code quality and Security improvements
6. PISP Solution
7. Demo working group

# Mojaloop OSS PI-12: Community

1. Change Control Board (**CCB**) and Special Interest Groups (**SIGs**) for each of the APIs
  - a. Admin API SIG
  - b. FSPIOP API SIG
  - c. Third-party API SIG
  - d. Settlement API SIG (in the works)
2. Design Authority (**DA**)
3. Bi-weekly scrum-of-scrums
4. Slack channels: **#general**, **#announcements** , **#help-mojaloop**, **#design-authority** , **#ml-oss-devs**, **#market\_focus**, **#ml-oss-bug-triage**

# ML OSS Community: Design Authority (DA) Overview

## 1. Responsibilities

Ensure cohesion and agreement in the broader context of API Design and Implementation by enforcing design best practices and uniform Architecture (Design Methodologies, Technical Strategies and Design/Verification of Architectural Standards)

## 2. DA Committee Membership

The Design Authority members consists of Five Elected Core Members and Workstream leads as per DA Restructuring Group Document. The chairperson is elected by all DA members. James Bush is new elected chairperson for 2021. Accountable to the TGB and the community.

## 3. DA Functioning

- a. Open to all to propose design related changes – any topic for DA related topic can be submitted via [github.com/mojaloop/design-authority](https://github.com/mojaloop/design-authority).
- b. Weekly Scheduled meeting every Wednesday and ad-hoc meetings on a need to have basis.
- c. Decisions made and progress reports are available anytime on the DA ZenHub Board

## 4. DA Transformation Activities

- a. Platform Design Principles
- b. DA Charter

# ML OSS Community: PI 12 DA Issues overview

mojaloop / design-authority / DA-Issue-Log

DA Backlog	PI 12 In-Progress	Deferred	Decided - Needs Followup
3 Issues - 0 Story Points design-authority #55 Contingency Plan discussion (Generic) for Open-Source Library Support changes PI 12	3 Issues - 0 Story Points design-authority #74 Design for reliable notifications on finalisation of a transfer PI 12	1 Issue - 0 Story Points design-authority #51 Should the PISP implementation live in the 'sdk-scheme-adapter', or should we make a new 'thirdparty-scheme-adapter'? PI 12	2 Issues - 3 Story Points design-authority #69 Proposal: Remove external dependencies that are tightly integrated into the Mojaloop Helm Chart PI 12
design-authority #73 Present the draft Fraud Risk Management architecture to the Design Authority for review PI 12	design-authority #72 IAC contribution to Mojaloop OSS PI 12	To be Discussed	3 theme-non-functional-support
design-authority #67 Differentiate between "core" components and demos/POCs/etc PI 12	design-authority #70 Review Original Mojaloop Project Principles PI 12	needs clarity theme-value-prop	design-authority #71 proposal: enforce conventional commit titles for Pull request reviews PI 12



mojaloop

## PI-12 Core “Focus Areas”

---

Features, Improvements & Community Support

# PI-12 Core Report-out: Team

1. Miguel de Barros
2. Sam Kummary
3. Shashi Hirugade
4. Steven Oderayi
5. Valentin Genev
6. Vijay Guthi



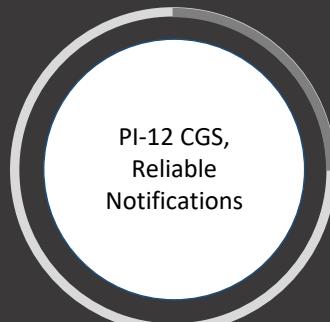
Phase-5

# PI-12 Overview: Focus Areas

1. **Continuous Gross Settlement:** Merging into mainstream, OSS readiness, QA
2. **Reliable Notifications:** Core services (transfers)
3. **Maintenance, Quality:**
  - a. Helm Releases / Release candidate preparation
  - b. QA updates
  - c. Bug Fixes
4. **Testing toolkit:** Comprehensive test suites for – Hub, FSP testing and automated setup
5. **Performance:** Performance / Architecture PoC

# Mojaloop PI-13

Phase-5 One Loop for All: Mojaloop in Motion



# PI-12 Reliable Notifications: Design

## 1. Problems being addressed

- a) As-is notifications do NOT currently support retries
  - i. No retry-mechanism to handle technical issues (e.g., temporary network availability)
- b) As-is notifications provide minimal feedback on notification delivery
  - i. Basic monitoring possible through Grafana dashboards (DFSP, Error-codes)
  - ii. No options for reporting & handling compensating actions
- c) As-is notifications' "adapter" is not easily pluggable
  - i. Core notification logic is tightly-coupled to notifications' "adapter"

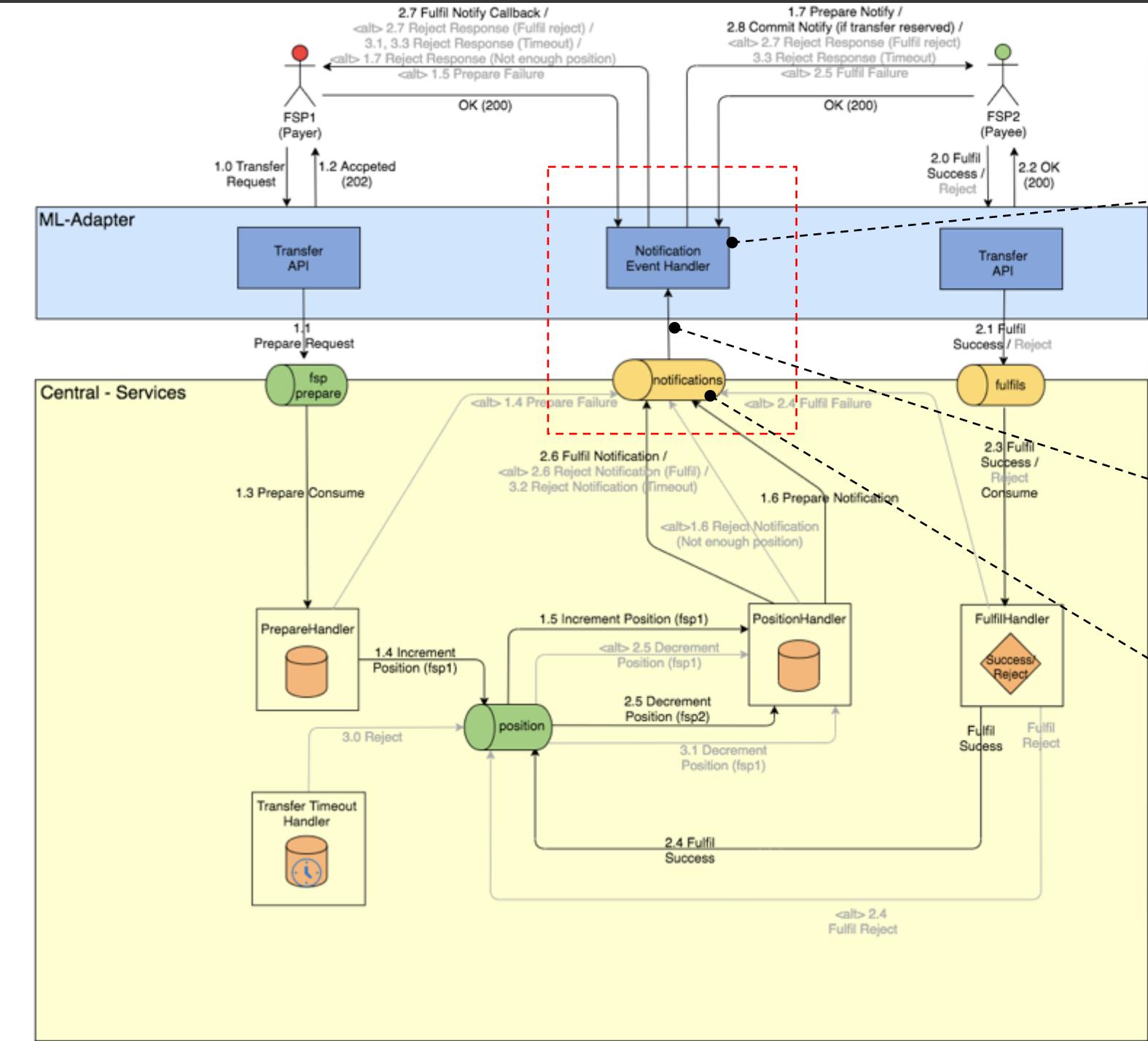
## 2. Tracking Issues

- a) Story: <https://github.com/mojaloop/project/issues/1869>
- b) DA: <https://github.com/mojaloop/design-authority/issues/74>

## 3. Pull Request

- a) <https://github.com/mojaloop/documentation/pull/288> (Draft - WIP pending detailed design)

## PI-12: Transfers Architecture As-Is



### Notifications Handler

- No retry mechanisms
- Tightly coupled notification logic to the ML API "Adapter", difficult to switch out as an "Adapter".

### Minimal Feedback Loop

- Notification results are not fed back to the core
- No opportunity for compensating actions
- Minimal oversight on notification results via Promfana-stack

### Notification Events:

- Mojaloop specific
- **Existing** events produced by core central-services
  - Prepares
  - Fulfilments
  - Aborts
  - Timeouts
  - Validation Failures
  - Errors
  - Alerts

# PI-13: Transfers Architecture To-Be

## Notification Cmd Handler:

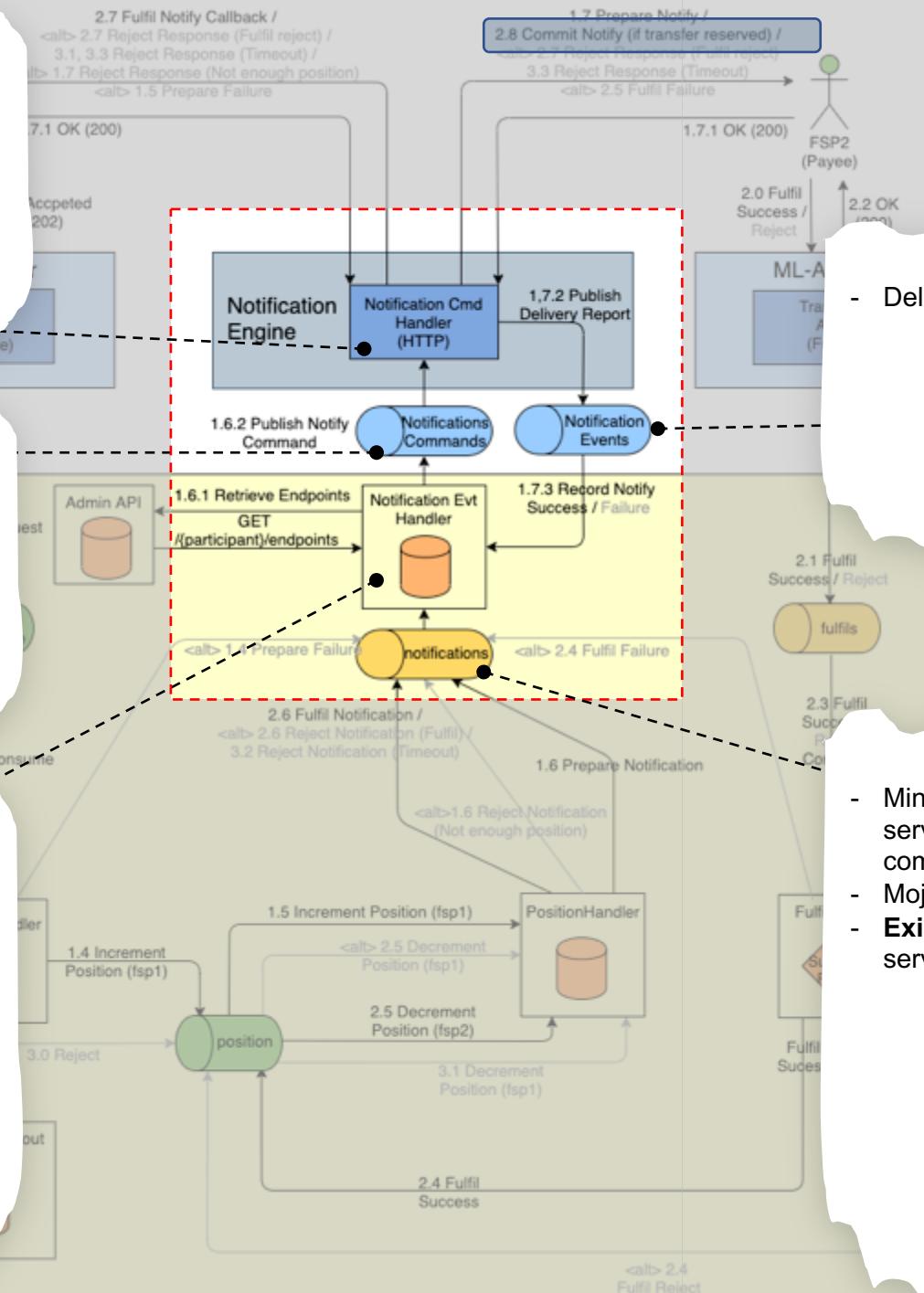
- Stateless notification engine
- Reads from Kafka streams
- Rest API to send Notifications
- Handles TLS and JWS
- Filters & Handles Cmd messages based on Transport + Mime-type
- Pluggable Adapter

## Notification Cmd Events:

- Generic & Use-case agnostic (i.e. not Mojaloop specific)
- Contains:
  - Transport info
  - Mime info
  - Reliable config
  - Headers
  - Payload (unprocessed)

## Notification Evt Handler:

- Stores Notification attempts, and results
- Logic to produce Notification CMDs based on "observed" events
- Rest API to query Notification reports/status
- Possible compensating actions
  - Re-schedule Notifications
  - Alerts
  - Transfer actions (aborts, etc)
  - Monitor/Enforce SLA



## Notification (Engine) Events:

- Delivery report
  - Delivery meta-data
  - Retry meta-data
  - Result status
  - Error info
  - Response (headers/payload)

## Notification Events:

- Minimal or no changes to existing core-services (depending on requirements for compensating actions)
- Mojaloop specific
- **Existing** events produced by core central-services
  - Prepares
  - Fulfilments
  - Aborts
  - Timeouts
  - Validation Failures
  - Errors
  - Alerts

# PI-12 Reliable Notifications: Summary

## 1. Summary

- a) Design provides ← high-level design has been approved by DA\*
  - i. Retry-mechanism
  - ii. Feedback via Delivery-report, reporting capability to enforce SLAs, and support for compensating actions
  - iii. Separation of concerns
    - i. Event-Handler: Core-logic, reporting, compensating actions
    - ii. Command-Handler: Notification-engine, Retry-mechanism

## 2. Next steps?

- a) Detailed design, and implementation in PI13

## 3. How can you help?

- a) Review & comment on PR:  
<https://github.com/mojaloop/documentation/pull/288>

# PI-12: Continuous Gross Settlement (CGS) Overview

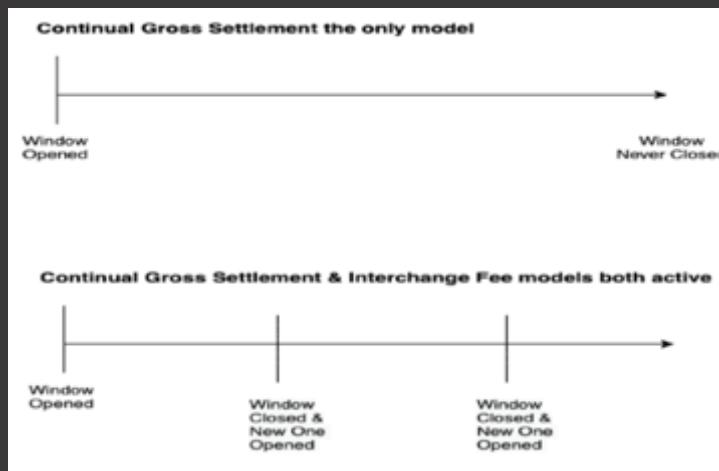
1. CGS is the real-time fashioned process of settling payments on an individual transfer basis
2. Starts as soon as a transfer is committed
3. Once completed, real-time gross settlement payments are final and irrevocable
4. NDC for a DFSP reflects its settled position and will vary with each successful transfer

1. Each DFSP has two accounts
2. Position account holds the not yet settled amount
3. Settlement account is NDC limit – Liquidity available
4. After COMMIT of a transfer, a message has been placed on a queue and is processed as soon as its consumed.  
Interchange fee calculation might happen at this stage of the process

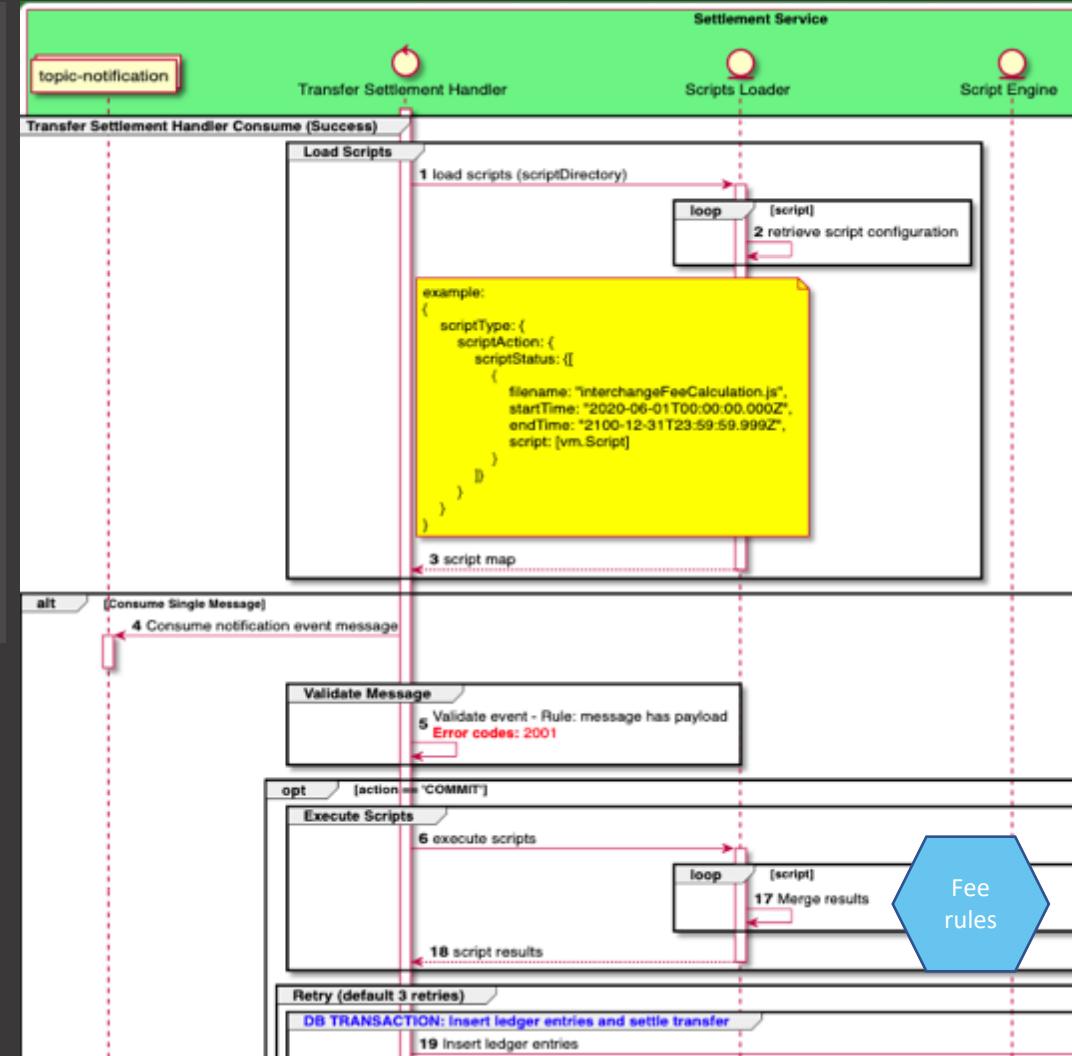


# PI-12: Interchange Fees Overview

1. Interchange fee settlement process is the same as Deferred Net Settlement strategy without the need of Liquidity check (required by the implementation)
2. The amounts due are calculated per transfer based on rules
3. The accounts are settled between DFSPs
4. The settlement process starts with closing of a given time period (window)
5. The process sums up all the totals for the selected settlement windows, using the Multilateral Deferred Net settlement Method
6. Once the Interchange fee bills have been approved then the Settlement Status is updated to SETTLED



Phase-5 PI-13



# CGS, Interchange fees: PI-12 Core Updates

## Central-ledger changes

1. Adding settlement model with mandatory currency
2. Create participants accounts per settlement model
3. Refactoring and code alignment to the OSS standards
4. PRs:
  1. <https://github.com/mojaloop/central-ledger/pull/808>
  2. <https://github.com/mojaloop/central-ledger/pull/811>

## Central-settlement changes

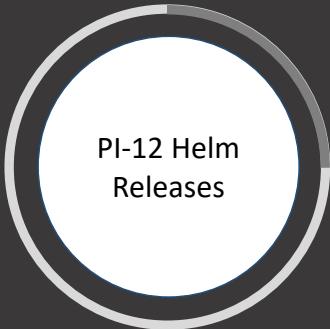
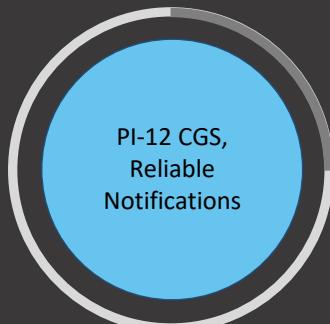
1. Split handlers into separate services
2. Refactoring of operational and deploy configs
3. PR: <https://github.com/mojaloop/central-settlement/pull/338>

## Roadmap

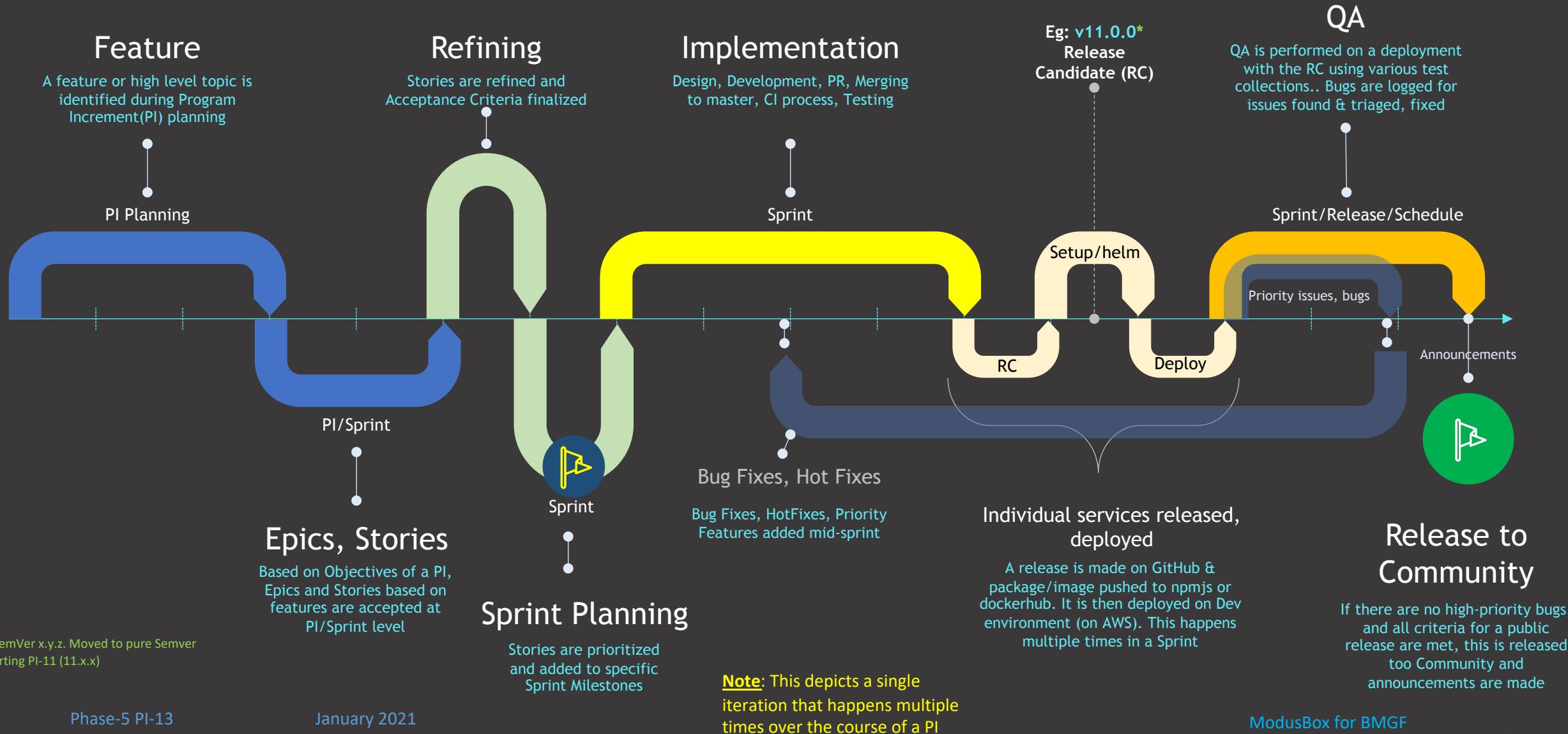
1. Separate rules engine from CGS handler
2. Refactor central-settlements domain and models to align with OSS design standards
3. Implement Event-SDK into central-settlements
4. Test scenarios in TTK for CGS

# Mojaloop PI-13

Phase-5 One Loop for All: Mojaloop in Motion



# ML OSS: (Helm) Release Mechanism



# PI-12 Maintenance: Helm

## 1. Helm Release v11.0.0 (Available)

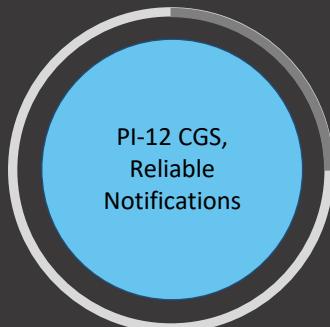
1. Mojaloop-Simulator is enabled by default ← utilizing SDK-Scheme-Adapter
2. Mojaloop Testing Tool-Kit (TTK is enabled by default)
  - a) Onboarding Hub
  - b) Onboarding Simulators
  - c) Golden-Path Tests
3. Mojaloop TTK Jobs ← Publish reports to S3 + Notification to slack
  - a) On-demand
  - b) Scheduled
  - c) Post-hook for Helm deployments ← Easier verifications, & future CI/CD pluggability
4. Helm Charts updated to support Kubernetes 1.16+ ← previously supported v1.13-1.15 only -  
<https://github.com/mojaloop/helm/issues/219>

## 2. Helm Release v12.0.0 (Release Candidate)

- a) Helm Tests
  - i. Toolkit Helm Tests
  - ii. Postman Helm Tests
- b) Includes Continuous Gross Settlement (CGS) functionality
- a) New Helm chart for the Transfers Handler
- c) Fixes
  - i. Docker-hub throttling ← <https://github.com/mojaloop/design-authority/issues/76>
  - ii. Multiple deployment within same namespace ← <https://github.com/mojaloop/project/issues/1963>

# Mojaloop PI-13

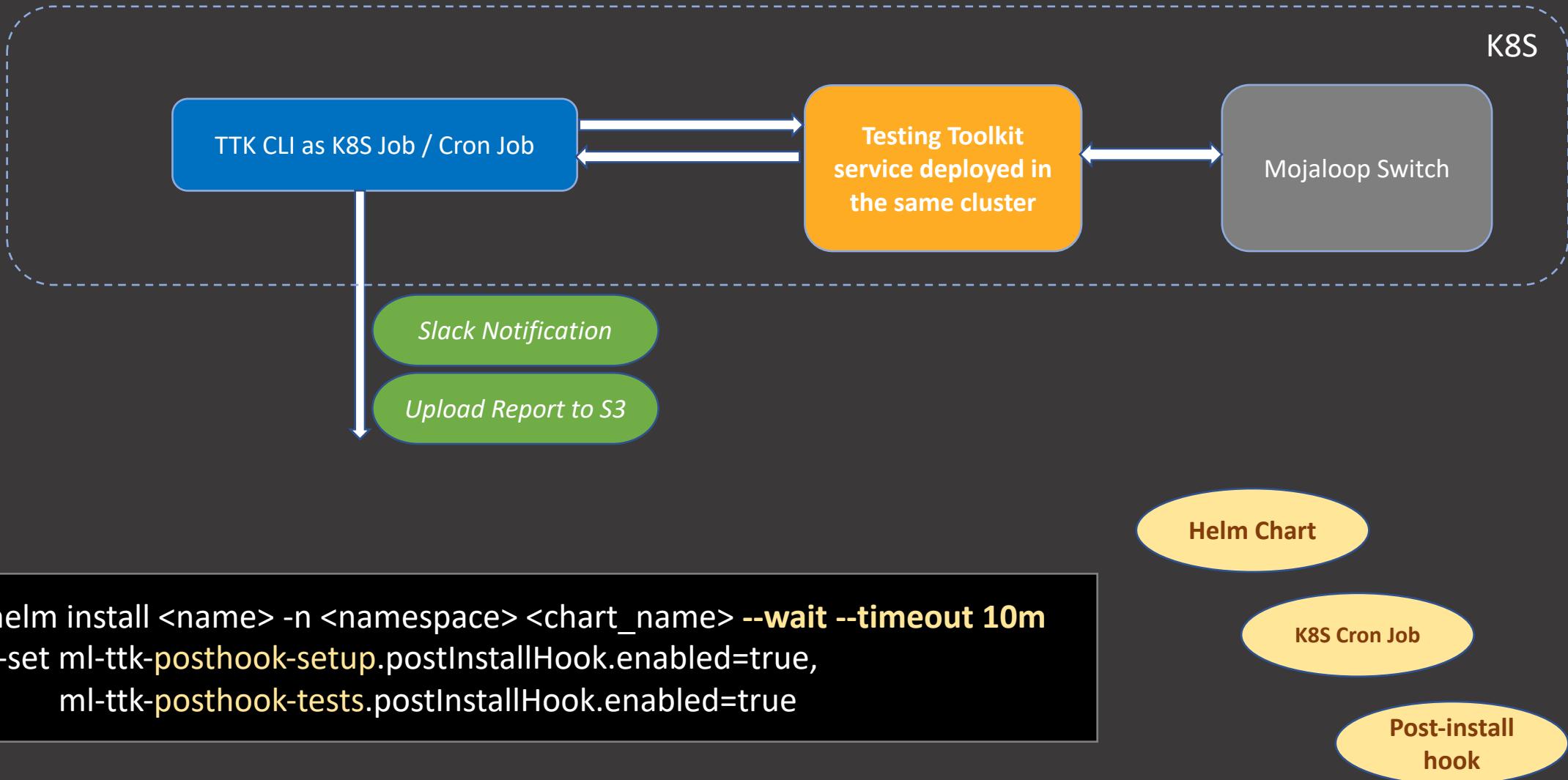
Phase-5 One Loop for All: Mojaloop in Motion



# PI-12: Maintenance Updates

1. Bulk payment functionality – simulator, SDK enhancements
2. v1.1 changes – QA, finalize changes
3. Support for SubID in sdk-scheme-adapter (ALS, GET calls)
4. Bug Fixes
5. PR Reviews, documentation changes

# Scheduled Regression tests using Testing Toolkit (PI-12)



## QA Updates – Bug Fixes

Bug Fixes in PI-12 (Representative below; *a total of ~15 bugs fixed*)

#1878 - Setting up bulk transfers environment (#1884)

#1759 - Helm upgrade v10.4.0 -> v11.0.0 upgrade issues

#1861 - Fix the /documentation & /swagger.json endpoints for account-lookup-service

#1887 - PUT /authorizations/{transactionId} is not validating FSPIOP Headers

#1957 - TTK is not being deployed with published mojaloop helm

#1906 - GET/POST /participants/{type}/{id}/{subId} & /parties/{type}/{id}/{subId} not working in ALS

# Thank you! Q&A on discourse

## Key items for roadmap

1. Decision on the Performance / Scalability PoC – Product readiness, differentiation & NFR
2. Cascaded timeouts, Reliable notification mechanism – Product readiness, differentiation & Support  
Live deployments
3. Business Operations Framework – Product readiness, differentiation & Accelerate Business/Dev onboarding
4. Standardize Admin API – Product readiness, differentiation & Support Live deployments