Mojaloop PI-11

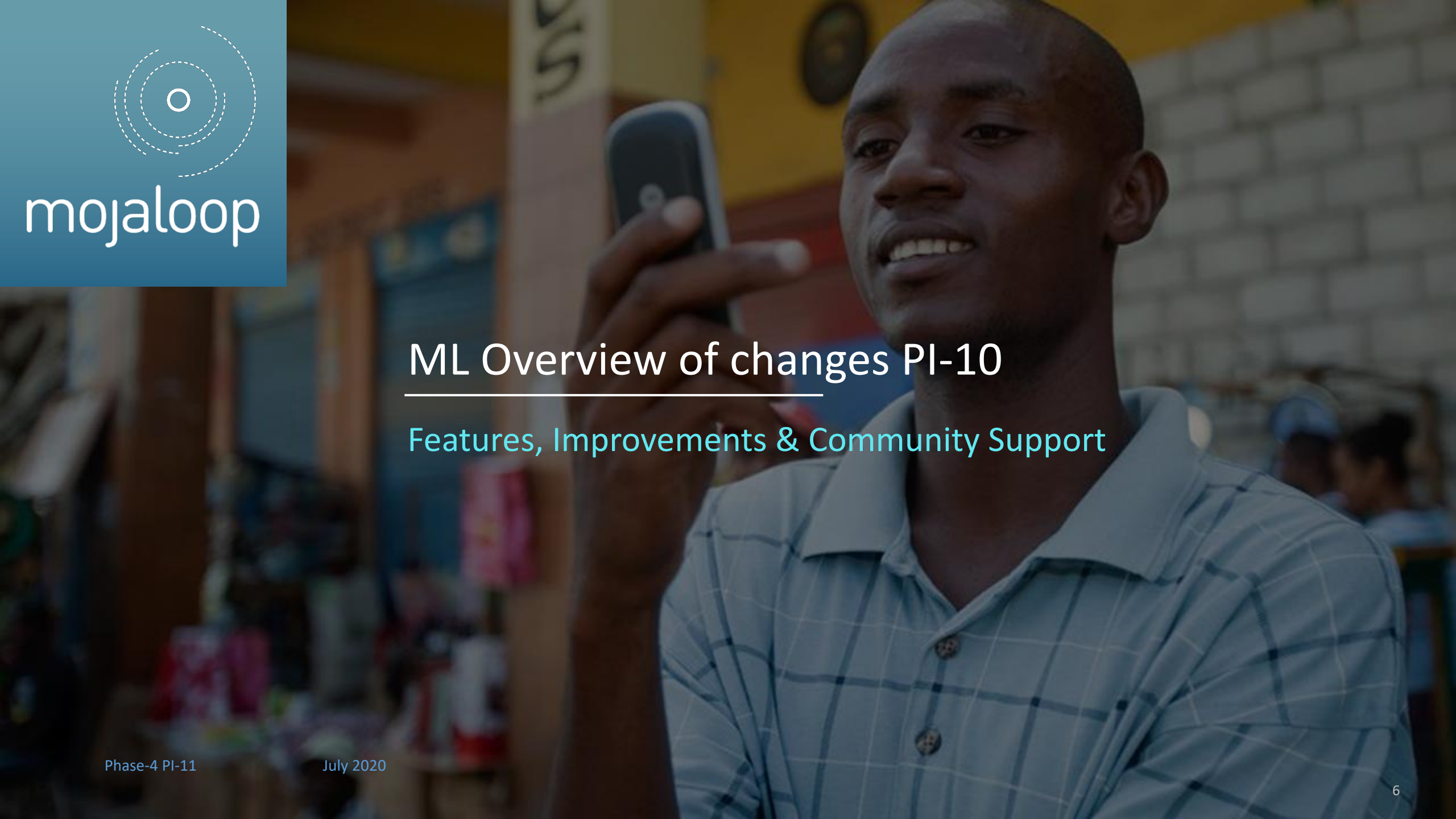
Phase-4 Going live!

Mojaloop PI-11

Phase-4 Going Live!

PI-10 Overview: Agenda

1. PI-10 Overview of changes and progress
2. OSS Community, Collaboration Updates
3. PI-10 Focus Areas



ML Overview of changes PI-10

Features, Improvements & Community Support

Mojaloop PIs Overview

Timeline	Summary
Phase-1 (2016 - 17)	**Level One Project** • Reference Implementation • 6 Program Increments (PIs)
Phase-2 (2018)	**Road To Productionization** • 1 – 4 Program Increments
Phase-3 (2019 Jan - Dec)	**Supporting Adoption & Deployment** • PI-5 (Feb – April): Account lookup, QA Framework, Streamlined CI, Release process, Error endpoints, Documentation, Node Upgrade, Bug Fixes & Community support, Bulk Transfers Design • PI-6 Event handling framework, Bulk Transfers PoC, API Gateway, OSS Settlements API, Quoting Service, ALS • PI-7 Event & Error Handling framework, Packaging, OSS Settlements, Performance testing capabilities, QA • PI-8 Consolidation, Performance, Community Support
Phase-4 (2020 Jan - Dec)	**Going Live** • PI-9: Performance Testing & Improvements, Merchant Request to Pay, Operational Monitoring, Testing toolkit, Settlement v2 • **PI-10**: Performance PoC, Standardizing Bulk Transfers, Testing toolkit, Settlement v2, Adopting FSPIOP API v1.1, Versioning • **PI-11** (July – October) • PI-12

Switch Functionality – Mojaloop Use Cases (Phase-4 PI-10)

Mojaloop FSPIOP API v1.0 – 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 (Phase-4 PI-10)

Mojaloop v1.0: Focus Use-cases

1. P2P
2. Merchant 'Request to Pay'
3. Bulk Payments

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

Bulk Payments*

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

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

* Enhancements done in PI-10

Switch Functionality: Mojaloop End-points (PI-9 → PI-10)

Mojaloop v1.0 – FSPIOP 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
[●] FSPIOP API v1.1 changes
[●] Fully implemented
[●] PoC / Initial Version
[●] Partially implemented
[●] Not implemented
[○] Future Roadmap

* Enhancements in PI-10
\# FSPIOP API v1.1 changes implemented
^ Caching Enhancements

PI-10 Switch Functionality – Operations

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

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
[o] Roadmap for PI10

PI-10 Switch Functionality – Settlement

Settlements v**1.0** [Supports some forms]

[●] 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

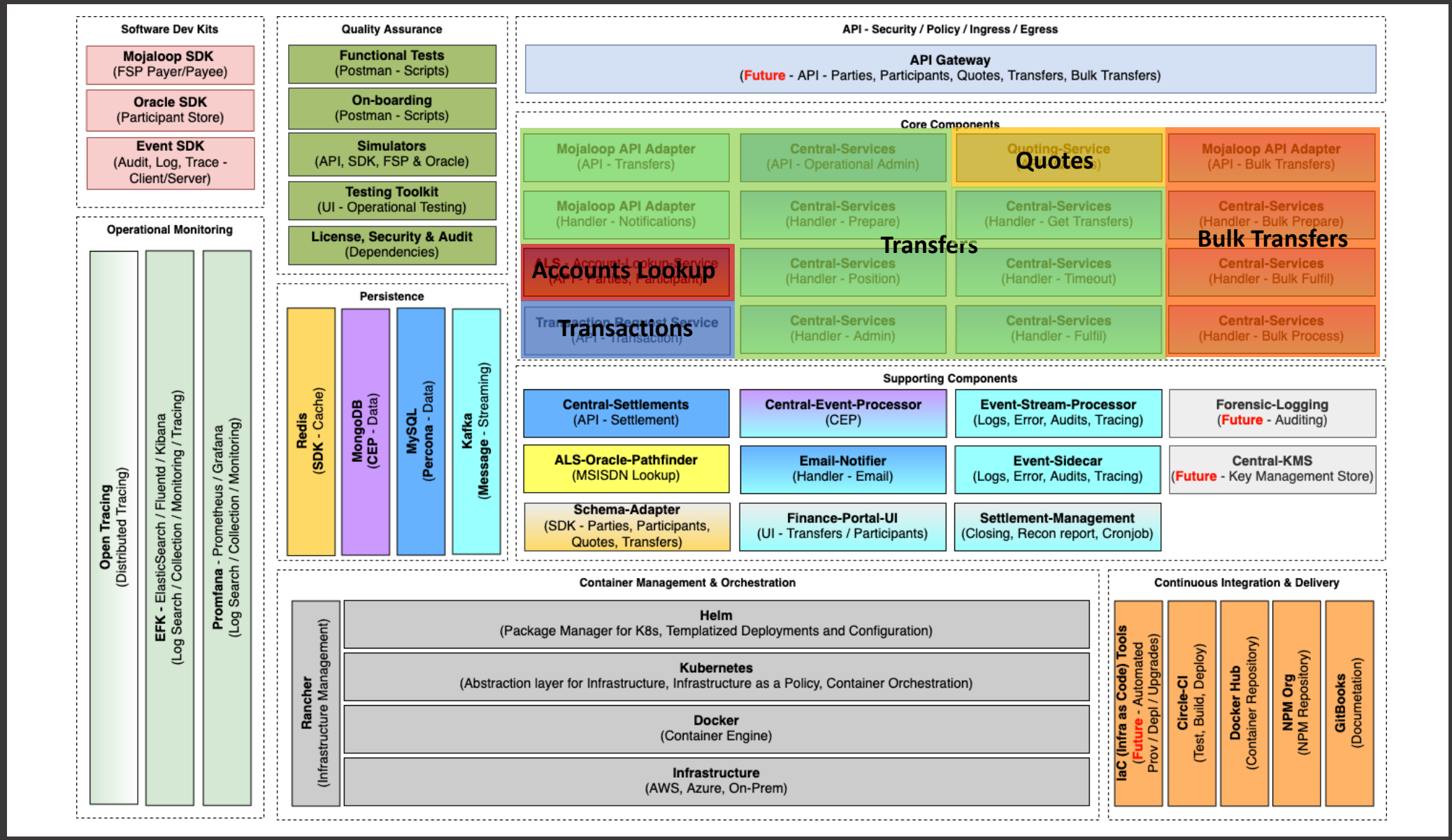
[●] Query Positions
[●] Manage Positions
  [●] Create Initial Value
  [●] Query
  [●] Update

Settlements v**2**

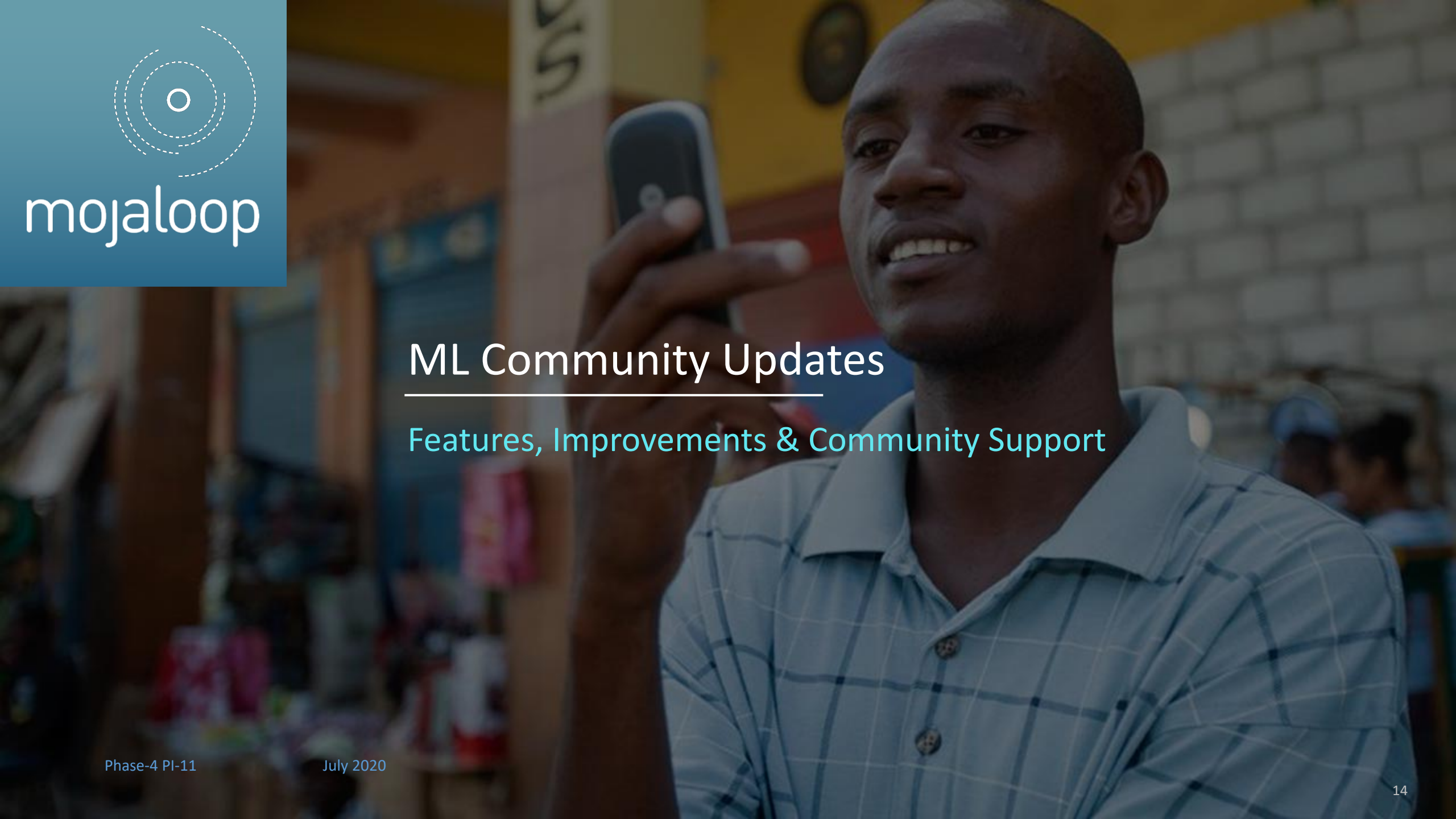
[●] Settlement models
[●] Designs
[●→●] Settlement by Currency
[○→●] Interchange Fees
[○→●] Continuous Gross Settlement

Key
[●] Fully implemented
[●] PoC / Initial Version
[●] Partially implemented
[●] Not implemented
[○] Roadmap for PI10

PI-10: Component Architecture



* Color-shading of core & supporting components match dependent persistent stores.



ML Community Updates

Features, Improvements & Community Support

PI-10 Overview: Mojaloop OSS Community

Contributions, Collaboration

1. Cross network/currency
2. Code quality and Security improvements
3. Versioning Standards
4. ISO Integration
5. Quality Assurance (bugs, coverage, Tests with mojaloop-simulator)
6. Enable JWS services on outbound messages
7. Settlement v2
8. PISP Solution

Mojaloop OSS PI-10: Community

1. Change Control Board (CCB)
2. Design Authority (DA)
3. Weekly scrum-of-scrums
4. Slack channels: #*general*, #*announcements* , #*help-mojaloop*, #*design-authority* , #*ml-oss-devs*

ML OSS Community: Design Authority

1. Responsibilities
2. Membership – driven by contribution
3. Frequency of meetings, boards used
4. Functioning overview

DA Issue Board

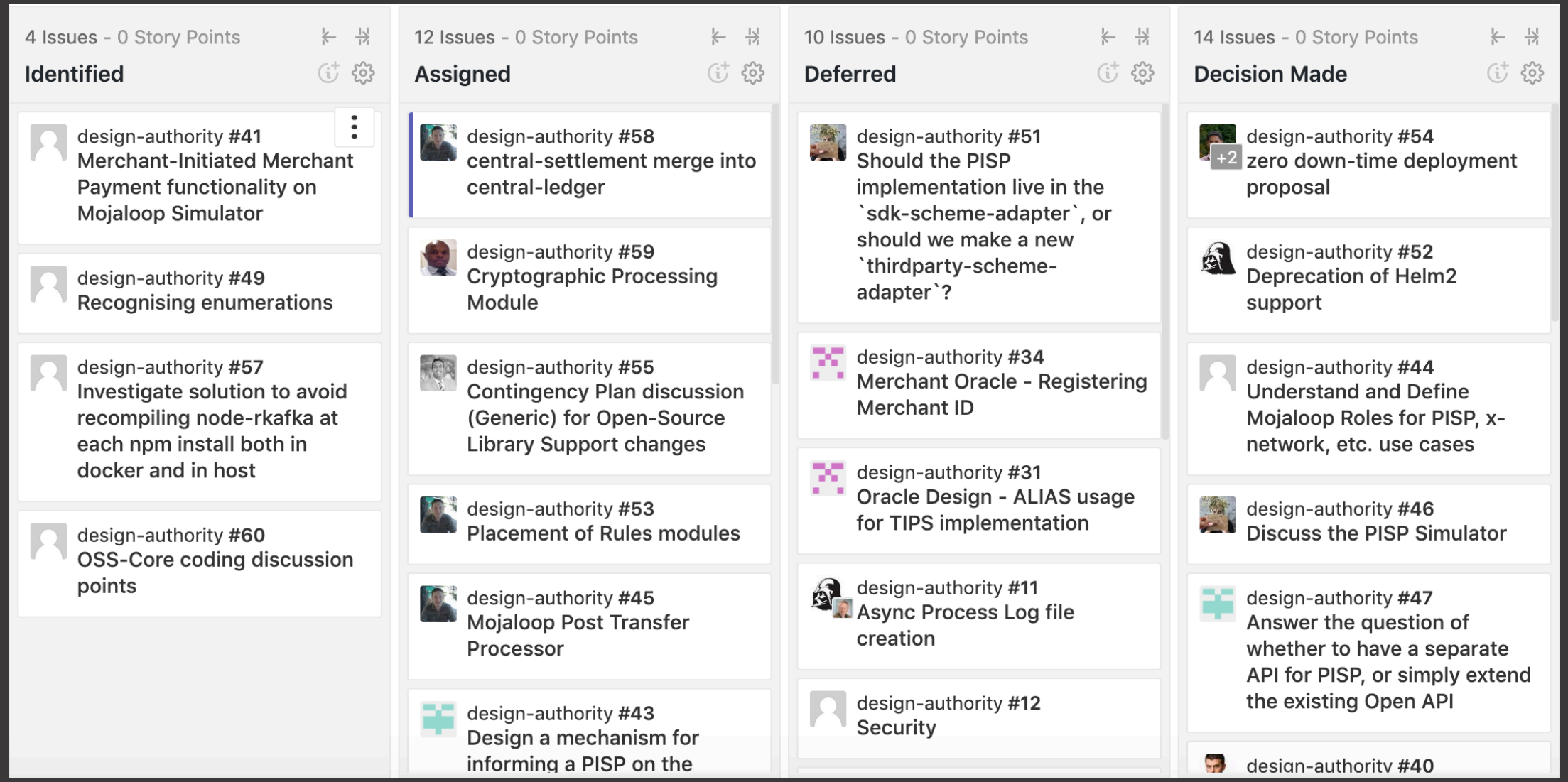
https://tinyurl.com/y6bnj7sz

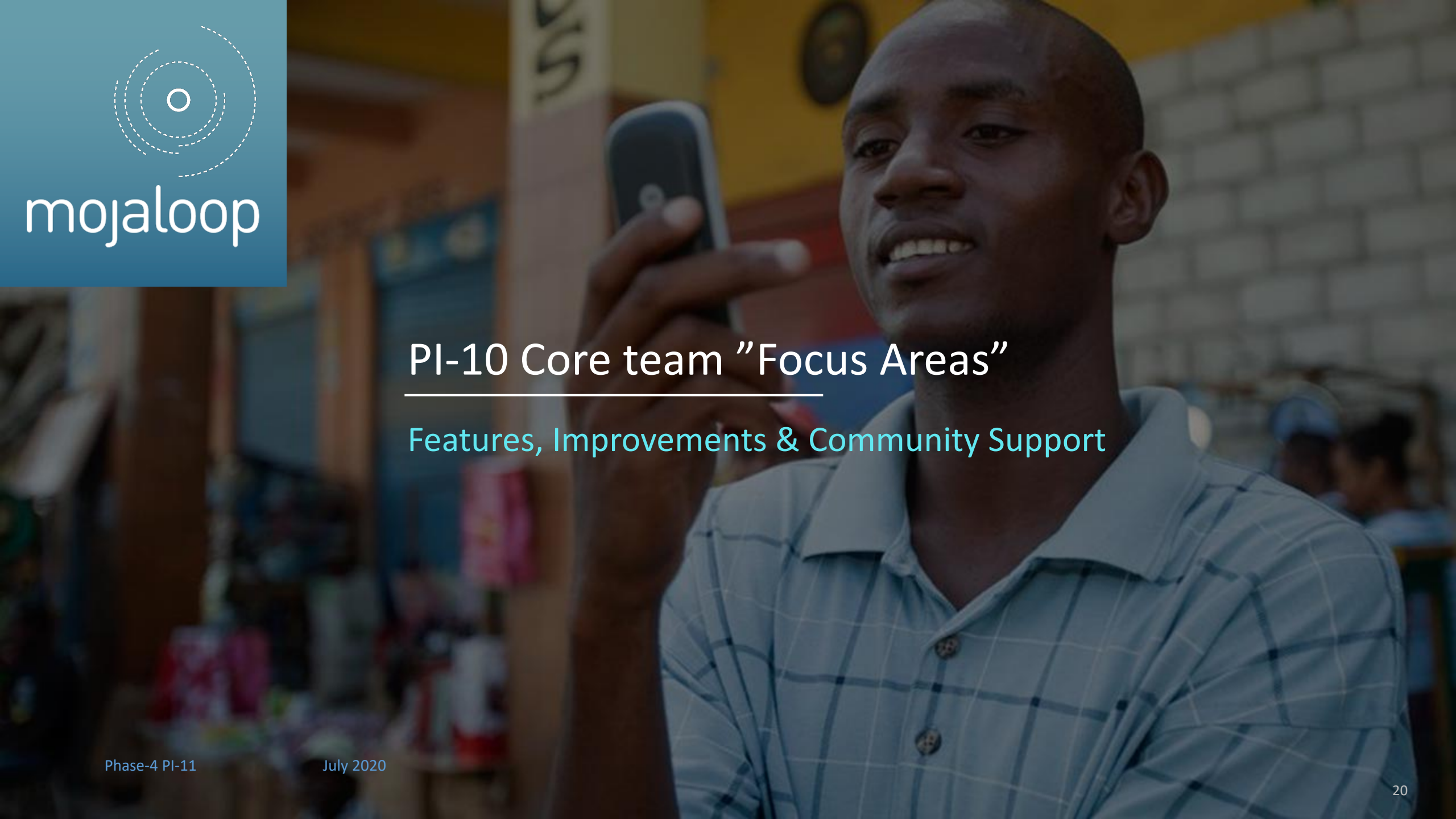
ML OSS Community: Design Authority

Responsibilities

- Ensure a uniform Architecture
- Provide a channel for bringing ideas for Design and Verification
- Define Technical Strategies
- Verify Architectural Standards
- Drive Design Methodologies

ML OSS Community: DA issues overview





PI-10 Core team ”Focus Areas”

Features, Improvements & Community Support

PI-10 Overview: Focus Areas

1. FSPIOP API v1.1 changes: Implementation
2. Maintenance: Helm releases, Support for accented characters, QA
3. Bulk Transfers: Standardization, -ve scenarios, Query
4. Testing toolkit: Support for - Hub testing, usability and adoption
5. Performance: Perf PoC, Support streaming improvements

PI-10: Performance

Completed/In-Progress

1. Kafka Workshops with Confluent
   a) Best-practices & Learnings on Kafka Architecture & deployment / infrastructure
2. Perf Environment
   a) Automated config (collaboration)
   b) Automated startup/shutdown (collaboration)
3. Mojaloop Kafka Client (Central-Services-Stream)
   a) Client Instrumentation: Monitoring Producer/Consumer internal queues (fetch, send lag), number of messages read/send, etc
   b) Provided support & verification of client queue management and parallel optimizations (collaboration)
4. Proof of Concept (PoC) utilizing Domain-Driven-Design, Event-driven CQRS & Event-Sourcing Architecture (collaboration)

Roadmap

1. Perf Environment
   a) Implement Best-Practices/learnings from Kafka Workshops (Infrastructure as Code)
   b) Automated config (roll-out)
   c) Automated startup/shutdown (roll-out)
2. Mojaloop Kafka Client (Central-Services-Stream)
   a) Optimized Producer from (PoC)
      a) High-level producer implementation (improved resilience) & config (improved performance)
   b) Optimized Consumer from (PoC)
      a) Simplified consumer implementation (improved maintainability & performance) & config
   c) Instrumentations
      a) Client Monitoring (Pull-request pending)
      b) Dashboards

Mojaloop PI-11

Phase-4 Going Live!

FSPIOP API v1.1 Changes - 1: Support Payee FSP Notification

[●] Option to notify completion of a transfer to a Payee - Story#1334

Behaviour in v1.0

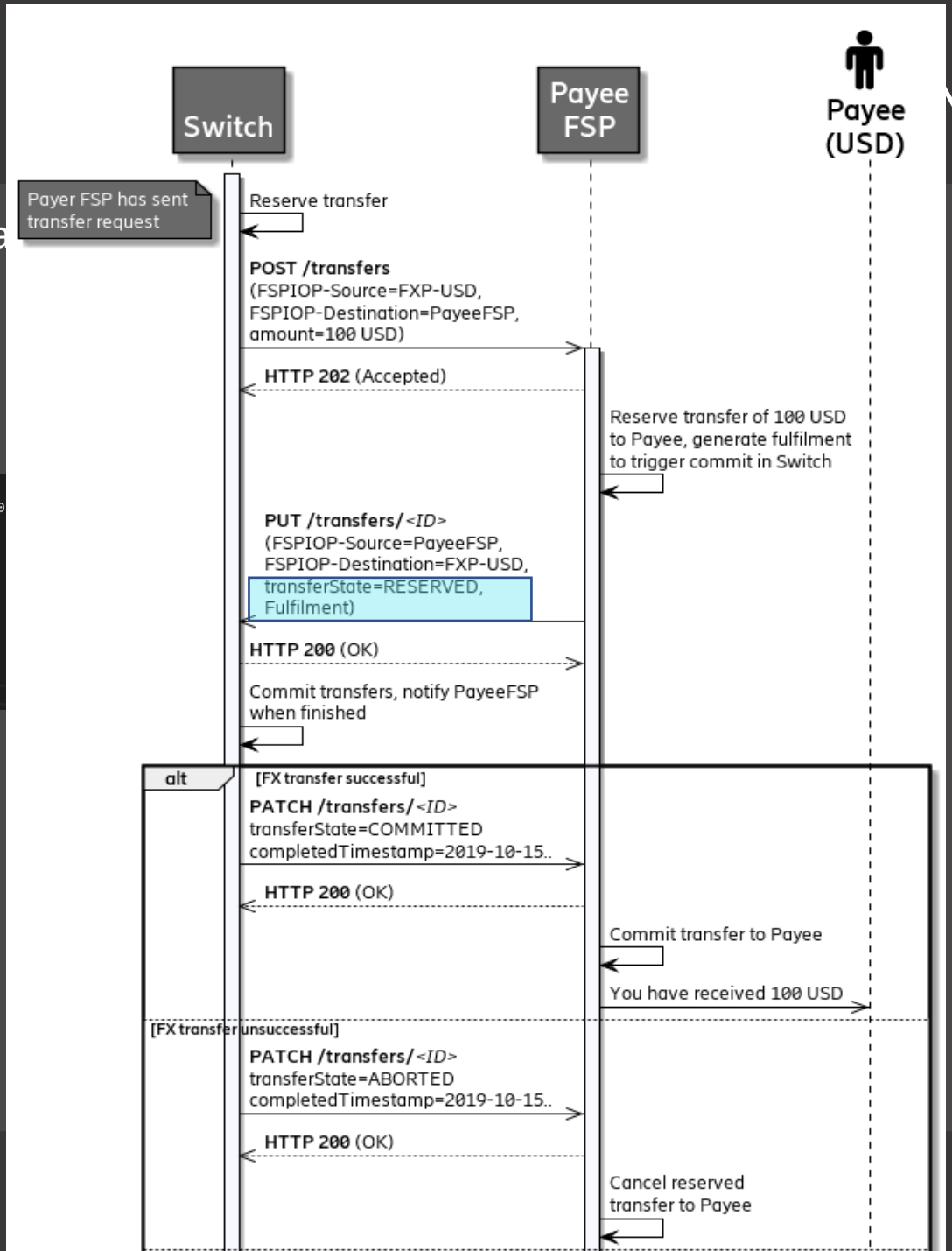
The Payee FSP needs to send a *GET /transfers/<ID>* message to the Switch to know the status

Desired behaviour

The Payee FSP can send the *PUT /transfers/<ID>* to the Switch with an option to indicate that it would like to receive a notification when the transfer is finalized in the Switch.

FSPIOP Notification

[●] Option to notify completion of a




```
PUT /transfers/85feac2f-39b2-491b-817e-4a03203d4f14 HTTP/1.1
Content-Type: application/vnd.interoperability.transfers+json;version=1.1
Content-Length: 166
Date: Tue, 15 Oct 2019 08:14:02 GMT
FSPIOP-Source: DFSP2
FSPIOP-Destination: DFSP1
FSPIOP-Signature: {"signature": "YXBwbGlj...JzK2pzb2"}
{
  "fulfilment": "mhPUT9ZAwd-BXLfeSd7-YPh46rBWRNBiTCSWjpku90s",
  "transferState": "RESERVED",
}
```

```
PATCH /transfers/85feac2f-39b2-491b-817e-4a03203d4f14 HTTP/1.1
Content-Type: application/vnd.interoperability.transfers+json;version=1.0
Content-Length: 166
Date: Tue, 15 Oct 2019 08:14:03 GMT
FSPIOP-Source: Switch
FSPIOP-Destination: DFSP2
FSPIOP-Signature: {"signature": "YXBwbGlj...JzK2pzb2"}
{
  "completedTimestamp": "2019-10-15T08:14:03.113+01:00",
  "transferState": "COMMITTED",
}
```

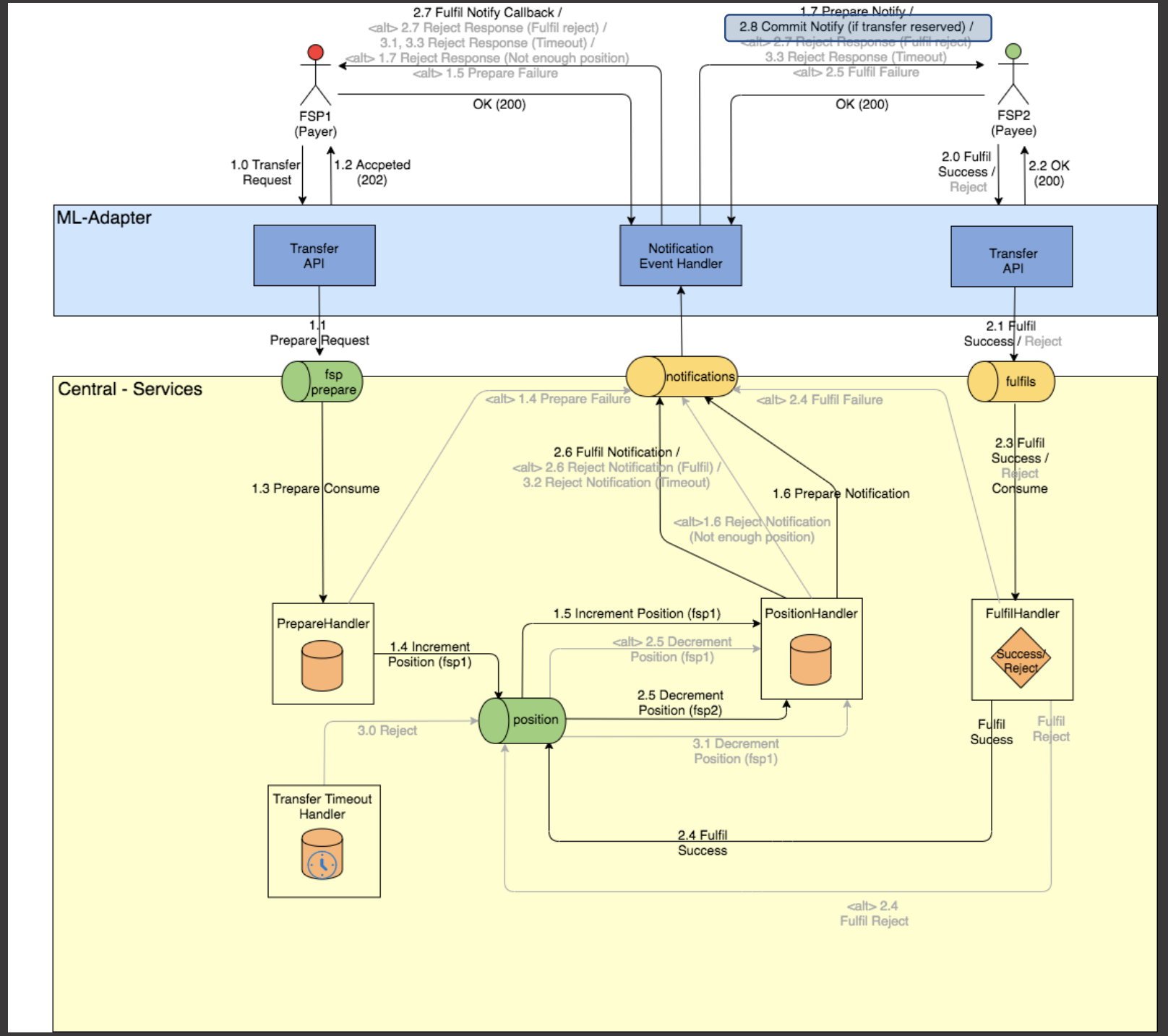
Updated central services

1. central-services-shared
2. central-ledger
   1. Fulfil handler
   2. Position handler
3. ml-api-adapter

Supporting updates

1. Simulator(s)
2. sdk-scheme-adapter
3. Postman collections
4. Documentation

PI-10: Transfers Architecture v1.1



FSPIOP API v1.1 Changes - 2: Use error callbacks for rejections

[●] Clarify usage of ABORTED state in the Fulfil step of a transfer - Story#1335

Behaviour in v1.0

Transfer could be aborted by two ways

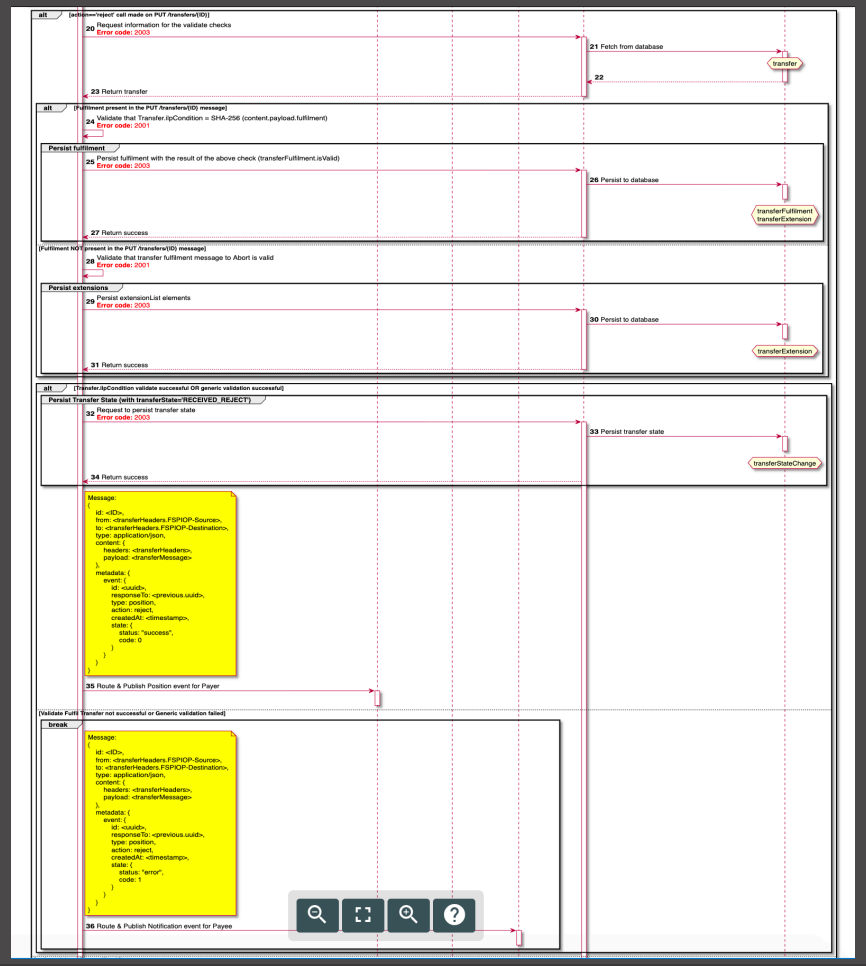
1. *PUT /transfers/{ID}* endpoint with 'transferState' as ABORTED.
2. *PUT /transfers/{ID}/error* endpoint with additional error information.

Requested Behavior

FSPs to ONLY use *PUT /transfers/{ID}/error* to abort / reject transfers during a Fulfil step

FSPIOP API v1.1 Changes - 2: Use error callbacks for rejections

[●] Clarify usage of ABORTED state in the Fulfil step of a transfer - Story#1335



Updates

1. Central-ledger
2. Postman golden path collection
3. Documentation

FSPIOP API v1.1 Changes - 3: ExtensionList element to */participants* requests

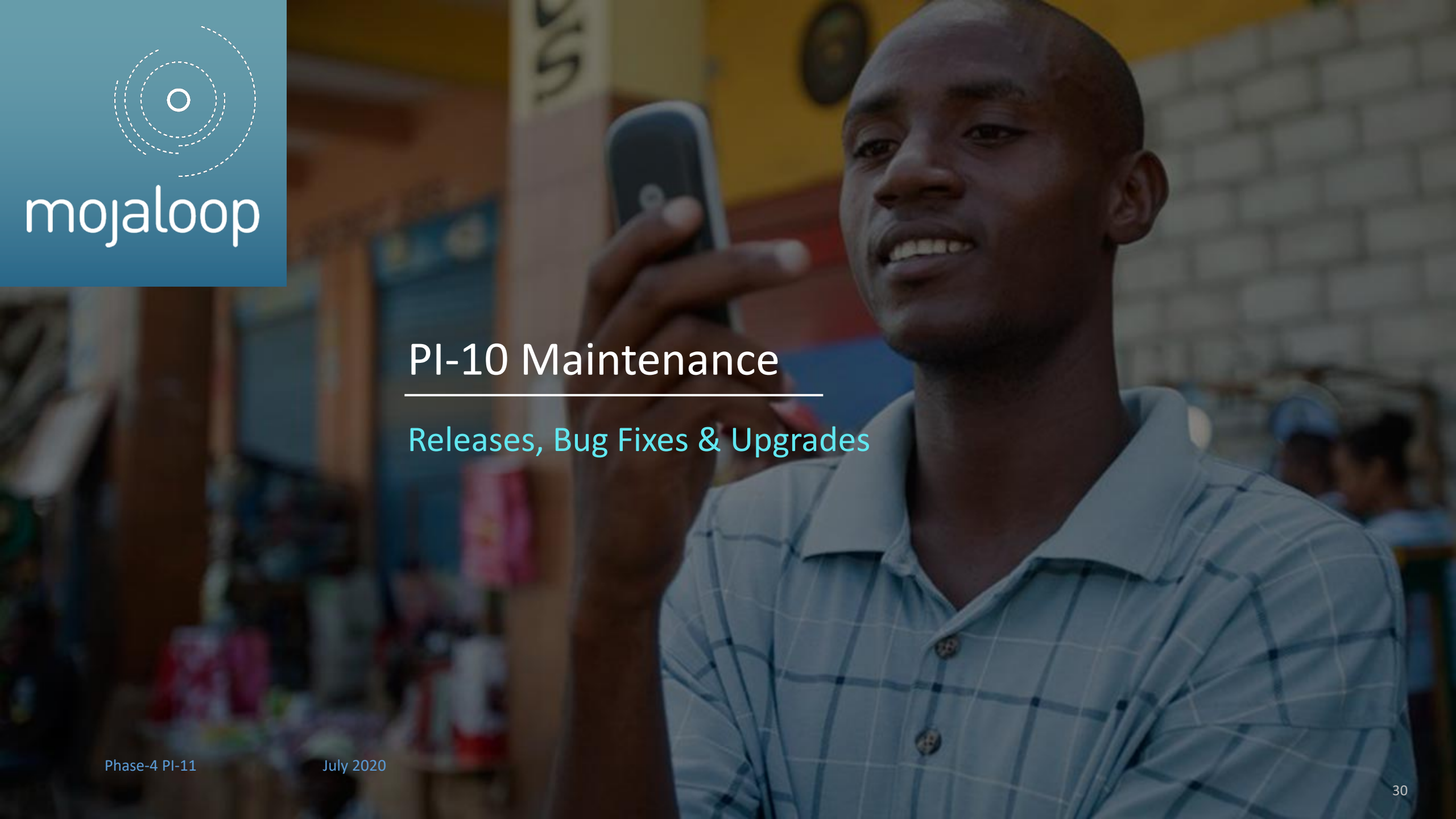
[●] Support for extensionList element in 'partyIdInfo' Object - Story#1336

Behaviour in v1.0

'partyIdInfo' object and POST */participants/{TYPE}/{ID}*, */participants/{TYPE}/{ID}/{SubID}* calls did not have an ExtensionList element

Requested Behavior

Additional data is required for some existing data items in order to meet the requirements of individual implementations. An optional *ExtensionList* element has been added to *'partyIdInfo'* object and POST */participants/{TYPE}/{ID}*, */participants/{TYPE}/{ID}/{SubID}* calls



PI-10 Maintenance

Releases, Bug Fixes & Upgrades

PI-10: Maintenance Updates

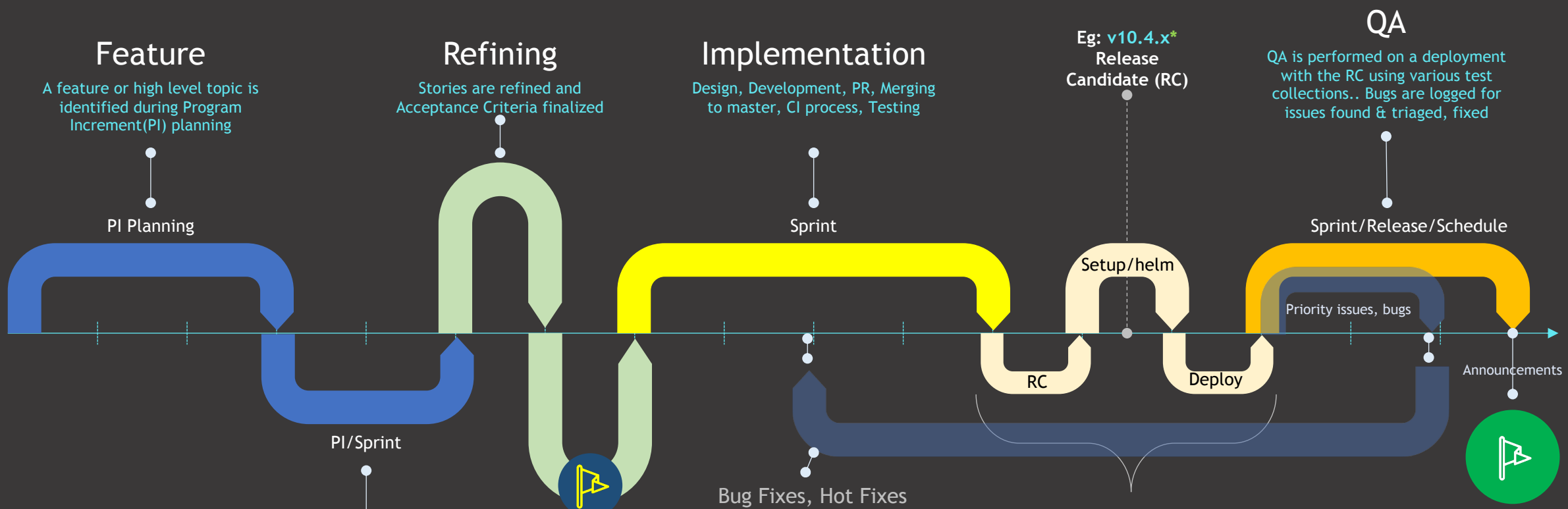
1. Helm Releases, maintenance
2. Support for accented characters in names
3. JWS signing enabled for outward facing interfaces (quoting service, ml-api-adapter, ALS)
4. QA, Bug Fixes
5. On-us transfers support

PI-10: Helm Updates

1. Helm Release v10.1.0*
2. Helm Release v10.4.0
3. Current charts support Helm v3 and v2
4. Backwards compatibility of charts with v2 verified on ML OSS environments

* Discussed on Mojaloop Foundation blog

ML OSS: (Helm) Release Mechanism



Epics, Stories

Based on Objectives of a PI, Epics and Stories based on features are accepted at PI/Sprint level

Sprint Planning

Stories are prioritized and added to specific Sprint Milestones

Individual services released, deployed

A release is made on GitHub & package/image pushed to npmjs or dockerhub. It is then deployed on Dev environment (on AWS). This happens multiple times in a Sprint

Release to Community

If there are no high-priority bugs, and all criteria for a public release are met, this is released too Community and announcements are made

* Inspired by SemVer; x.y.z – where 'x' is the PI number, 'x.y' is the Sprint number and 'z' is the number of release in a Sprint

Note: This depicts a single iteration that happens multiple times over the course of a PI

PI-10: Accented characters in names

1. Allowing names with accent characters in names
    a. Updated regex to attempt match any letter/number from any language script
    b. Current issues with some scripts investigating alternative libraries for regex validation
2. Library: *hapi-openapi -> openapi-backend*
    a. Allows for more control over validations libraries
    b. OpenAPI 3.0 compatible
    c. Works with our current libraries and plugins

QA Updates – Bug Fixes

Major Bug Fixes in PI – 10 (*A total of ~20 bugs fixed*)

#1331 – Unsupported HTTP Method for all resources – fix for 405 status code

#1168 – Timeout error in quoting service

#1378 - Extension List missing when unsupported version requested for services

#1381 - PAYEE flag disables notification to PAYER

#1404 - Accept header is sent in the callbacks for Post/Quotes

#1408 - FSPIOP-URI header missing in PUT /transfers callback

#1412 - Span finished. no further actions allowed ALS

PI-10 QA Updates – Test collections

Postman collections

1. Mojaloop Simulator collections are used for validation, referenced as part of Helm release with Mojaloop-simulator
2. *JWS signing enabled for Switch services, postman collections, mojasims*
3. Legacy simulator collections are used for validation, referenced as part of Helm releases as well
4. Authorizations based tests are included in the golden path test.
5. Bulk transfers collection is provided and used for validating Bulk transfers functionality

Testing toolkit

1. Testing toolkit used to validate and demonstrate
    a. Hub implementation in generalOn-us transfers, v1.1 changes, Bulk transfers

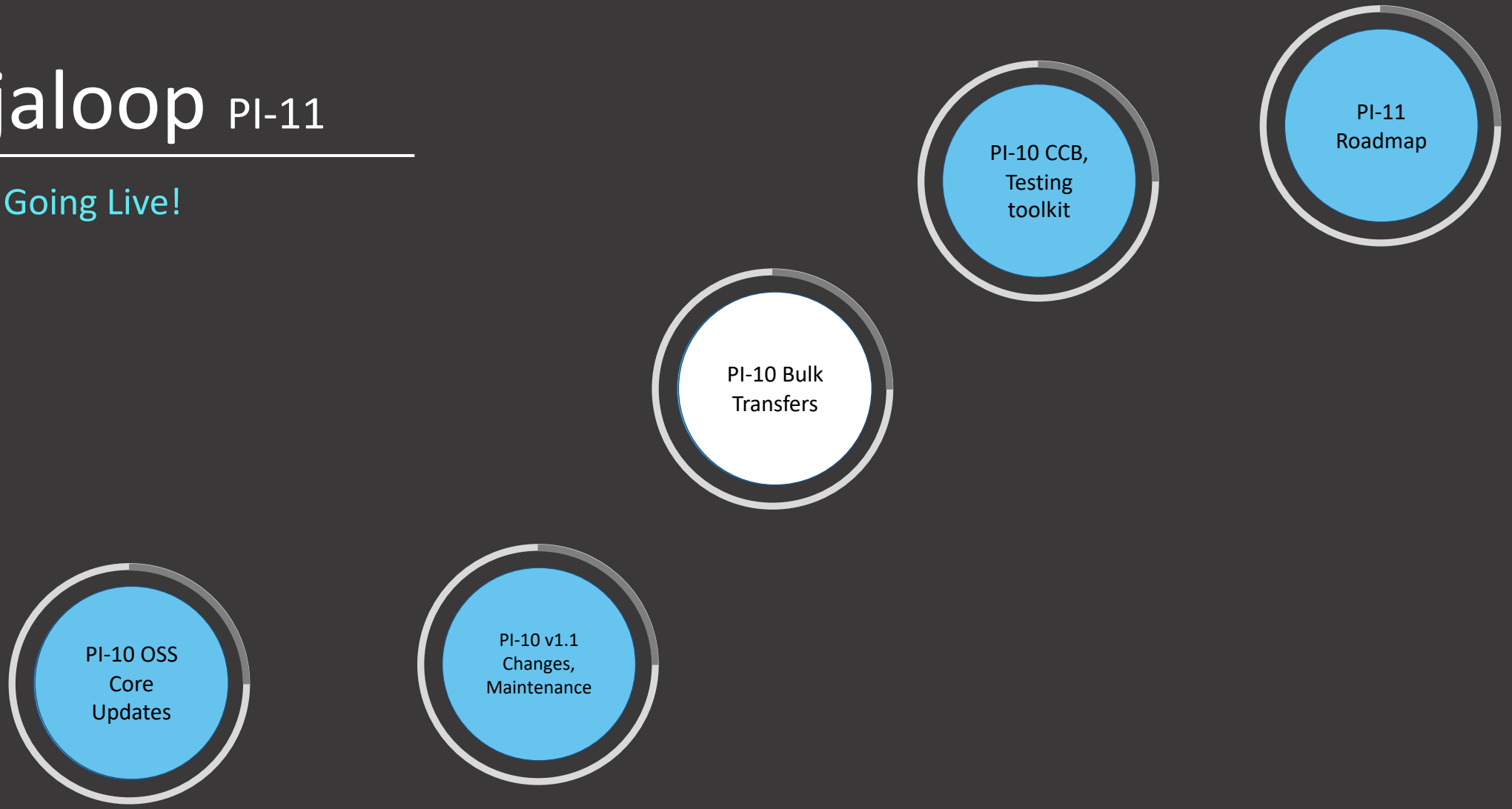
QA Roadmap for PI-11

QA Stream for PI 11 – If prioritized as a high-level item

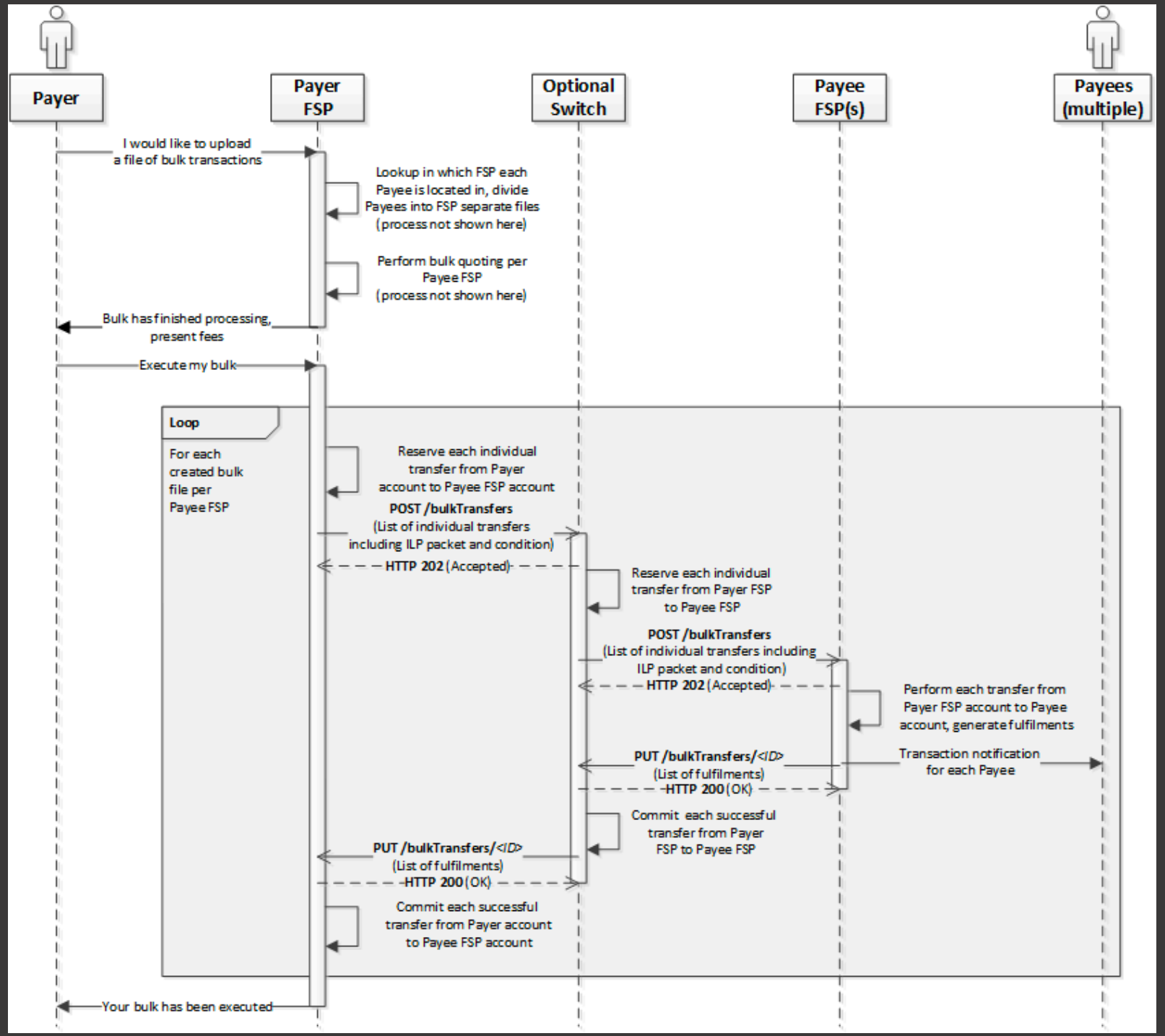
1. Update test collections to correlate with Sequence diagrams for functionality; Update sequence diagrams as needed to add sufficient detail.
2. Improve coverage on features, API Services (double number of tests / coverage)
3. Streamlining non-golden path collections
4. Improve reliability of tests
5. Using testing toolkit to provide at least as much as the current test coverage
6. Port the delta from Legacy Simulator collection to MojaSim collection
7. Cron job regression testing setup with multiple collections (Legacy, MojaSim, Bulk)

Mojaloop PI-11

Phase-4 Going Live!



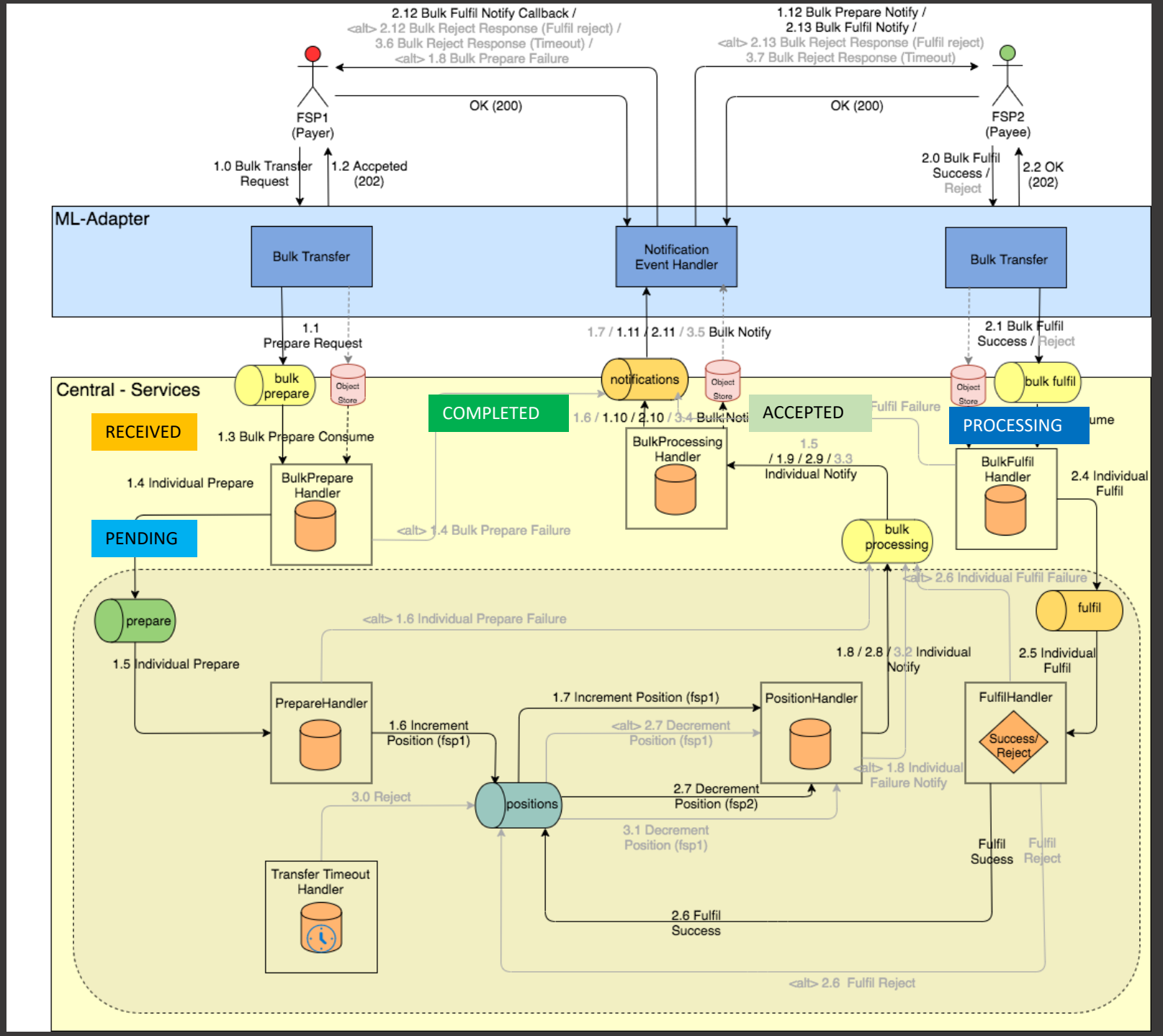
PI-10: Bulk Transfers v1.0



PI-10: Bulk Transfers standardization

1. Bulk Transfers
    a. *PoC done in PI7*
    b. PI-9: Happy Path standardized with tests and Simulator support, Postman tests
2. Standardization in PI-10
    a. Negative scenarios support: *PUT bulkTransfers/{ID}/error*
    b. *GET bulkTransfers/{ID}* functionality with individual responses for Payer and Payee FSPs
    c. *POST /bulkQuotes* and *PUT bulkQuotes/{ID}* implemented now
    d. Support added in Simulators
    e. End to end tests with simulators

PI-10: Bulk Transfers Architecture v1.0



Bulk Transfers: Roadmap

1. Bulk Lookup
    a. If prioritized, design a solution (API level, sequence)
    b. Propose to the CCB as a Change request
2. Bulk make?
    a. An FSP provides a bulk with individual transfers at one go
    b. The Switch takes up the responsibility for all three steps;
        a. Breaking it out into smaller bulks based FSPs after Lookup
        b. Perform bulk quoting (With option to aggregate them and send to Payer FSP)
        c. Perform bulk transfer
3. Configurable options such as all or nothing, etc.

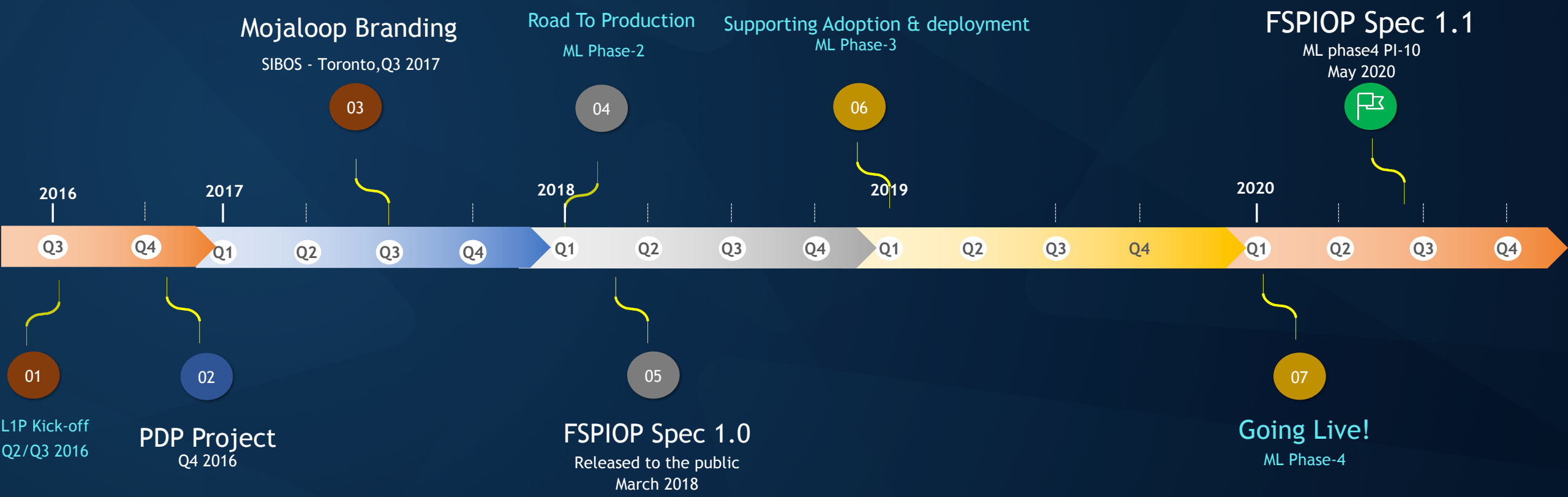
Mojaloop PI-11

Phase-4 Going Live!

ML OSS Community: Change Control Board

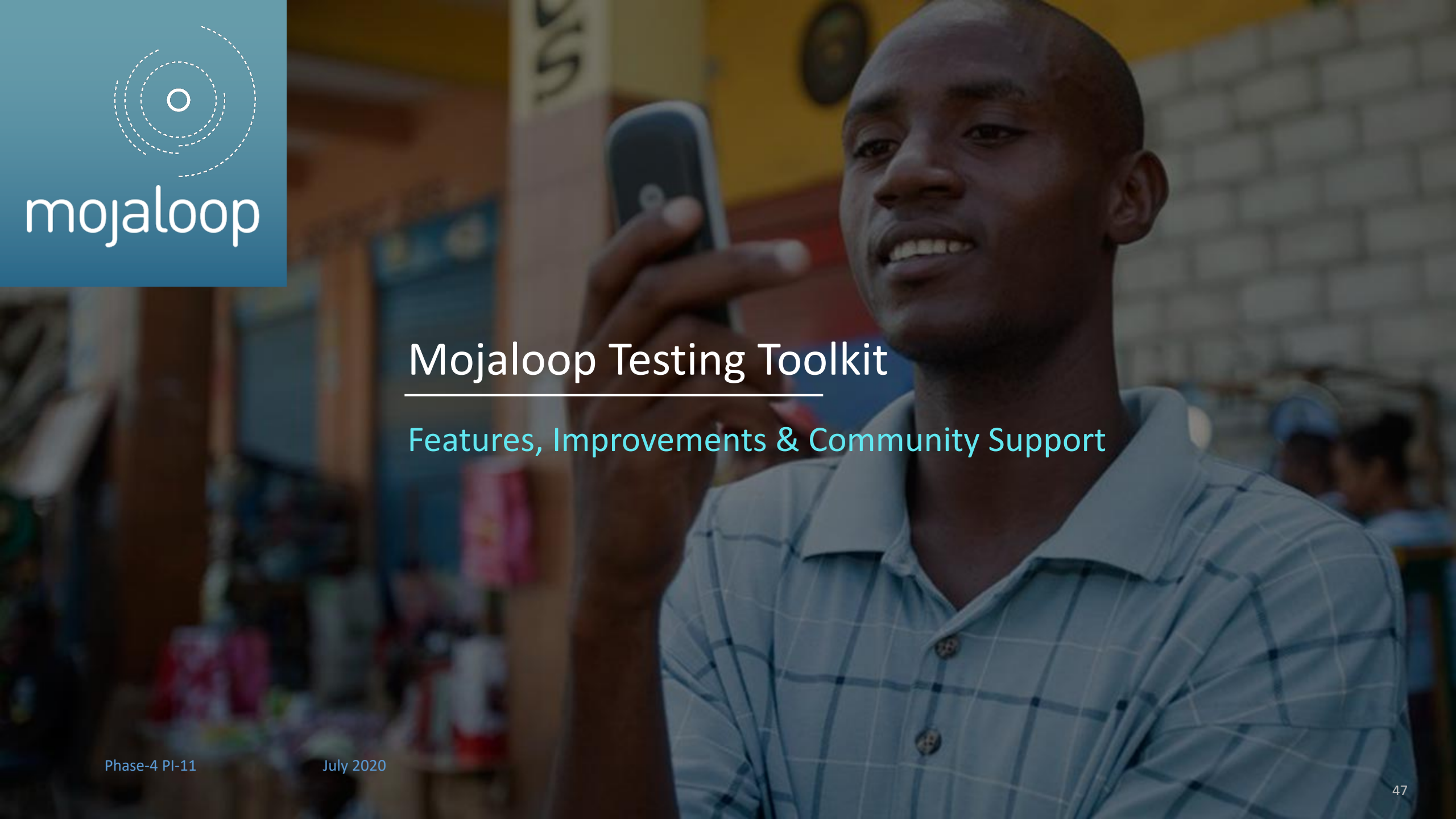
1. Goals, purpose – ML FSPIOP API
2. Current Membership
    i. BMGF (Matt Bohan, Miller Abel)
    ii. Ericsson (Henrik Karlsson)
    iii. Huawei (Chen Hill)
    iv. Mahindra Comviva (Ritvik Sinha)
    v. ModusBox [Non-voting] (Michael Richards, Sam Kummary)
    vi. Mowali (John Mark Ssebunya)
    vii. Telepin (RJ Wilson)
    viii. BoT TIPS (Mutashobya Mushumbusi)
3. Frequency of meetings, boards used
4. Change requests, Solution proposals, Bugs

Mojaloop FSPIOP API: Evolution



CCB – ML FSPIOP API: v1.1 changes

1. Current status:
    a. FSPIOP API v1.1 published in May 2020
    b. Changes in pipeline for v2.0 (PISP, Cross network/currency, other breaking changes)
    c. Separate APIs and boards proposal
2. Overview of changes in v1.1
    a. List of changes: https://github.com/mojaloop/mojaloop-specification/issues/52
    b. Option to notify completion of a transfer to a Payee
    c. Updating description, examples to address inaccuracies and omissions
    d. Clarify usage of ABORTED state in the Fulfil step of a transfer
    e. Describe quote rejection flow
    f. Add notes to indicate the case-insensitive nature of HTTP headers
    g. Clarify usage of FSPIOP-Destination header
    h. Adding ExtensionList element to several items in the data model



Mojaloop Testing Toolkit

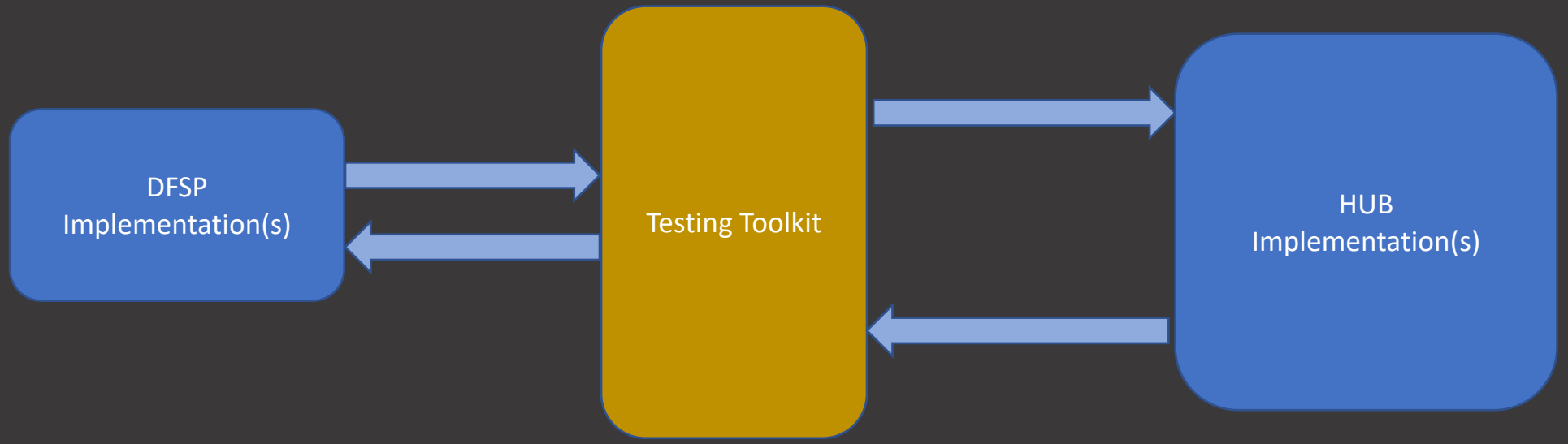
Features, Improvements & Community Support

Phase-4 PI-11 July 2020

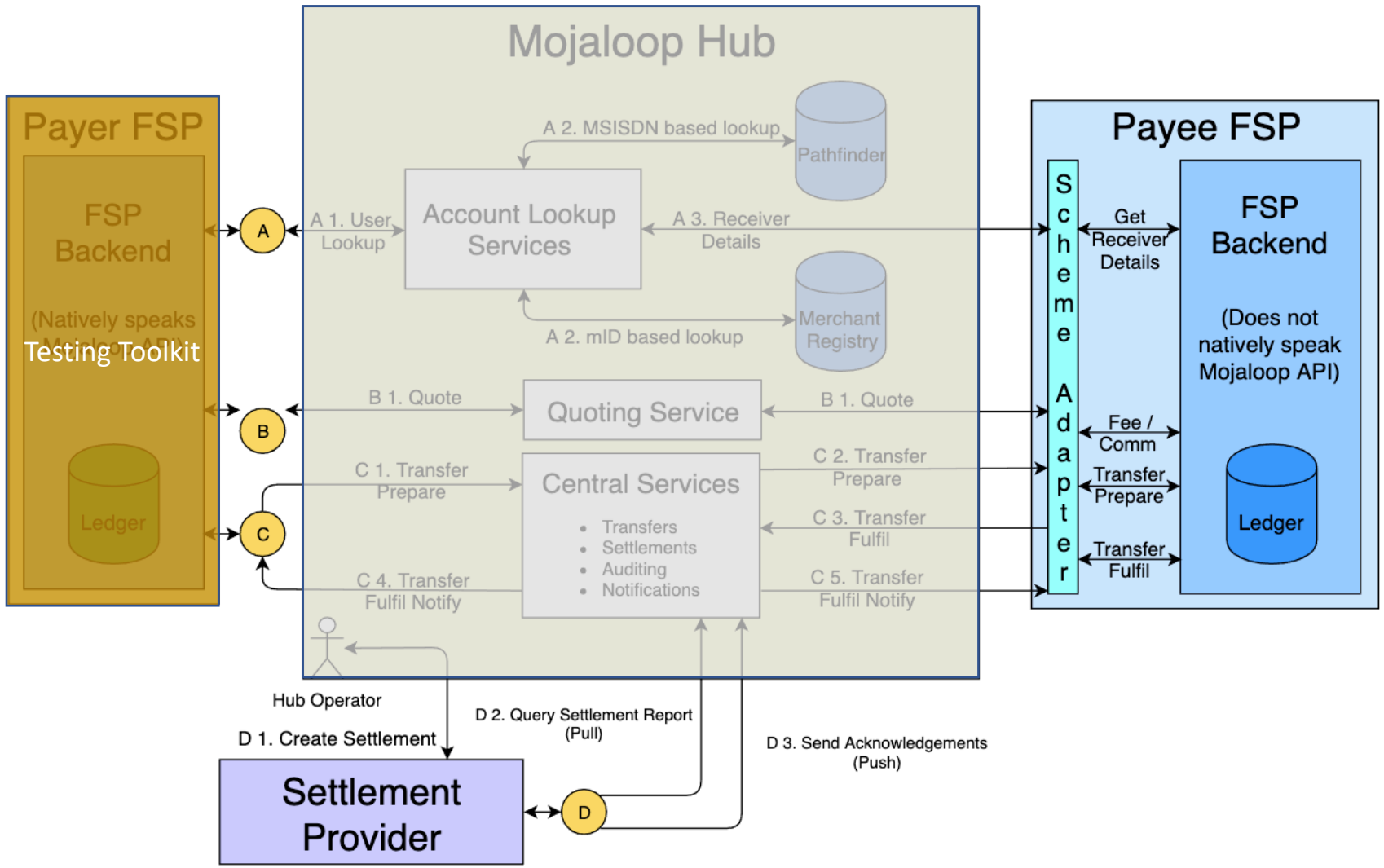
Testing toolkit: Goals

1. Test any Mojaloop FSPIOP API implementation
2. Simple to use
3. Support different versions of Mojaloop API
4. Highly configurable (Configurations portable)
5. Can validate Inbound requests
6. Can generate Outbound requests

What is the Testing Toolkit?



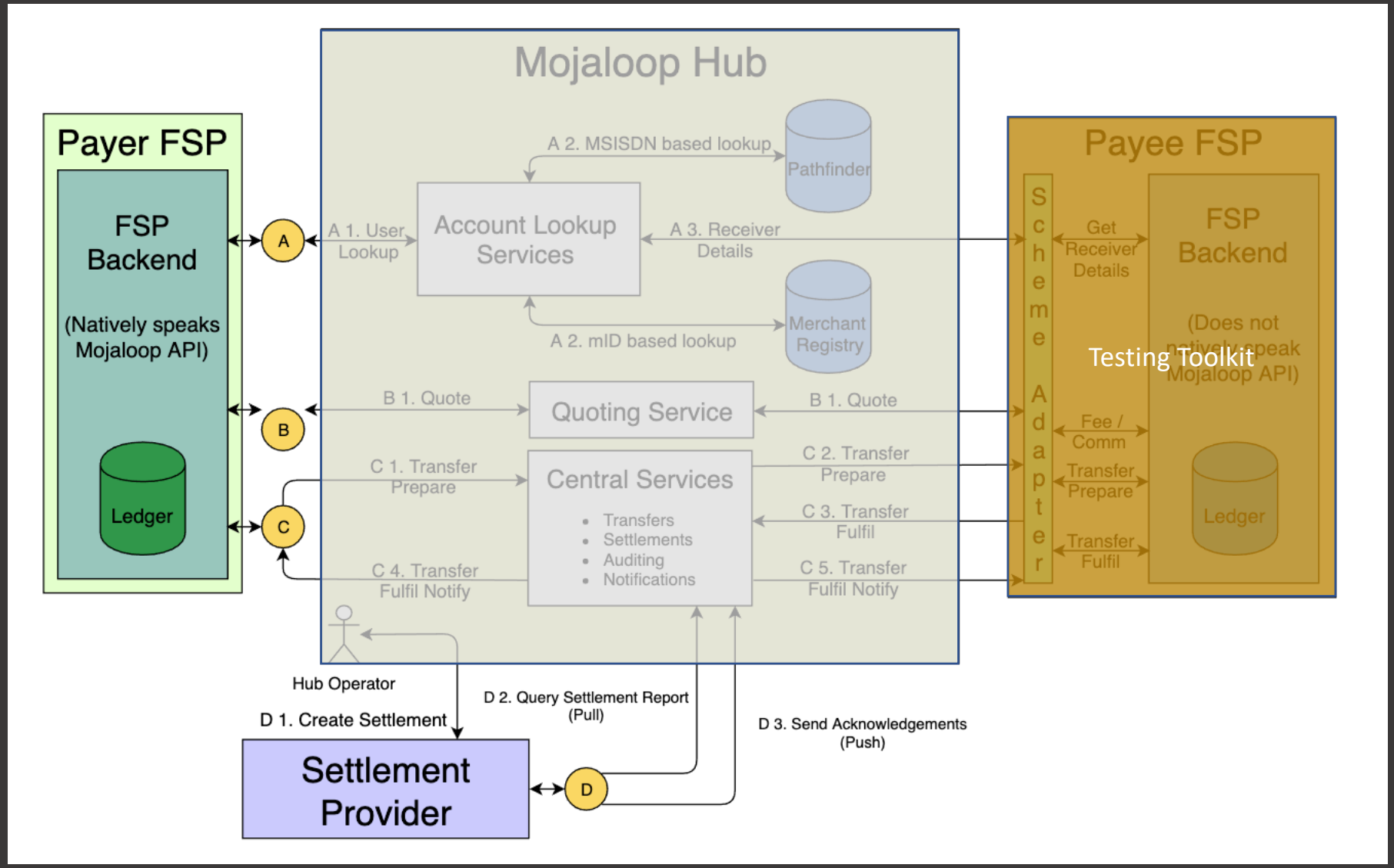
Where does it fit?



Testing Toolkit to HUB

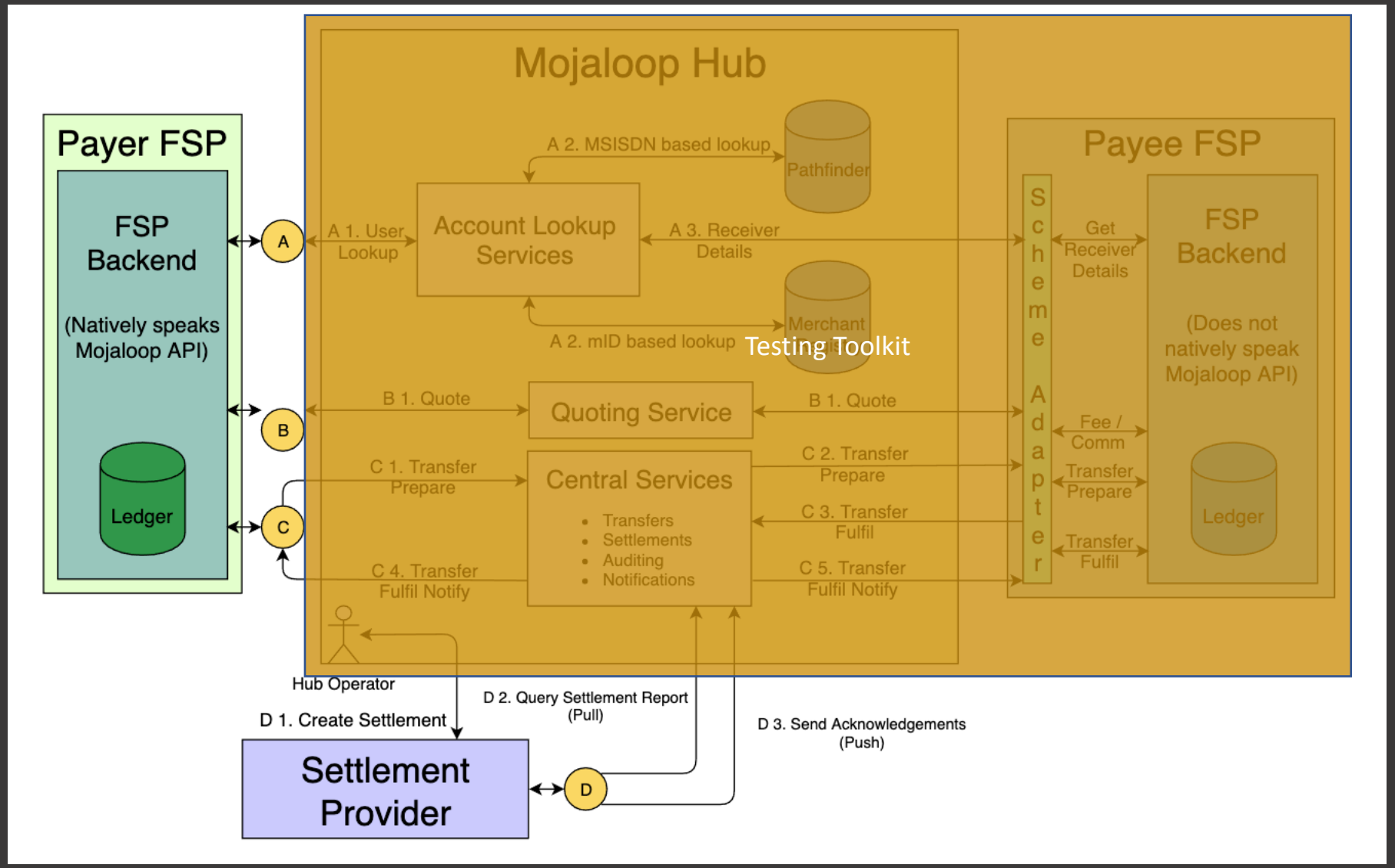
Testing Toolkit to DFSP

Where does it fit?



- Testing Toolkit to HUB
- Testing Toolkit to DFSP
- HUB to Testing Toolkit
- DFSP to Testing Toolkit

Where does it fit?



- Testing Toolkit to HUB
- Testing Toolkit to DFSP
- HUB to Testing Toolkit
- DFSP to Testing Toolkit
- As a Hub and Payee FSP

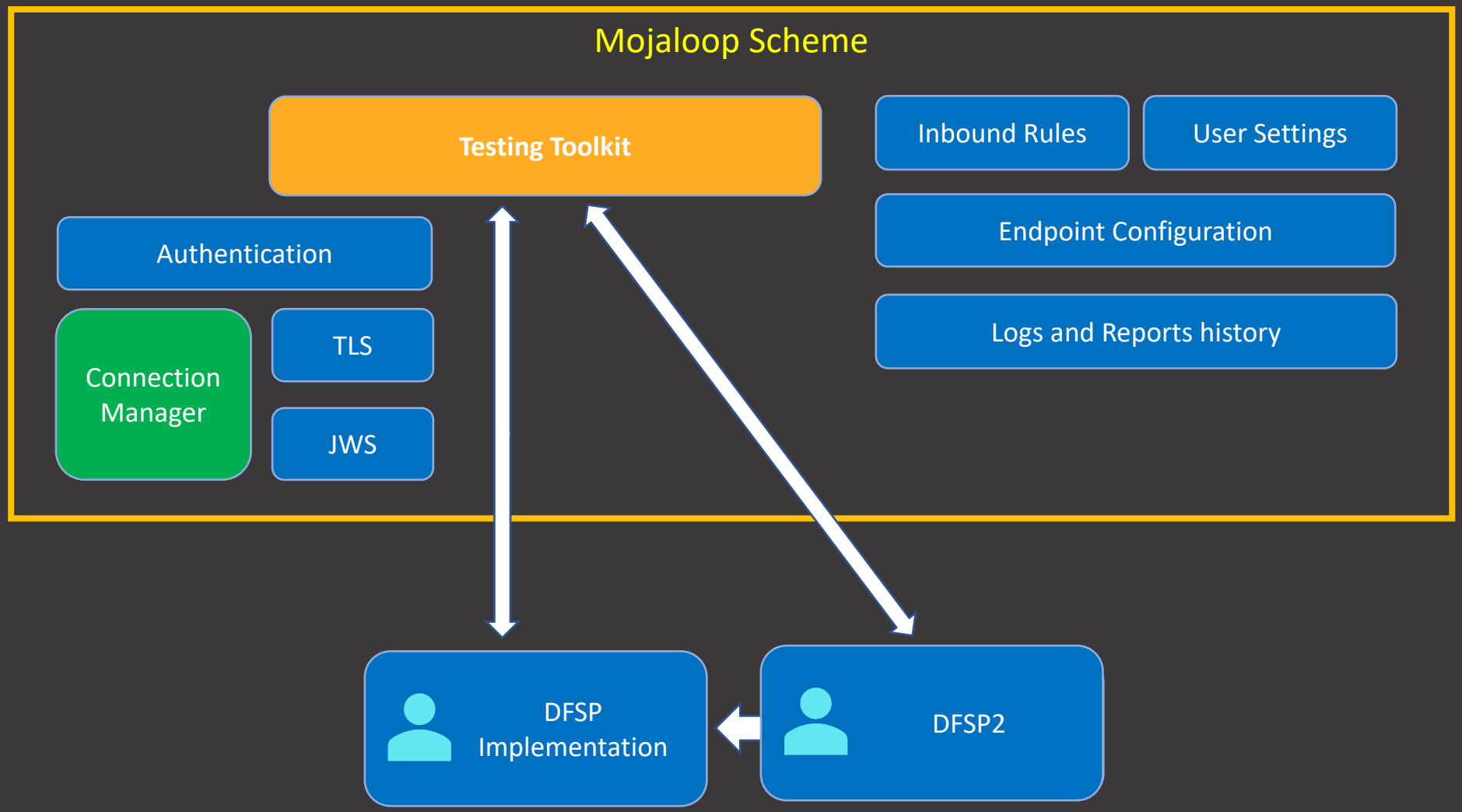
PI-10 Toolkit Features

1. User Interface for QA / Product / Business users
2. Version validation and negotiation, schema validation & additional validation
3. Dynamic callback and error callback generation based on rules
4. Initiation of use cases (outbound) - Assertions and report generation
5. Simultaneous support for multiple APIs
6. Separate test sets for Hub and DFSP implementations
7. Synchronous & Asynchronous APIs
8. Supports JWS and mTLS
9. Command line client (CLI) for scheduling & devops automation
10. Easily portable (Light weight and import and export configuration options)

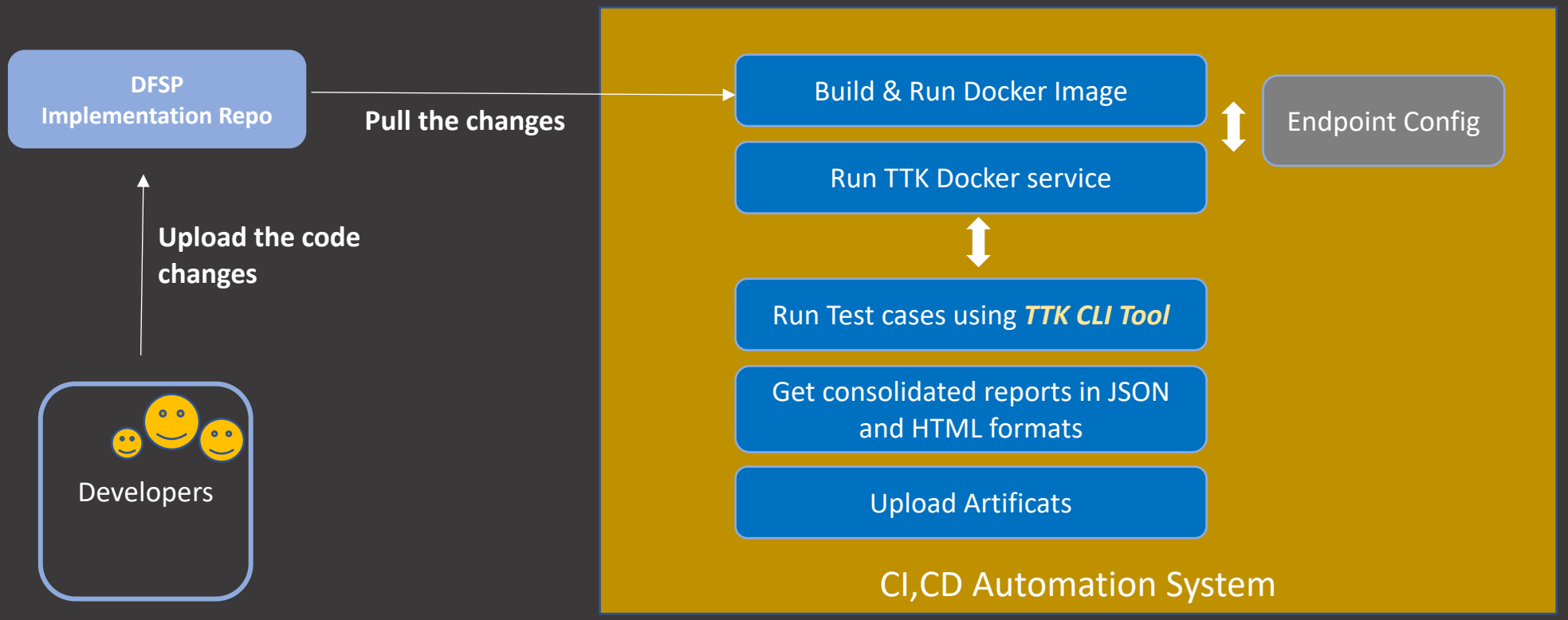
Testing toolkit: Enhancements in PI-10

1. Expand support for all the resources in the FSPIOP API
2. Command line client (CLI)
3. Import and Export capability of rules & settings
4. Improved Test case execution
    a. Separate test case sets for Hub & FSPs
    b. Test Reports and exporting options
    c. Curl (code) option for outbound messages
    d. Scripting capability for outbound request editor
5. Helm chart for testing toolkit
6. Verifying condition and fulfilment
7. Prioritize rules & delay simulation
8. Caching functionality in rules
9. QA: >98% test coverage
10. Initial version with hosting capability

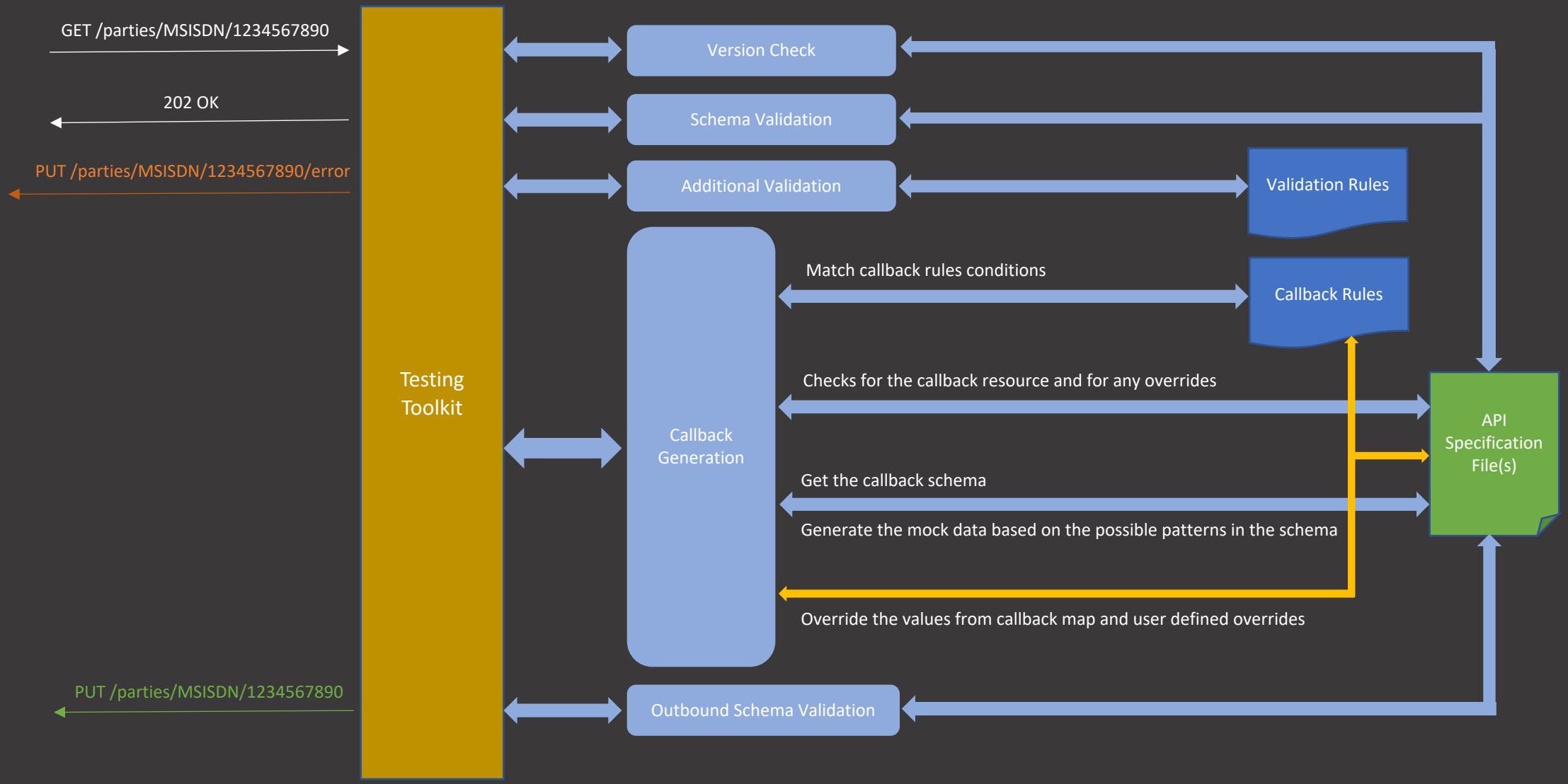
Adding Capability for Hosted Solution



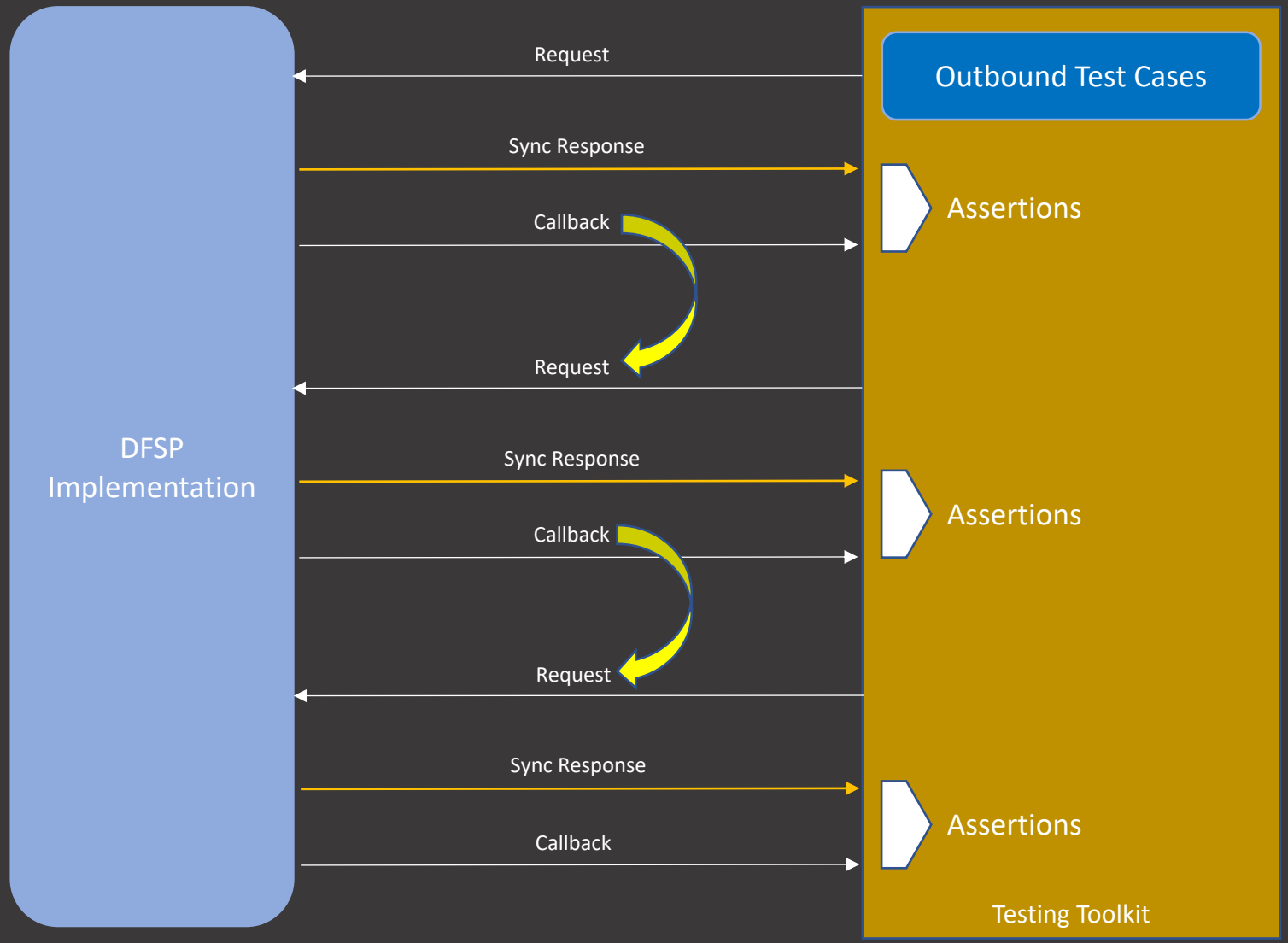
Integrating TTK into DFSP's CI, CD



How Testing Toolkit Works - Incoming requests



How Testing Toolkit Works - Test Case Initiation and Assertions

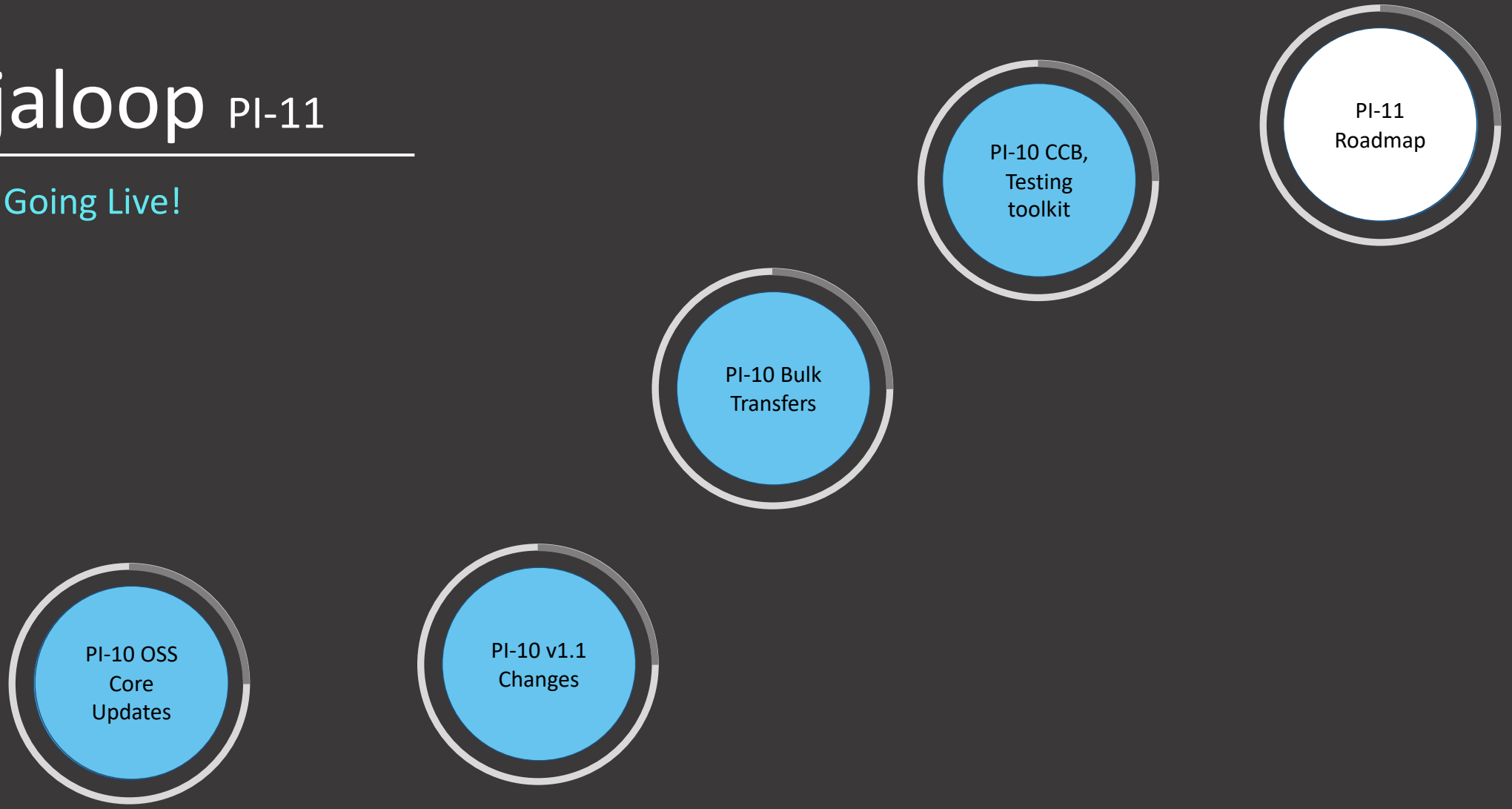


Testing toolkit: Roadmap for PI-11

1. Enable easy and ***secure hosting*** of testing toolkit and ensure ***scaling*** with separation of – 1) event handling, 2) Rules, 3) User Settings and 4) Logs & Reports storage
2. ***Postman collection conversion*** to TTK test cases
3. Improve coverage to test Hubs
4. Templates, Rules for all use cases from the Mojaloop Specification
5. Performance Tuning to withstand load (not for perf testing)
6. Event framework
7. Provide two interfaces – one for basic use and another for advanced usage

Mojaloop PI-11

Phase-4 Going Live!



Phase-4 PI-10: Objectives - 1

1. **Bulk Payments**: Deliver a standardized bulk transfer service for golden path which includes error & negative scenarios
2. **FSPIOP v1.1 changes**: Enhance the ML OSS implementation to support v1.1 changes including other maintenance items.
3. **Testing Toolkit**: Stand-alone version that can be used with features to export, import rules and settings is made available, with capability for hosting the toolkit.
4. **Settlement**: A flexible and configurable settlement service.

Phase-4 PI-10: Objectives - 2

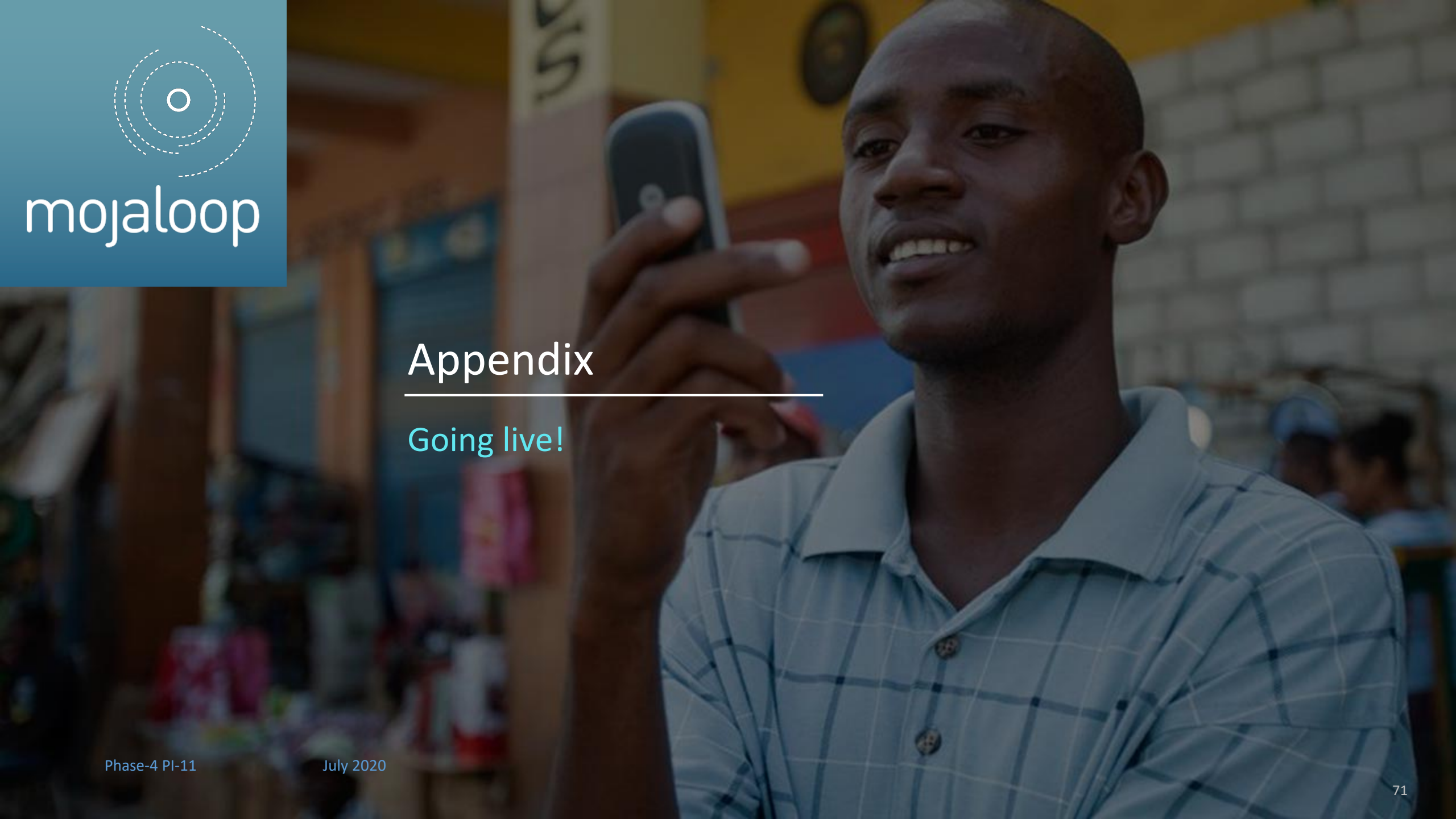
5. **Cross Currency/Network**: Finalize a design for the Mojaloop to non-Mojaloop flow including API changes and getting buy in from CNPs and the CCB
6. **Performance**: Deliver productionized Mojaloop deployment using Kafka best-practices; Evaluate Event-Sourcing PoC and Optimize SQL report and enhanced dashboards
7. **Fraud & Risk Mgmt**: Define, investigate and validate a backlog and MVP for a FRM system/service against the APRICOT modelling for existing/prospect operators; and identify partners to build/implement a FRM system/service
8. **PISP**: Ability to do an end to end PISP transfer (from demo PISP) and present a well-defined relationship between USER, PISP, and their DFSP
9. **OSS onboarding**: Updated documentation and end to end review of the onboarding process
10. **Versioning**: Propose Versioning standards for Mojaloop OSS releases focusing on implementation, deployments that include details on OSS support for LTS releases, branching & upgrade strategies.

Phase-4 PI-11: Roadmap – To be prioritized

1. Forensic Logging, Secure auditing
2. Performance: PoC learnings implementation
3. Ticketing system
4. Portals for Onboarding & Managing DFSPs, etc
5. Rules API
6. Split Payment Capability (SNAPP)
7. Standardize operations (Admin) API
8. Streamline Testing & QA
9. ISO 20022 adaptation

Phase-4: Roadmap – Ongoing

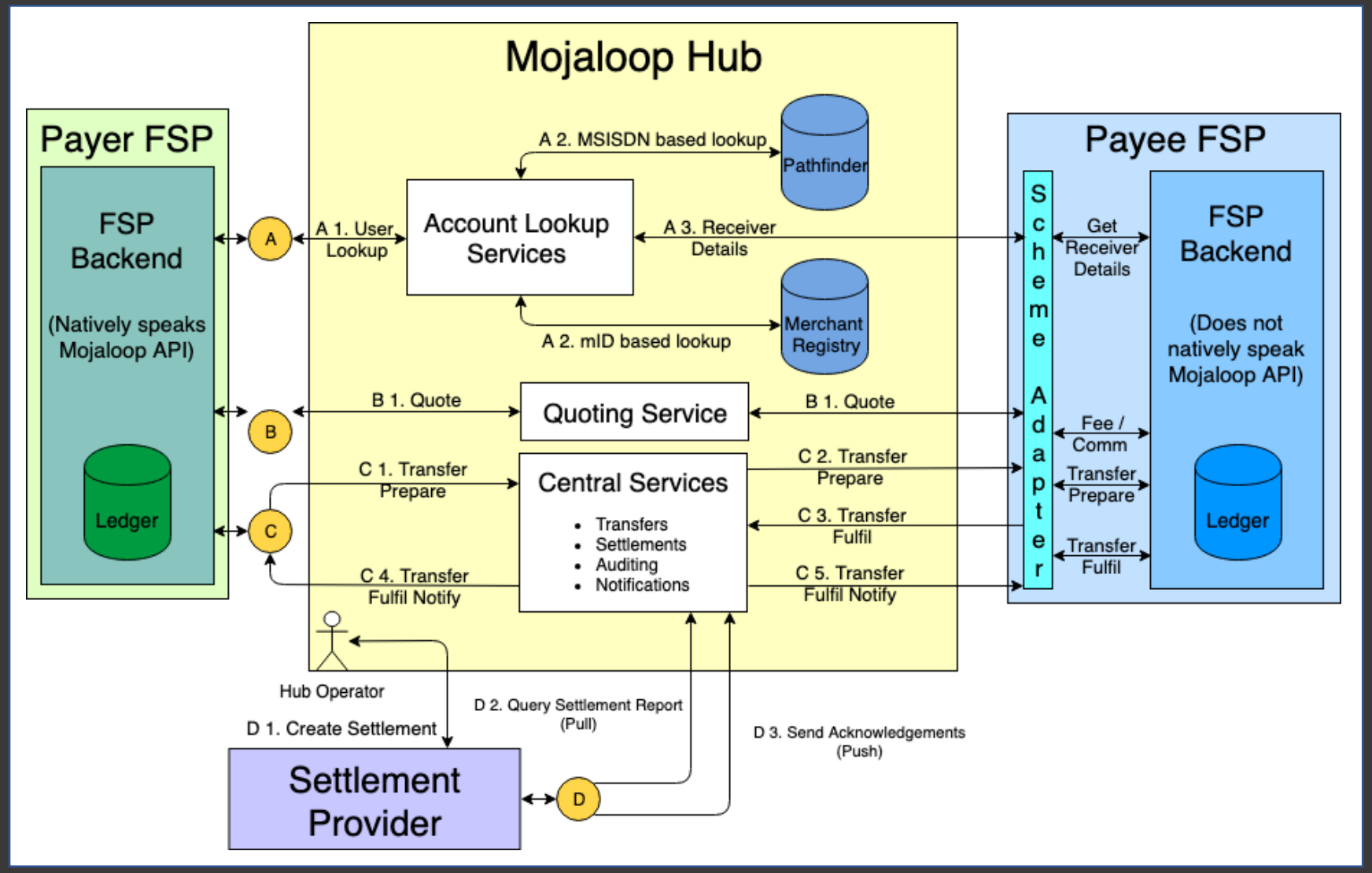
10. Code quality and Security
11. Cross Network / Currency (CNP/FXP)
12. Fraud Management
13. Leadership & Community Management (events, governance, communication, tools, etc.)
14. LPS Adapter Enhancements and future Use Cases (ATM, POS)
15. Payment Initiation Service Provider – Solution
16. Payment Hub (MIFOS)
17. Settlement v2
18. Testing tool-kit
19. Versioning standards, version maintenance
20. Web Payments (Coil)
21. GSMA Lab



Appendix

Going live!

Mojaloop Overview



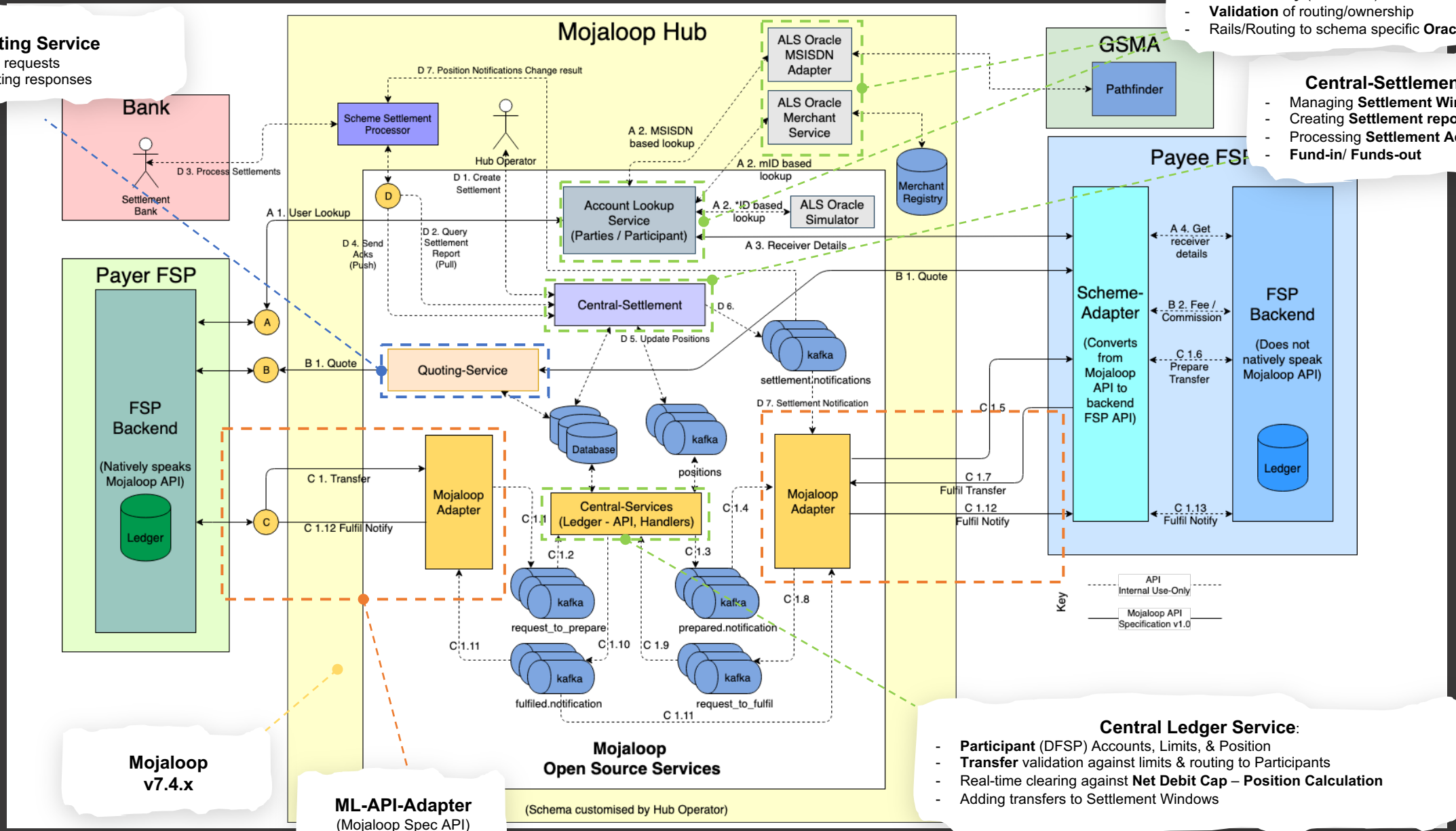
DEMO: http://mojaloop.io/docs/CentralServices/mojaloop-architecture-static-demo/index.html

High-level Architecture (PI7)

Account Lookup Service:
- Resolve **Participant** (DFSP) Routing
- Resolve **Party** (Customer) details via DFSP
- **Validation** of routing/ownership
- Rails/Routing to schema specific **Oracle registries**

Quoting Service
- Initiate Quote requests
- Resolve Quoting responses

Central-Settlement Service:
- Managing **Settlement Windows**
- Creating **Settlement reports**
- Processing **Settlement Acks**
- **Fund-in/ Funds-out**

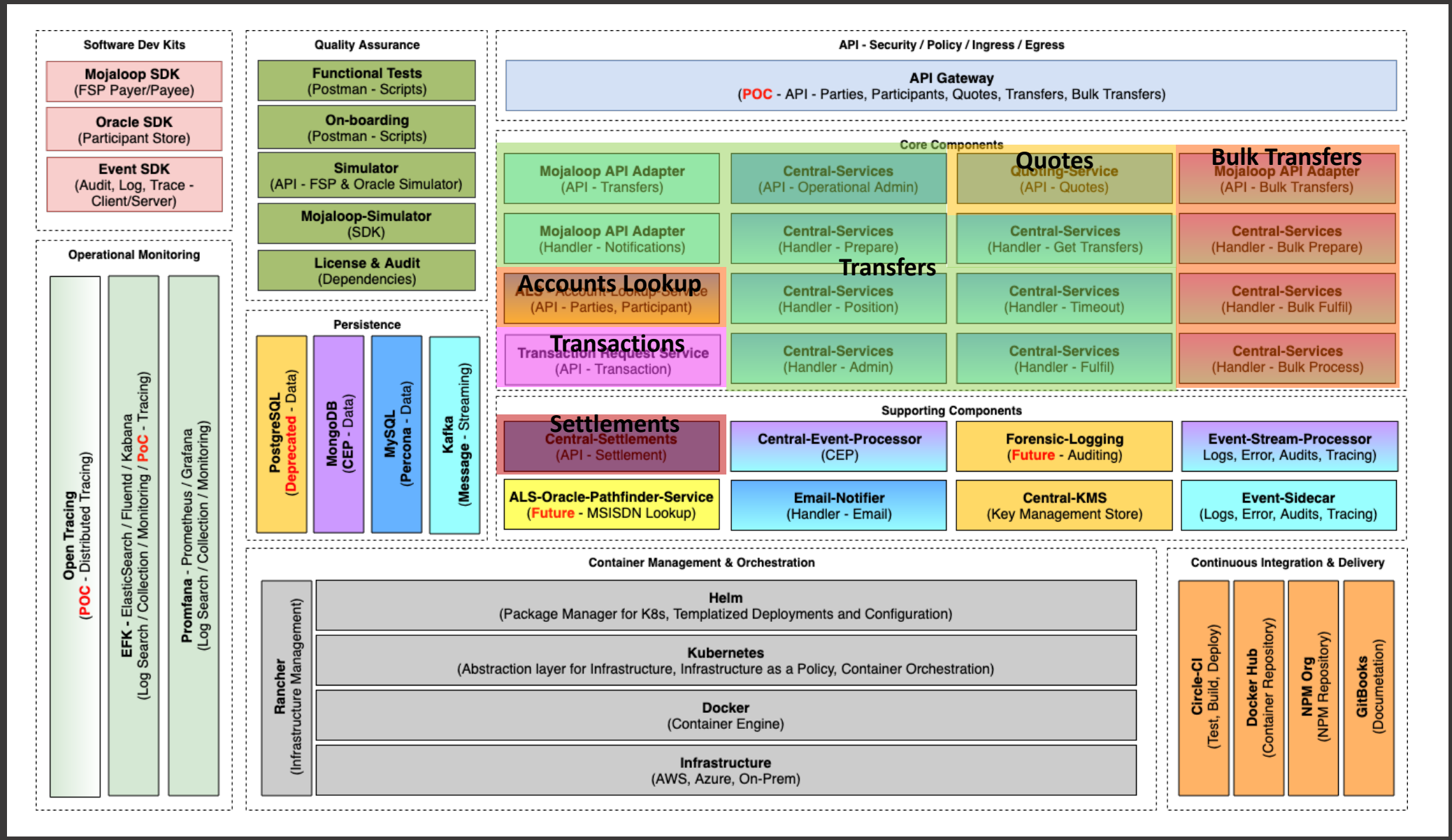


Mojaloop v7.4.x

ML-API-Adapter
(Mojaloop Spec API)

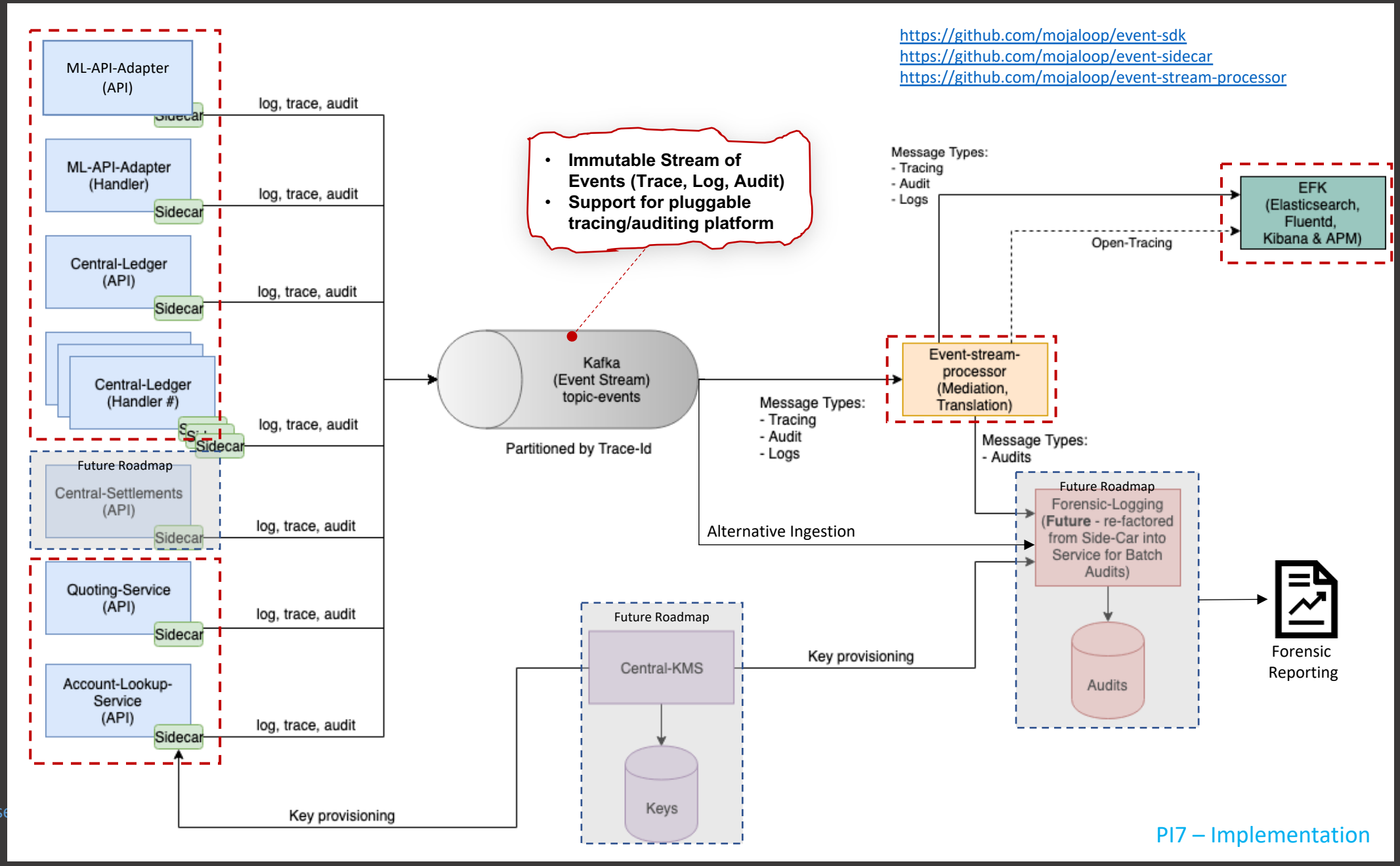
Central Ledger Service:
- **Participant** (DFSP) Accounts, Limits, & Position
- **Transfer** validation against limits & routing to Participants
- Real-time clearing against **Net Debit Cap** – **Position Calculation**
- Adding transfers to Settlement Windows

Component Overview (PI7)



* Color-shading of core & supporting components match dependent persistent stores.

Event Framework – Current Functional Overview



Run-time Platform – Kubernetes

What is Kubernetes?
Open-source system for automating deployment, scaling, and management of containerized applications.

kubernetes

Why Kubernetes?

Deploy your applications quickly and predictably
- Infrastructure as a Policy
- Abstraction of Infrastructure (Cloud, On-Prem)

Scale your applications on the fly
- Policy rule based scaling
- Elastic scaling (horizontally up/down)
- Limit hardware & resources via Policies

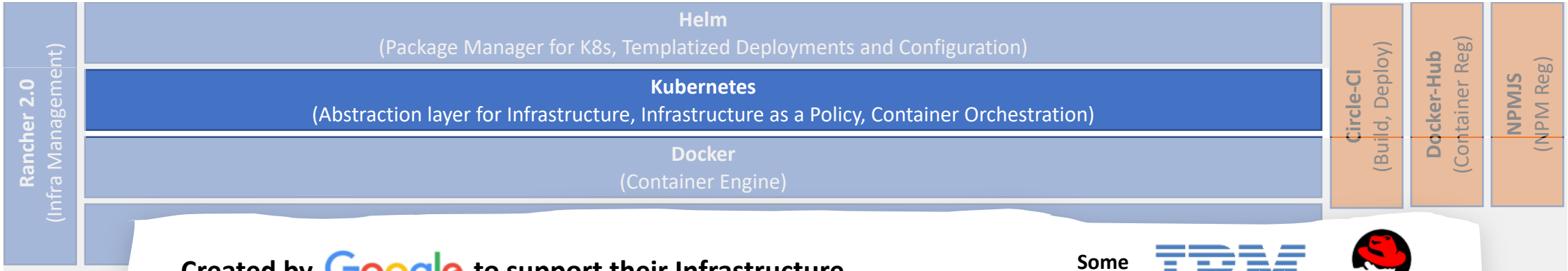
Roll out new features seamlessly
- Rolling updates

Discoverability
- Dynamic service resolution via DNS

Durability
- Self-healing
- Auto-[placement, restart, replication, scaling] based on Policies
- Load Balancing

Security
- Isolation through Containers, Network and Namespaces

Operations
- App config & secrets stored in distributed key-value store (etcd)
- Monitoring of containers



Created by Google to support their Infrastructure.

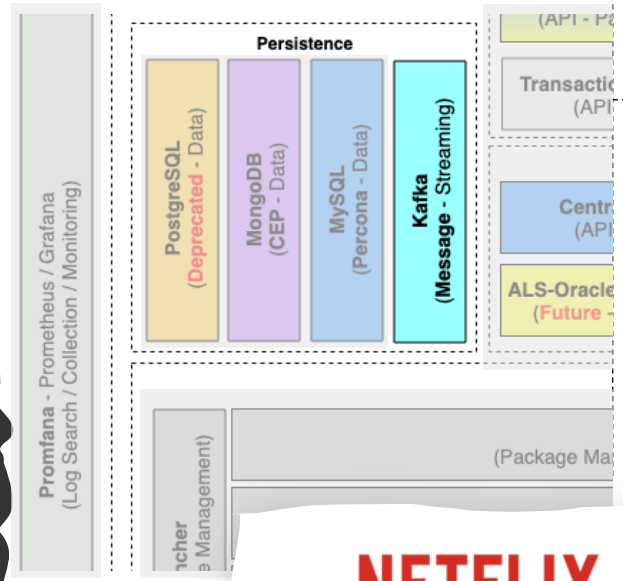
Some contributors: IBM redhat

Messaging Platform – Kafka

kafka

What is Kafka?

Apache Kafka is a distributed message streaming platform.



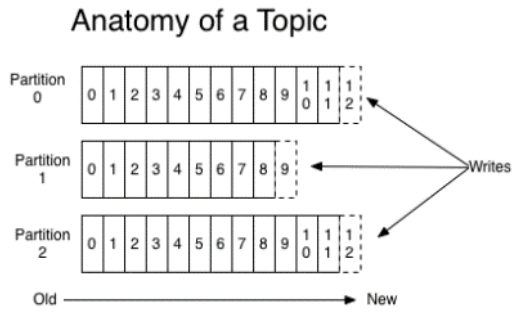
Ref: https://kafka.apache.org/

Why Kafka?

Publish and subscribe to streams of records, similar to a message queue or enterprise messaging system.

Store streams of records in a fault-tolerant durable way with history being saved for a desired period.

Process streams of records as they occur.



- For each topic, the Kafka cluster maintains a partitioned log.
- Each partition contains a sequence of records that is
  - Ordered; and
  - Immutable
- The records in the partitions are each assigned a sequential id number called the *offset* that uniquely identifies each record within the partition.

NETFLIX PayPal ORACLE IBM Rabobank

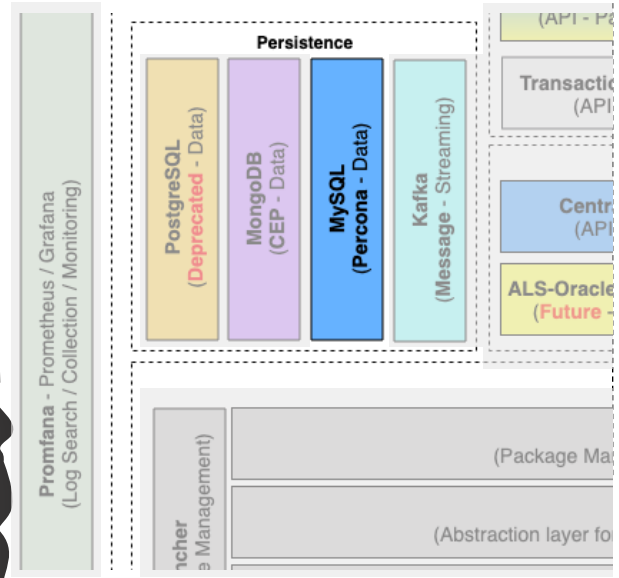
Used By

Storage Platform – Percona XtraDB Cluster

PERCONA

What is Percona XtraDB?

An open source, cost-effective, and robust MySQL clustering solution for businesses.



Ref: https://www.percona.com/software/mysql-database/percona-xtradb-cluster

Why Percona XtraDB?

- Cost-effective HA and scalability for MySQL with both Open Source and Enterprise support options
- Increased read/write scalability
- Zero data Loss
- Multi-master replication
- Works on-premises, cloud, hybrid, WAN, LAN, Kubernetes support (Helm)

Used By

Deployment Architecture – Rancher Overview

Ref: http://rancher.com

What is Rancher?

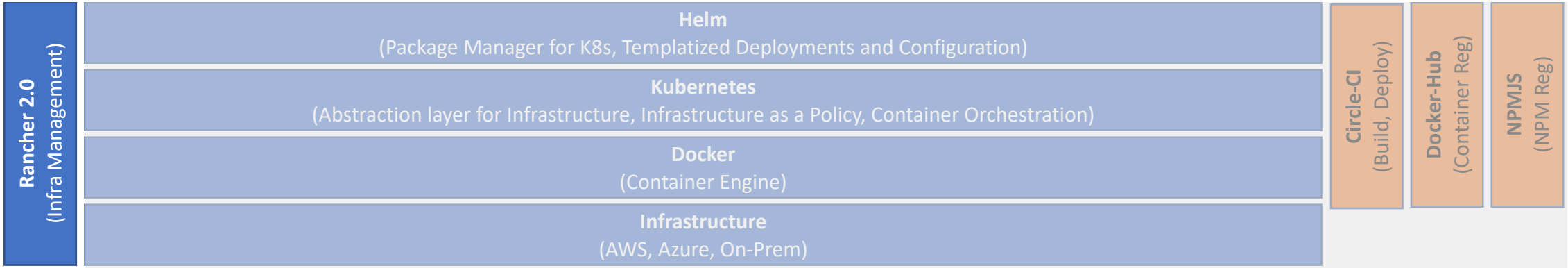
Rancher is enterprise management for Kubernetes.

Every distro. Every cluster. Every cloud.

Why Rancher?

- Kubernetes Management (v1.8 to *v1.9)
- Container Management
- *Access Management (RBAC)
- *Helm Repository Management
- Multi-environment Management (multi k8s clusters, On-prem, Azure, Google, AWS, etc)
- Multi-Provider provisioning (On-prem, Azure, Google, AWS, vSphere)
- Easily scale up/down Kubernetes clusters

** New in Rancher v2.x*



Deployment Architecture – Helm

Ref: http://helm.sh

What is Helm?

Open Source Package Manager for Kubernetes through the use of Charts.

Charts help you define, install and upgrade releases for Kubernetes deployment via templates and configuration.

Why Helm?

Manage Complexity

Charts describe even the most complex apps; provide repeatable application installation, and serve as a single point of authority.

Easy Updates

Take the pain out of updates with in-place upgrades and custom hooks.

Simple Sharing

Charts are easy to version, share, and host on public or private servers.

Rollbacks

Use `helm rollback` to roll back to an older version of a release with ease.

Microsoft Google codefresh bitnami ticketmaster codecentric

Deployment Architecture – CircleCI Overview

circleci

What is CircleCI? Cloud based Continuous Integration & Deployment Platform

Ref: http://circleci.com

VCS Integration
CircleCI integrates with GitHub, GitHub Enterprise, and Bitbucket. Every time you commit code, CircleCI creates a build.

Automated Testing
CircleCI automatically tests your build in a clean container or virtual machine.

Automated Deployment
Passing builds are deployed to various environments so your product goes to market faster.

Notifications
Your team is notified if a build fails so issues can be fixed quickly.

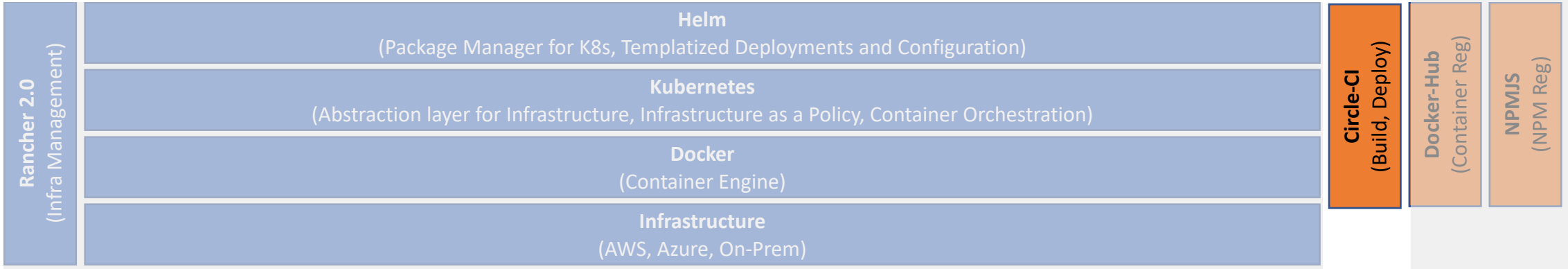
Why CircleCI?

Workflows for Job Orchestration
Orchestrate customizable job execution (such as build, test, deploy), giving complete control over your development process.

Language-Agnostic Support
Supports any language that builds on Linux or macOS, including C++, Javascript, .NET, PHP, Python, and Ruby.

First-Class Docker Support
Run any image from Docker's public/private registry or other common registries. Build Docker images, access Docker layer caching, Compose.

Powerful Caching
Speed up builds with expanded caching options, including images, source code, dependencies, and custom caches. Full control over cache save and restore points for optimal performance.



** Forrester names CircleCI a leader (https://www2.circleci.com/circleci-forrester-wave-leader-2017.html)*

Documentation – GitBooks Overview

GitBook

Ref: http://gitbook.com
Ref: https://en.wikipedia.org/wiki/Markdown

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?

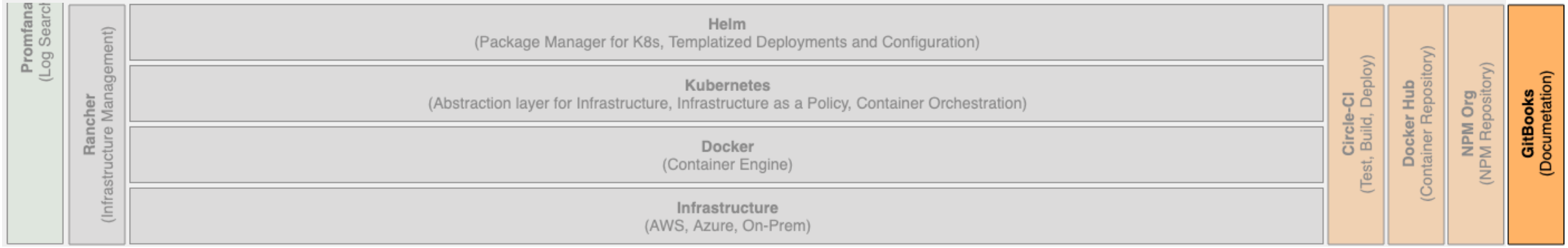
Markdown
Lightweight markup language with plain text formatting syntax supporting standard HTML, and CSS.

Imbed 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.



82

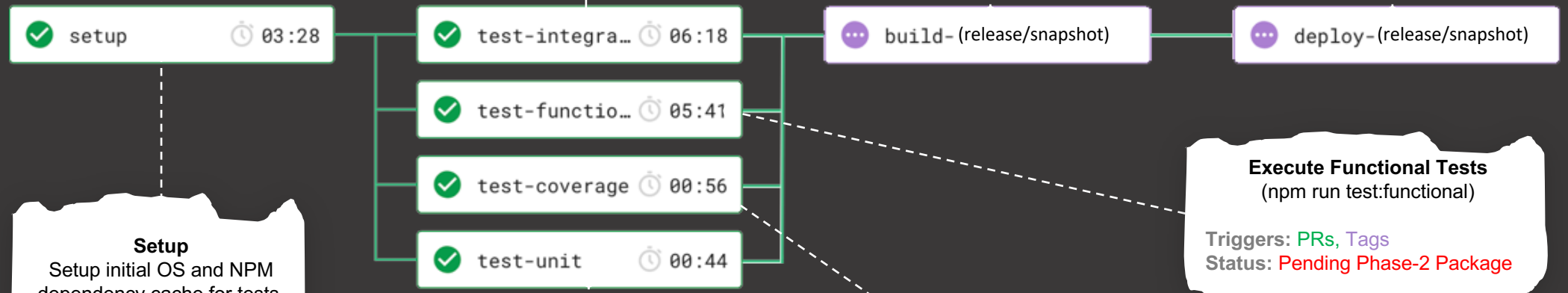
Deployment Architecture – CI/CD Pipeline

circleci

Triggers:

- [●] Pull-Requests (inc. from external Forks)
- [●] Publishing tagged Releases
- [●] Publishing tagged Snapshots

Ref: https://circleci.com/projects/gh/mojaloop



Execute Integration Tests
(npm run test:integration)

Triggers: PRs, Tags
Status: Working

Build & Publish Artifact
Docker or NPM build and publish to repository

Triggers: Tags
Status: Pending Phase-2 Package

Deploy Artifact
Execute either helm install or upgrade (if existing deployment) with latest Artifacts

Triggers: Tags
Status: Pending Phase-2 Package

Setup
Setup initial OS and NPM dependency cache for tests, builds, and deployments.

Triggers: PRs, Tags
Status: Working

Unit Tests
(npm run test:unit)

Triggers: PRs, Tags
Status: Working

Execute Functional Tests
(npm run test:functional)

Triggers: PRs, Tags
Status: Pending Phase-2 Package

Coverage Check
Executes Code Coverage checks. The results will be published to SonarQube if the branch is "Master".
(npm run test:coverage)

Triggers: PRs, Tags
Status: Working

Event Monitoring Framework – SDK Example

REFERENCE

New Span:
- Generate **Trace Context** (**traceId** & **spanId**)

Finish Span:
- Close **Span**
- Record **Trace**

```
// Creates a new parent span for given service
// this sets new traceId and new spanId.
let parentSpan = Tracer.createSpan( service: 'parent service')
```

```
// Finish the span. This also sends the trace context to the tracing platform. All further operations are forbidden after the span is finished.
await parentSpan.finish(event)
```

Recorded events
- **Logs** (info, warn, debug, verbose, perf, error)
- **Audit**

```
// Logs message with logging level info from the parent span
await parentSpan.info(event)
await parentSpan.warning( message: 'event')
await parentSpan.error('event')
await parentSpan.debug('message')
await parentSpan.verbose( message: 'message')
await parentSpan.performance( message: 'message')
await parentSpan.audit( message: 'message')
// Logs message with logging level debug from the parent span
await parentSpan.debug('this is debug log')
```

Add Tags
- Add arbitrary **tags** for **metadata** as part of **Trace Context**

```
// Set tags to the span
IIChildSpan.setTags({ one: 'two' })
```

Child Span
- **Create Child Span** from a parent span
- Generate **spanId**, and set **parentId**

```
// Creates child span from the parent span with new service name.
// The traceId remains the same. The spanId is new and the parentSpanId is the spanId of the parent.
let IIChildSpan = parentSpan.getChild( service: 'child II service')
```

Trace Context
- **Inject Trace Context** into a Message
- **Extract Trace Context** from a Message
- **Create Child Span** from **Trace Context**

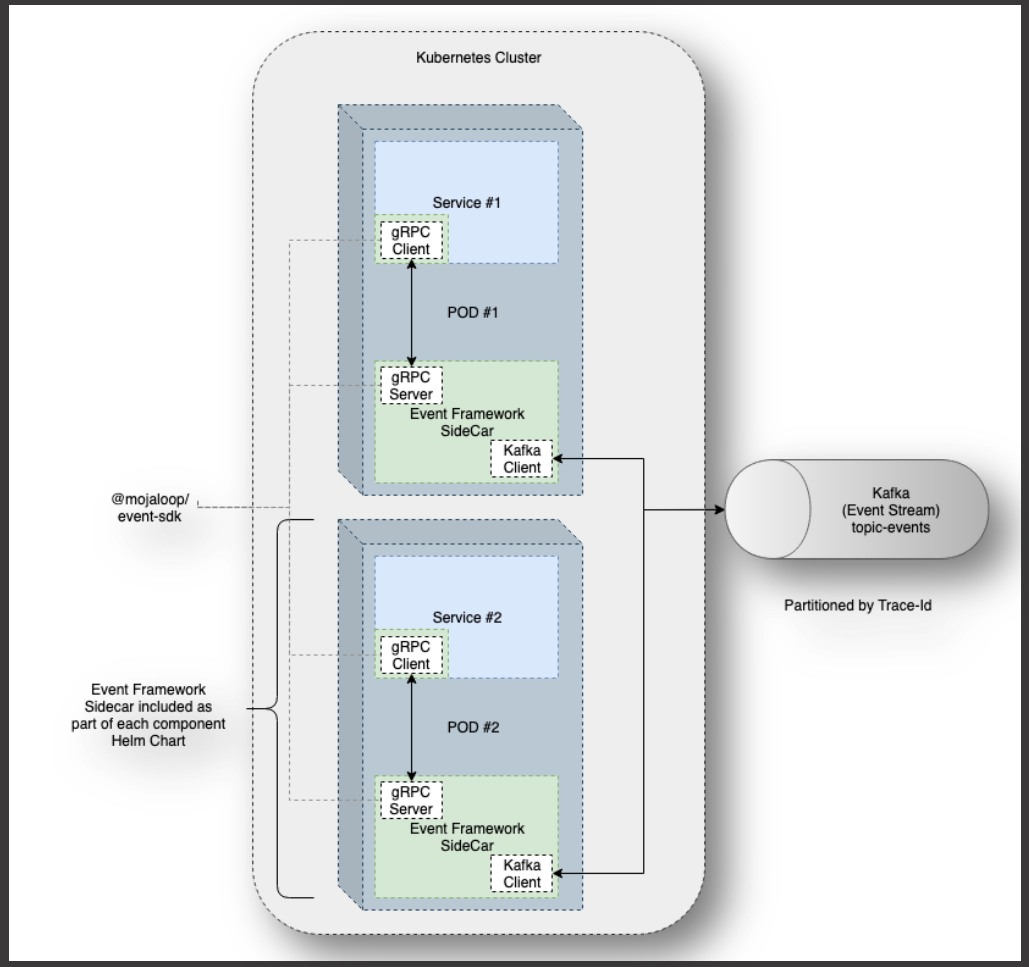
```
// Injects trace context to a message carrier. When the trace is carried across few services, the trace context can be injects in the carrier that transports the data.
let messageWithContext = await IIChildSpan.injectContextToMessage(event)
// await sleep(2000)

// Extracts trace context from message carrier. When the message is received from different service, the trace context is extracted by that method.
let contextFromMessage = Tracer.extractContextFromMessage(messageWithContext)

// Creates child span from extracted trace context.
// let IIIChild = Tracer.createChildSpanFromContext('child III service', contextFromMessage, { defaultRecorder: new DefaultLoggerRecorder() })
let IIIChild = Tracer.createChildSpanFromContext( service: 'child III service', contextFromMessage)
```

July 2

Event Framework – Deployment Arch & Roadmap



Deployment Architecture

Future Roadmap

1. Design considerations:
   a. Audit requirements / functionality
   b. Crypto signatures for Audit logs (single & batch)
   c. Opentracing / Zipkin dashboards if EFK is not adequate
   d. Sidecar inter-lock
2. Implementation:
   a. Audits sig & processing
   b. Official releases
   c. Mojaloop Helm Chart integration

GitHub PoC Repositories*

1. https://github.com/mojaloop/event-sdk
2. https://github.com/mojaloop/event-sidecar
3. https://github.com/mojaloop/event-stream-processor
4. https://github.com/mojaloop/apm-agent-nodejs
5. https://github.com/mojaloop/apm-agent-nodejs-opentracing
6. https://github.com/mojaloop/opentracing-javascript

** Note:*

1. *Event PoC code currently in feature branches*
2. *Snapshot releases currently available*

Event Monitoring Framework

1. Overview
2. Functionality
3. SDK Examples
4. Roadmap
5. Demo

Event Monitoring Framework – Overview

What

1. Unified framework to capture all Mojaloop events and ingest them appropriately
2. Event Types:
   a. Logs
   b. Errors
   c. Audits
   d. Traces

Why

1. Operational monitoring of requests end-to-end
   a. End-to-end request visualization
   b. Enabler for alerts
   c. Issue resolution
2. Enabler for auditing and fraud management

How

1. Standardized framework for capturing events (types, actions, metadata, etc)
2. Every request is given a trace-id at the boundary.
3. Standard common (Event-SDK) library that will publish events to a sidecar component utilising a light-weight highly performant protocol (*e.g. gRPC*)
4. Sidecar module will publish to a Kafka messaging stream for all events utilising Event-SDK
5. Each Mojaloop component will have its own tightly coupled Sidecar
6. Leverage on open source standards and solutions where possible (*e.g. APM, OpenTracing, Zipkin, etc*)

Event Framework – Functionality

REFERENCE

Event SDK Features

1. Spans
   1. Create new span
   2. Create child spans
2. Event Types:
   1. Logs (info, debug, error, performance, warning, verbose)
   2. Audit (default, start, finish, ingress, egress)
   3. Traces (Spans)
3. Client/Server:
   1. gRPC client
   2. gRPC server
4. Recorders:
   1. Async gRPC (event-sidecar)
   2. Sync gRPC (event-sidecar)
   3. Logger (Own Logger Instance)
5. Supports arbitrary tags
   1. Add additional information as part of the trace context (e.g. transferId)
6. Context
   1. Inject Context into Message
   2. Extract Context from Message
   3. Create Child Spans from Context
   4. Support for Messaging (e.g. Kafka) & HTTP Transports (WC3 Tracing *standard)
   5. Support for ”traceState” header from WC3 Tracing *Standard

* https://www.w3.org/TR/trace-context-1/

Event Sidecar

1. Publishes Messages to Kafka Event Topic

Event Stream Processor

1. Record **logs** & **audits** to **EFK**
2. Record **traces** to APM server

Helm

1. Helm Charts with Event-sidecar support:
   a. ML-API-Adapter
   b. Central-Ledger
   c. Quoting-Service
   d. SDK-Schema-Adapter
   e. Legacy Simulator

Improvements in PI8

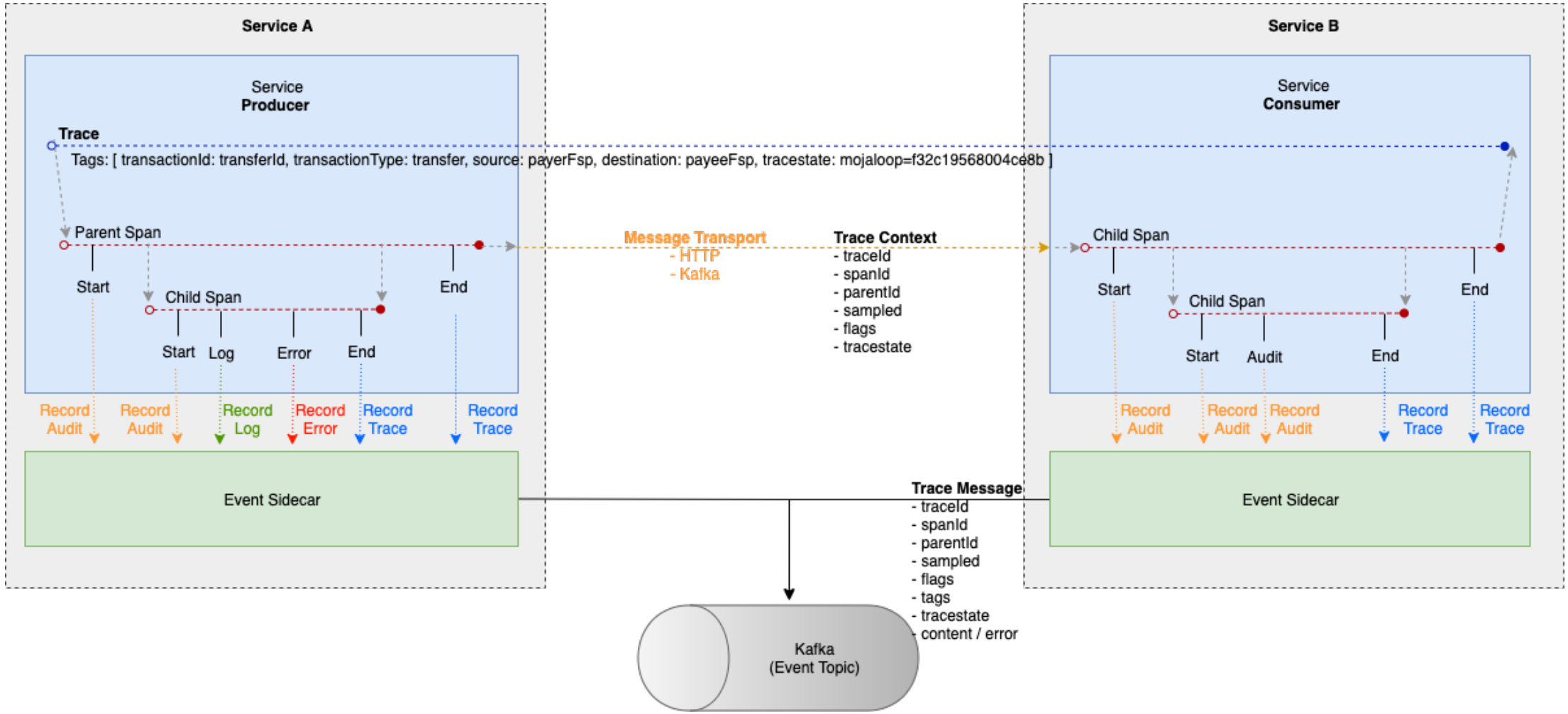
1. Enhanced WC3 Tracing Standard Support
   a. Tracestate header
2. General
   a. Sync - Async configuration for Event levels (trace, audit, log)
   b. Precise control over logging (enable/disable, local/remote & metadata)
   c. Improved stability

Future Roadmap

1. Roll-out
   a. Account-Lookup-Service
2. Performance testing
   a. What is the impact of enabling Tracing
   b. What is the optimum/acceptable level of tracing/logging

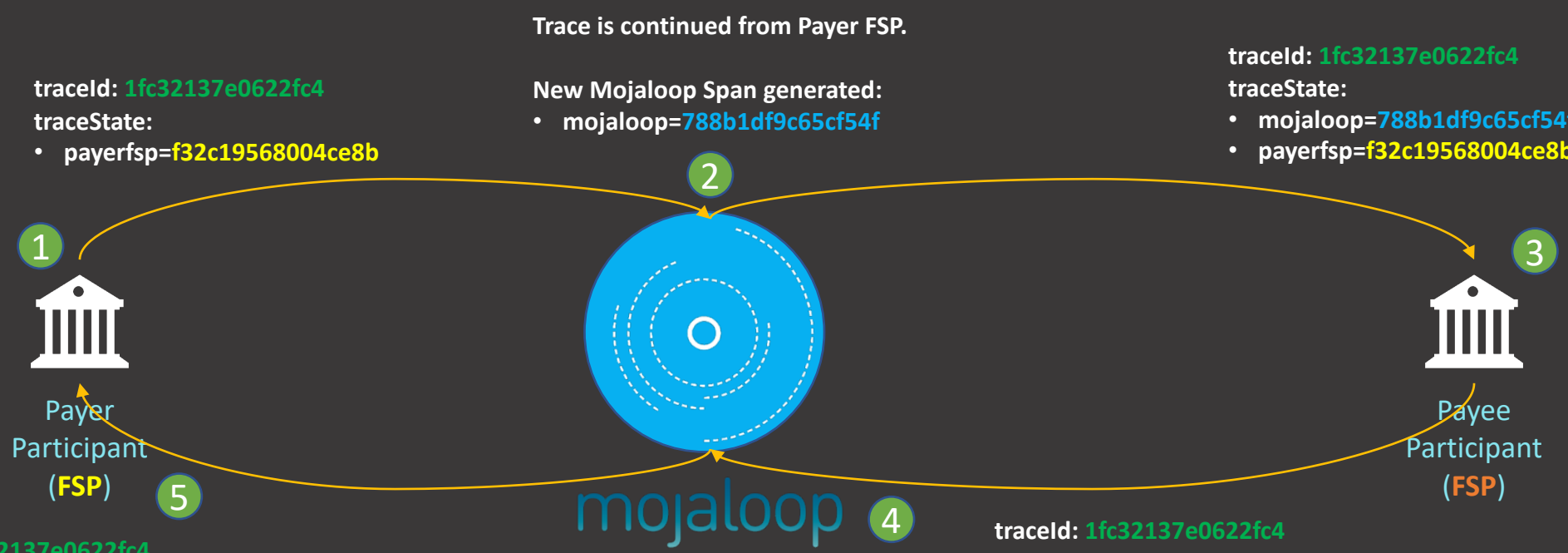
Event Monitoring Framework – Trace Architecture

REFERENCE



Event Flow of Events

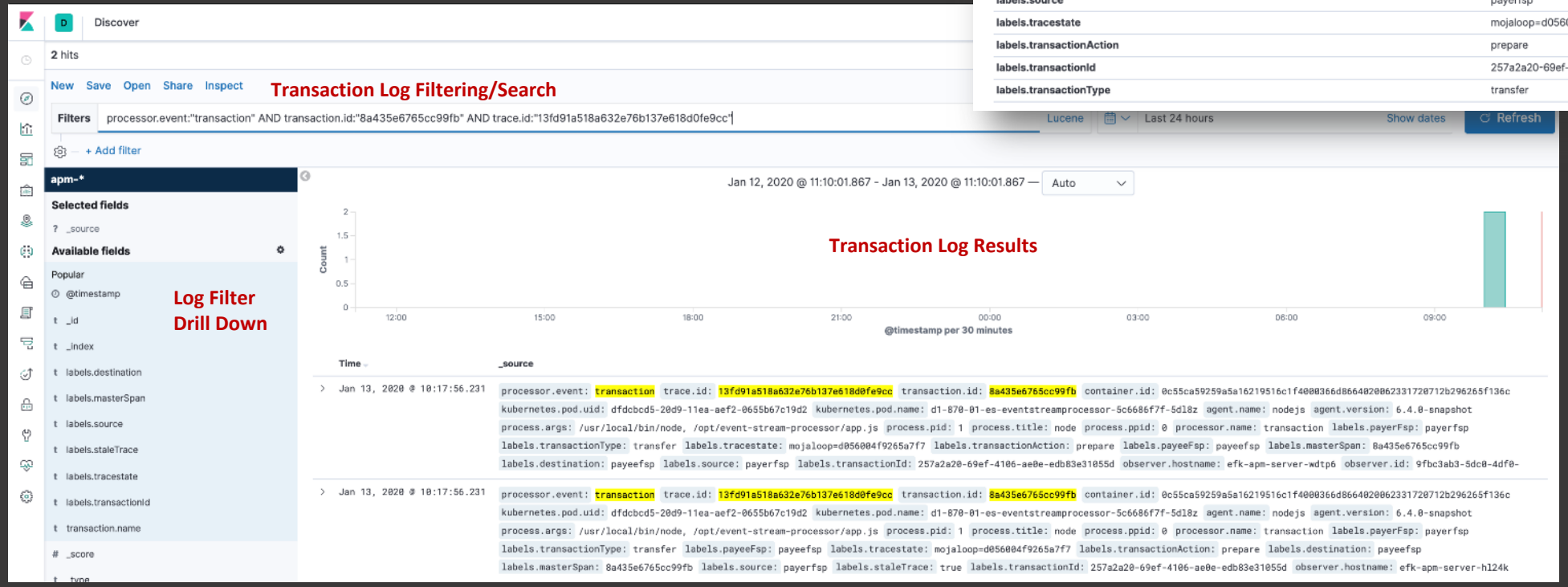
Tracing Enhancement – Federated Tracing

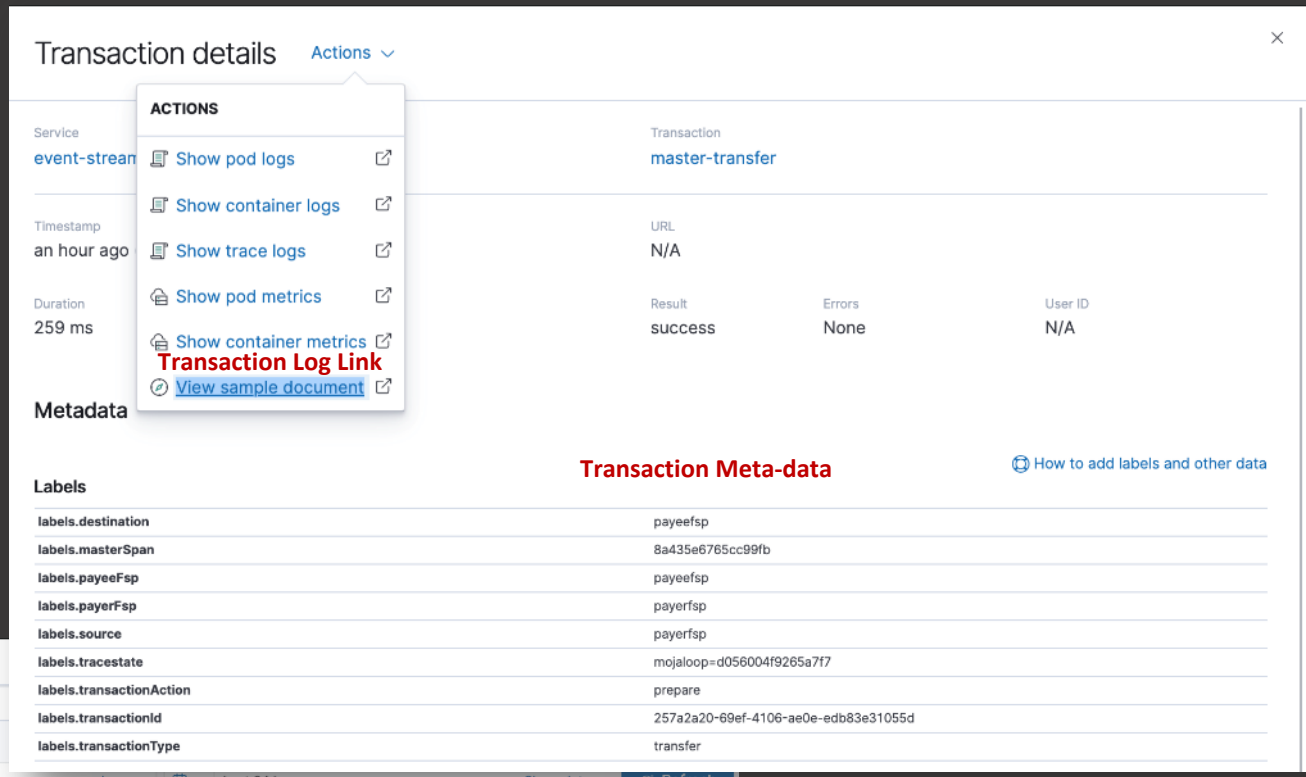


REF: https://www.w3.org/TR/trace-context-1/#relationship-between-the-headers

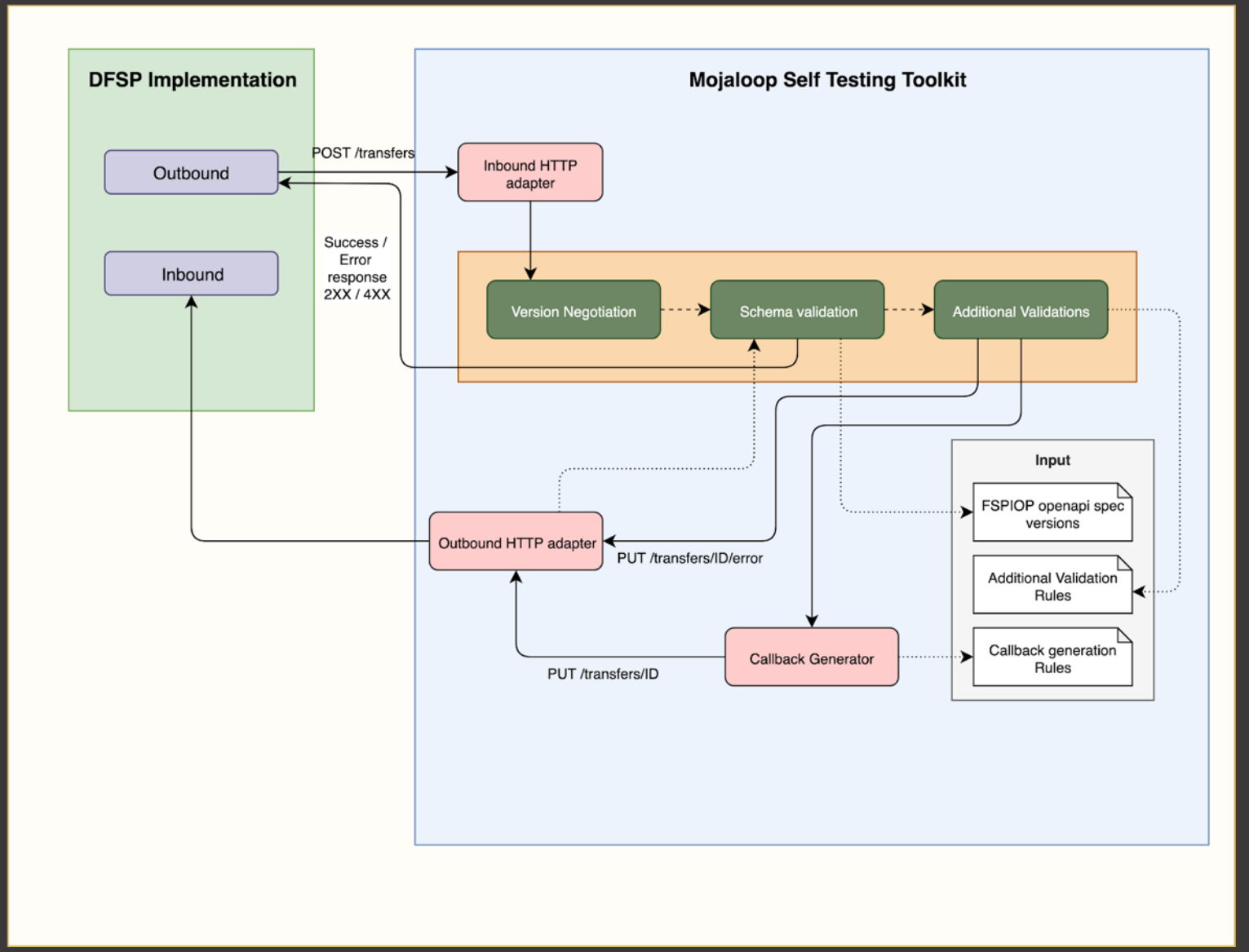
Event Monitoring Framework – Dashboard Sample (Continued)

Direct link from Tracing Dashboard to Processing & Tracing logs





Architecture Diagram – Part 1



Architecture Diagram – Part 2