



mojaloop

# Mojaloop End-of-Phase-3

---

Supporting Adoption & Deployment

# Mojaloop End of Phase-3

## Supporting Adoption & Deployment



# Mojaloop PIs Overview

Timeline	Summary
Phase-1 (2016 - 17)c	<p><b>Level One Project</b></p> <ul style="list-style-type: none"><li>• Reference Implementation</li><li>• 6 Program Increments (PIs)</li></ul>
Phase-2 (2018)	<p><b>Road To Productionization</b></p> <ul style="list-style-type: none"><li>• PI – 1 (Feb - April)</li><li>• PI - 2 (April - June)</li><li>• PI - 3 (June - August)</li><li>• PI – 3.5 (September)</li><li>• PI – 4 (November-December 2018): Performance, Settlements, CEP, QA Framework, Operational Monitoring, Managed backlog for Phase-3</li></ul>
Phase-3 (2019 Jan - Sep)	<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>• <b>PI-7 Event &amp; Error Handling framework, Packaging, OSS Settlements, Performance testing capabilities, QA</b></li></ul>

# PI7 Topics - Features

1. Event Handling Framework
  - a. Framework
  - b. Instrumentation
2. Error Handling Framework
3. *OSS Settlements*

# PI7 Topics - Enhancements

1. QA Framework
2. Packaging, Releases
3. Idempotency in Mojaloop
4. Operations monitoring support

# PI7 Topics - Updates

1. Performance Testing Capabilities
2. Bug Fixes
3. On-boarding improvements
4. Deployment enhancements

# PI-7 Mojaloop OSS Community

## Collaboration

- a. Documentation, On-boarding (CrossLake, ModusBox)
- b. Quality Assurance (CrossLake, Mowali, ModusBox)
- c. Account Lookup Service (Mowali, TIPS, ModusBox)
- d. Scrum-of-scrums (Wider community)
- e. Design Authority
- f. mojaloop-simulator (Mowali, ModusBox, Crosslake)
- g. Oracle shared library (TIPS, OSS)

# Switch Functionality – Mojaloop End-points (PI6→PI7)

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

# End-of-PI7 Switch Functionality – Mojaloop End-points

## Mojaloop v1.0 – API Specification

### Transfers

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

1. Event Framework
2. Error Framework
3. Instrumentation
4. Standardizing

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

# End-of-PI7 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

1. Event Frameworks
2. Error Frameworks
3. Standardizing

### Settlements v1.0 [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

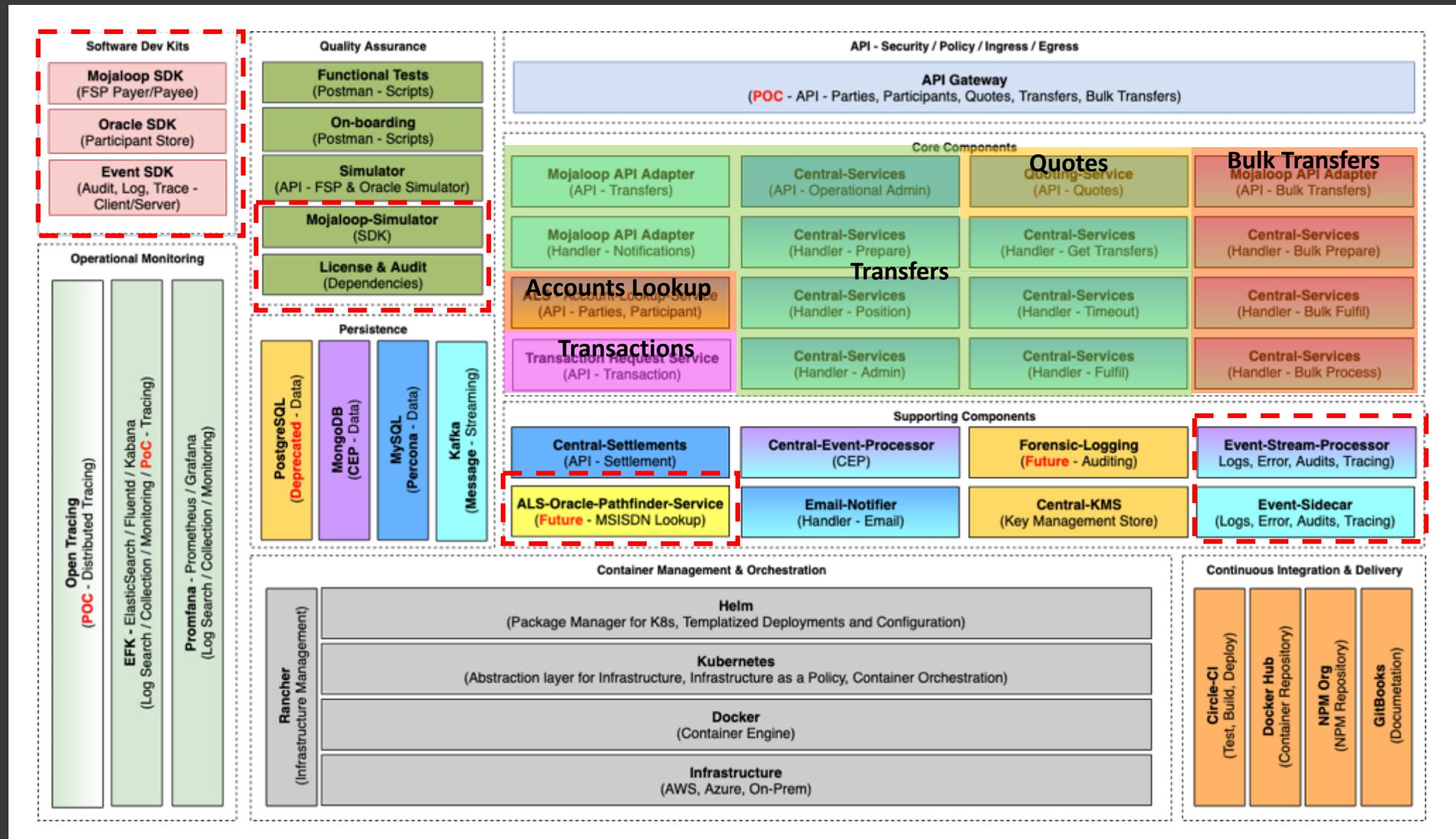
### Oracles

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

#### Key

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

# PI-7: Component Architecture



# PI-7: Objectives

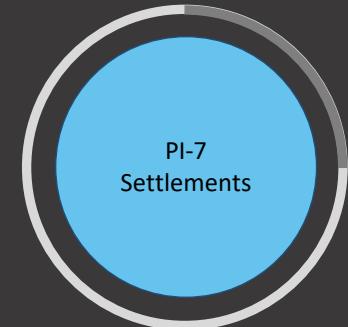
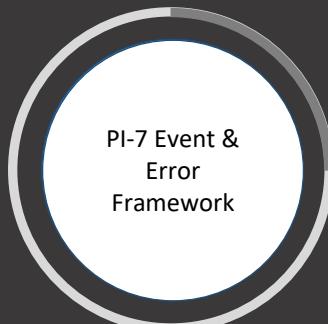
1. Have a production ready **release**, with clean licenses, updated health checks, documentation, appropriate code coverage which meets/exceeds requirements – **ModusBox/Crosslake**
2. **Error & Event Handling** Framework that supports end-to-end Tracing, Visualization and Instrumentation
3. **QA Framework** that supports PI-7 Features (FX, Charging, etc.) and test results are published
4. Ability to setup and **run performance tests** on an ML OSS environment
5. **Settlements** that support Net & Gross Settlement by Currency + Ledger Type AND NDC Check by Currency + Ledger Type

# PI-7: Additional accomplishments

1. Idempotency [Duplicate check enhancements]
2. Adopting Mowali Simulator
3. Community support
4. QA, Bug Fixes
5. Enhancing numbers processing in the Switch
6. Releases
7. AWS environment maintenance [Dev, QA]
8. Onboarding improvements

# Mojaloop End of Phase-3

## Supporting Adoption & Deployment



# PI-7 Event Framework - Overview

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

# Event 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 – Current PI7 Functionality

## 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 (Mojaloop Common Logger library)
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 standard)

## Event Sidecar

1. Publishes Messages to Kafka Event Topic

## Event Stream Processor

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

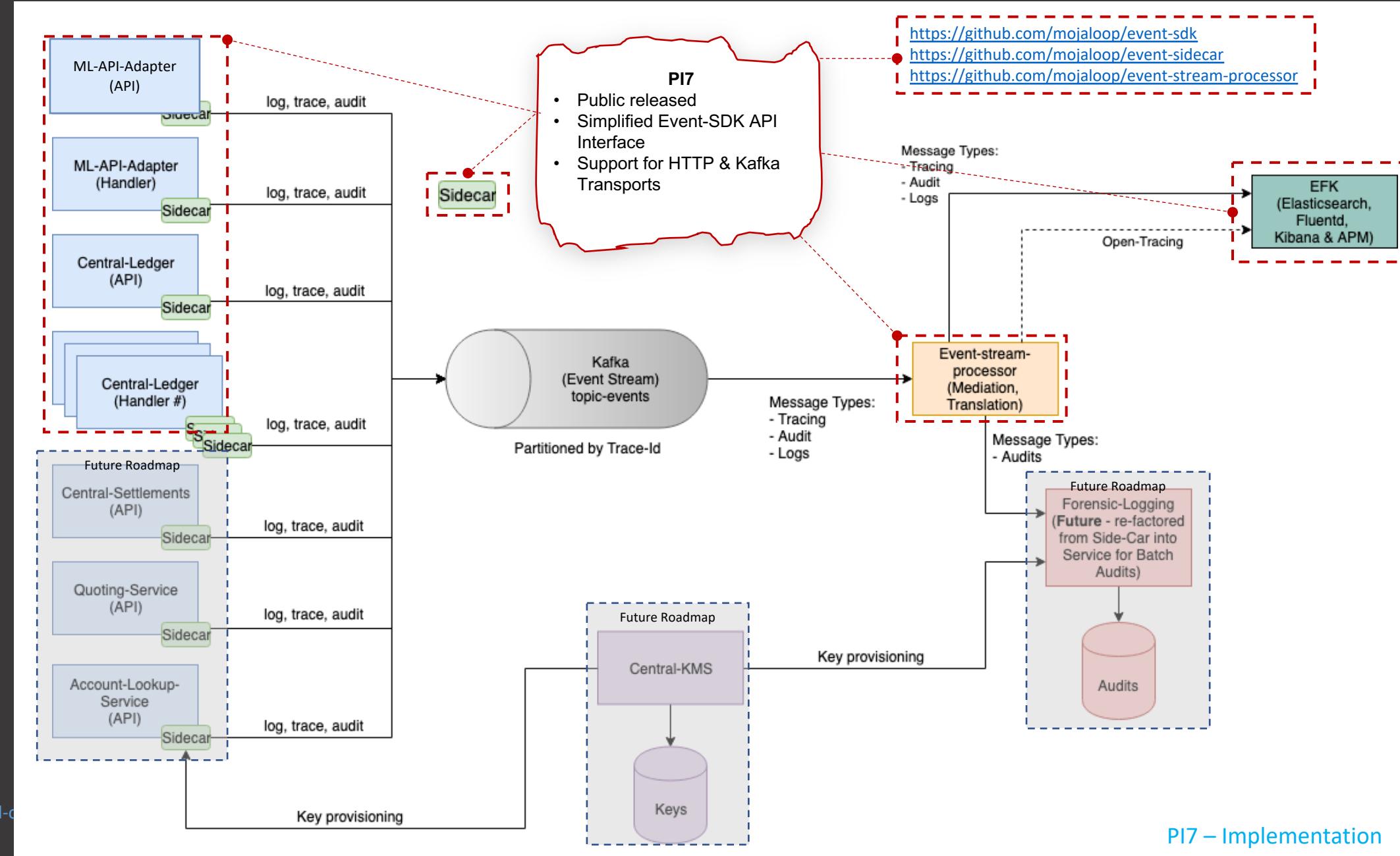
## Helm

1. EFK updated to latest version to be compatible with APM
2. APM added to EFK deployment
3. Added [Event-sidecar](#) to Central-Ledger and [ML-API-Adapter](#) charts

## Improvements in PI7

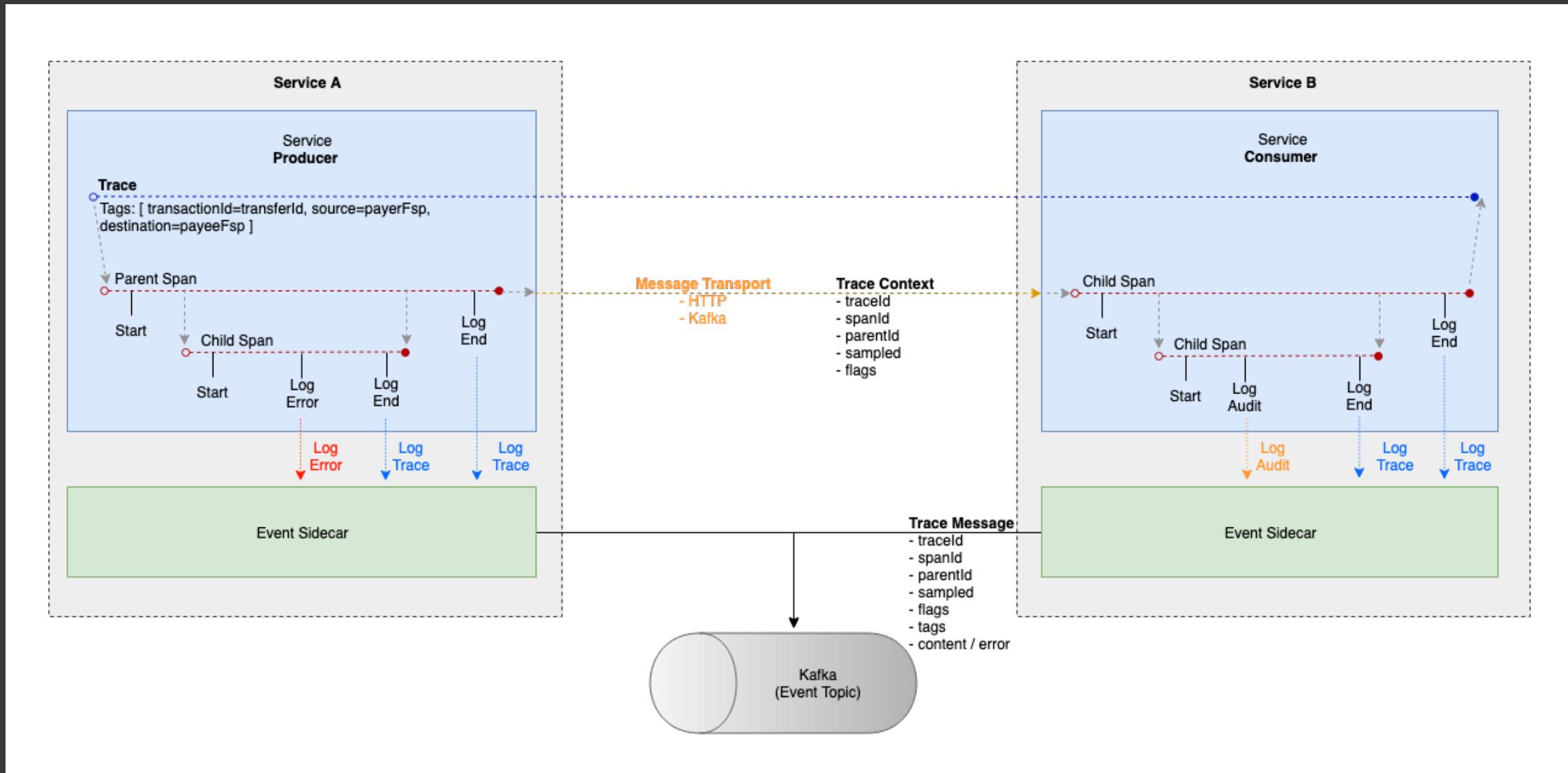
1. General
  - a. Upgraded to support latest version of APM
  - b. Simplified Event-SDK API for easier instrumentation
  - c. Created custom Mojaloop index
  - d. Aligned elasticsearch & APM metadata fields to improve user experience
2. Mojaloop EFK Schema Template
  - a. Defined a custom Mojaloop Schema Template for all Mojaloop index's to support dynamic message payloads
3. HapiJS Plugin
  - a. Manages Spans, Trace Context, etc
  - b. Handling of HTTP Transport (wc3 standard) headers

# Event Framework – Current PI7 Functionality





# Event Framework – Trace Architecture





# Event Framework – SDK Example

## 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 injected 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)
```

# Event Framework – Roadmap

## Event-SDK Features:

1. Testing \*
  - a. Automated Testing
  - b. Validate end-to-end synchronous requests.

## Kibana

1. Custom dashboards to support Performance reporting
2. Custom dashboards to report operational health

## Instrumentation

1. ALS
2. Quotes
3. Settlements

## Mojaloop SDK

1. Native support for Tracing Instrumentation

## Side-car

1. Testing
  - a. Automated Testing
  - b. Validate end-to-end synchronous requests.
2. Security
  - a. Crypto Signatures of Audits
  - b. Side-car Interlock to validate host container is intended version and ensure that no code changes have been injected

## Stream Processor

1. Improve capture of stack-trace integration into APM
2. Potential support for Batch Audits & Crypto Signatures

## Helm

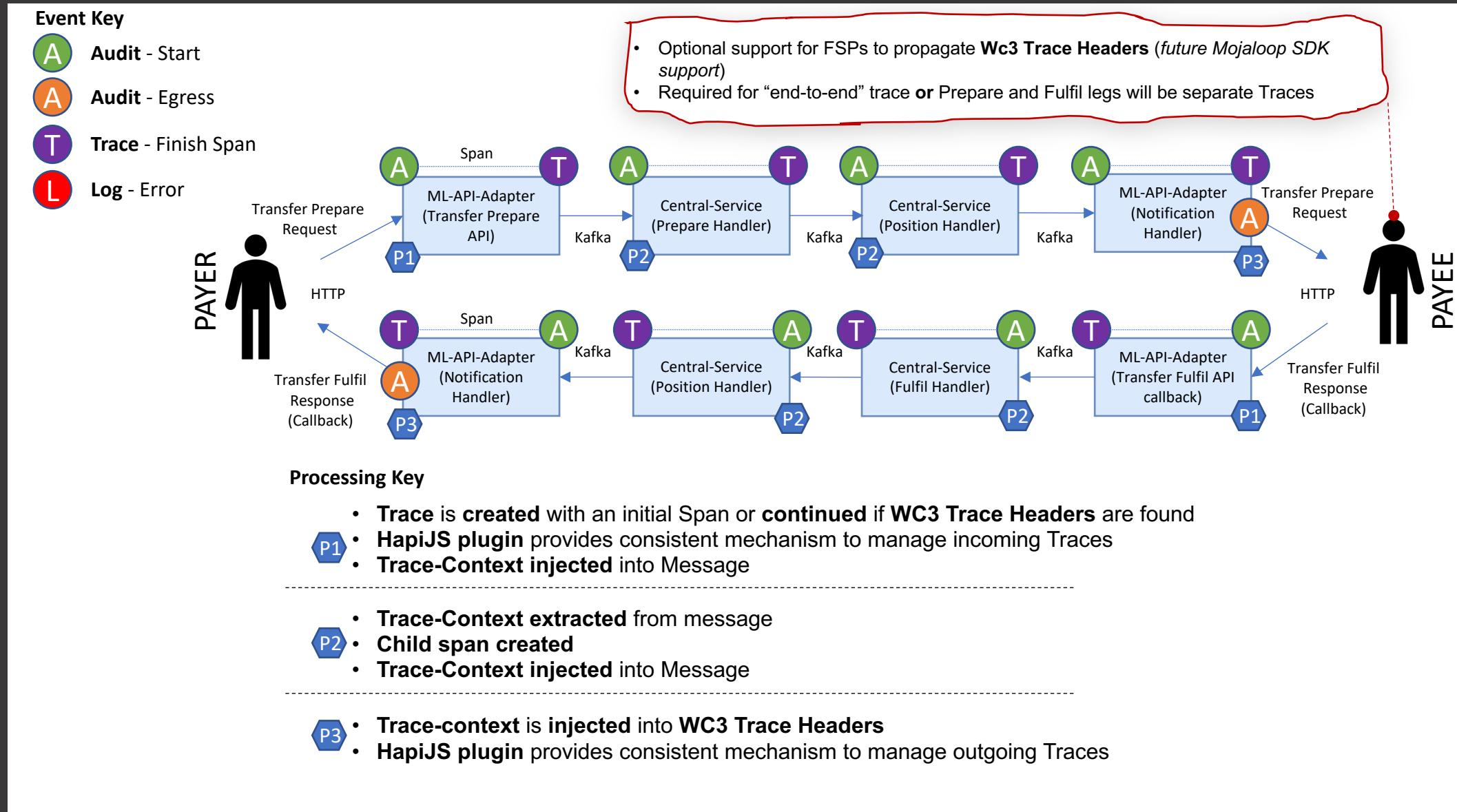
1. Publish EFK + APM changes publicly
2. Updated release for remaining Instrumented services (e.g. account-lookup-service, quoting-service, etc)

## Documentation

1. Support for developers on how to instrument services using Event-SDK.
2. Ops Monitoring on how to view, use and understand the dashboards/reports.
3. Guidelines for FSPs to use WC3 Trace Headers to support end-to-end tracing

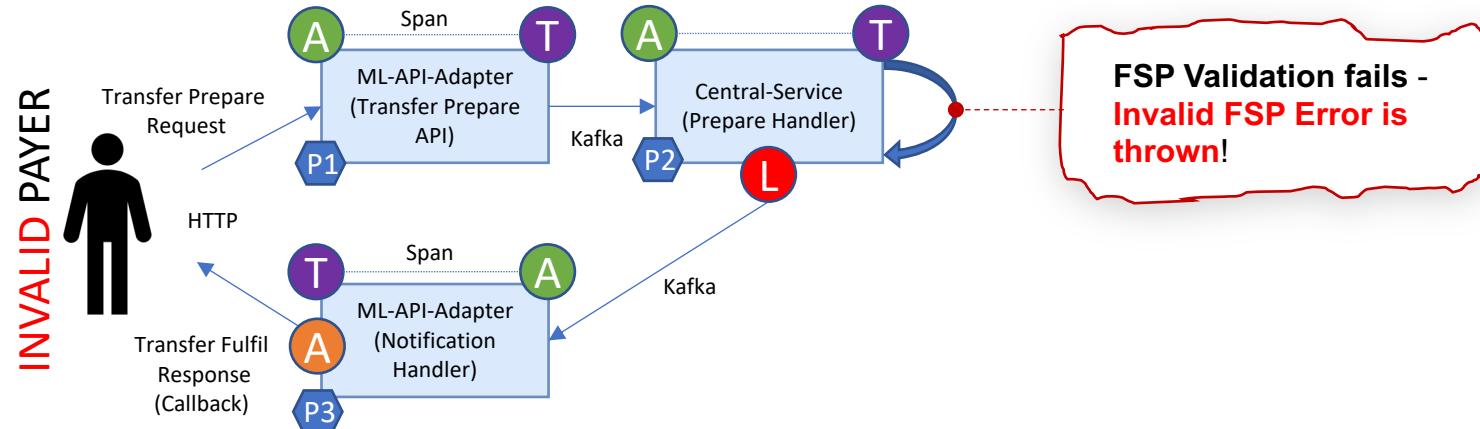


# Event Framework Demo – Success End-to-end Scenario



# Event Framework Demo – Failed End-to-end Scenario

Event Key	
A	Audit - Start
A	Audit - Egress
T	Trace - Finish Span
L	Log - Error



## Processing Key

- Trace is **created** with an initial Span or **continued** if **WC3 Trace Headers** are found
  - **HapiJS plugin** provides consistent mechanism to manage incoming Traces
  - **Trace-Context injected** into Message
- 
- **Trace-Context extracted** from message
  - **Child span created**
  - **Trace-Context injected** into Message
- 
- **Trace-context is injected** into **WC3 Trace Headers**
  - **HapiJS plugin** provides consistent mechanism to manage outgoing Traces

# PI-7 Error Framework - Overview

1. Overview
2. Functionality
3. Example
4. Roadmap

# Error Framework – Overview

## What

1. Unified framework to handle all Exceptions within the Mojaloop Stack appropriately

## Why

1. Ease and consistency in error analysis for Operational Monitoring
2. Consistent framework to handle Errors within Mojaloop Stack (aligned to Mojaloop spec v1.0)

## How

1. Utilise Mojaloop SDK Library as base for the Error Framework
  - a. Error Enums
  - b. Error Objects
2. Enhance existing Central-Services-Error-Handling library to provide:
  - a. Factory methods to help create Error Objects
  - b. Validation methods for Errors
  - c. Reformat methods to transform general errors into a valid Mojaloop Errors
  - d. Transformation methods to Mojaloop v1 Spec ErrorInformation model
3. Links to code:
  - a. <https://github.com/mojaloop/central-services-error-handling>
  - b. <https://github.com/modusbox/mojaloop-sdk-standard-components>



# Error Framework – Functionality

## Features:

1. Supported Functions
  - a. createFSPIOPError
  - b. createFSPIOPErrorFromJoiError
  - c. createFSPIOPErrorFromErrorInformation
  - d. createFSPIOPFromErrorCode
  - e. createInternalServerFSPIOPError
  - f. reformatFSPIOPError
  - g. findFSPIOPError
  - h. validateFSPIOPError
2. Stack-cause:
  - a. Added as an extension '\_cause'
  - b. Propagated internally for issue resolution
3. Enums:
  - a. Mojaloop Spec v1.0 Errors
    - i. Codes
    - ii. Description
    - iii. HTTP Status Code
    - iv. Regex Group (e.g. Generic server error – 20xx, Client Errors – 3xxx)
4. Hapi:
  - a. Common Plugin to handle validation errors & map/transform to Mojaloop Spec v1.0 Errors

## Implementation

1. Enhanced the following components with the Error Framework
  - a. MI-api-adapter
  - b. Central-Ledger
  - c. Account-lookup-service
  - d. Quoting-Service
  - e. Central-Settlements
  - f. Transaction-Request-Service
2. Improved Errors being handled by the system:
  - a. Made errors more specific where applicable
  - b. Identified & fixed incorrect errors in some locations



# Error Framework – Examples

## Factory Functions

- Standard factory functions to generate Mojaloop Spec errors
  - Errors codes are validated against standard ENUMS
  - Basic params are validated:
    - Extensions, etc

## Reformat Errors

- General Errors will be reformatted into an Internal Servier Mojaloop Error if it is not recognized
  - Error is returned as is if it's a Mojaloop Error

## Cause

- Example of the “cause” extension
  - Propagates detailed stack trace

## Validate Error Codes

- Validates strings, text or ErrorEnum objects
  - Returns ErrorEnum if successful
  - Throws an Internal Server Error on failure

```
491 factoryTest.test('reformat an FSPIOSError from another FSPIOSError returning the original error', function (test) {
492     const error = new Error('Invalid format')
493     const cause = Factory.createFSPIOSError(Errors.MALFORMED_SYNTAX, message: 'Malformed parameter test', error, replyTo: 'dfsp1')
494     const fspiopError = Factory.reformatFSPIOSError(cause)
495     test.ok(fspiopError)
496     test.deepEqual(fspiopError, cause)
497     test.end()
498 })
```

```
470
471     factoryTest.test('reformat an FSPIOSError from a general error', function (test) {
472         const cause = new Error('Test Cause')
473         const fspiopError = Factory.reformatFSPIOSError(cause)
474         test.ok(fspiopError)
475         test.deepEqual(fspiopError.toApiErrorObject(), {
476             errorInformation: {
477                 errorCode: '2001',
478                 errorDescription: 'Internal server error - Test Cause',
479                 extensionList: {
480                     extension: [
481                         {
482                             key: 'cause',
483                             value: fspiopError.stack
484                         }
485                     ]
486                 }
487             })
488         test.end()
489     })
490 }
```

```
factoryTest.test('validateFSPIOPError should validate an integer errorCode', function (test) {
  const errorCode = 2000
  try {
    const result = Factory.validateFSPIOPError(errorCode)
    test.ok(result)
  } catch (err) {
    test.ok(err instanceof Factory.FSPIOPError)
    test.fail(err)
  }
  test.end()
})
```

## ErrorInformation

- Creates ErrorInformation object for API requests & responses as Mojaloop API Model v1 Specification.

# Error Framework – Roadmap

## Error Features:

1. Enhance Error Framework to support customized Error Codes
2. Propose changes based on observations to the CCB to cater for specific use cases

## Implementation

1. Future components to enhance with Error Framework:
  - a. Sidecars
  - b. Simulators
  - c. Central-Event-Processor
  - d. Email-Notifier
  - e. Bulk-Transfers
2. QA Framework to be updated for:
  - a. Improvements to errors being more specific where applicable

# Enhancements for Operational Monitoring

## What

1. Changes done to support Operational Monitoring

## Why

1. Event spans need to be added to request object:
  - a. Spans are generated automatically on request
  - b. Error events automatically trigger on error callbacks
2. More detailed log levels needed for event types

## How

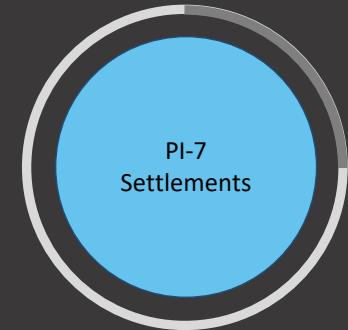
1. Enhanced existing central-services-shared to include:
  - a. Reusable plugins to be registered to Hapi
  - b. Log levels adjusted to be more in-depth and align to event and error frameworks

## Implementation

1. Enhanced the following components with common code library
  - a. ml-api-adapter
  - b. Central-Ledger
  - c. Account-lookup-service
  - d. Central-Settlements
  - e. Transaction-Request-Service

# Mojaloop End of Phase-3

## Supporting Adoption & Deployment



## PI-7: Enhancements

1. QA Framework improvements
2. Packaging, Releases
3. Idempotency in Mojaloop [Duplicate check enhancements]

## PI-7: QA Enhancements

1. Collections updated with
  - a. Tests for fees
  - b. Tests for multiple currencies in settlements
  - c. Additional tests for JWS and ALS
2. Postman repo: <https://github.com/mojaloop/postman>

## PI-7: QA Framework Collections

1. Golden Path
2. New deployment Setup
3. OSS API tests
4. OSS Feature tests
5. mojaloop-simulator collection

## PI-7: QA Enhancements – Others

1. Tests added for bug fixes
2. Reliability
3. Onboarding support
4. Tagged releases for correlating with Helm release versions



## Tests Report

Collection

Golden\_Path

Time

Sun Sep 08 2019 23:07:59 GMT+0000 (Coordinated Universal Time)

### Base Information

	Total	Failed
Iterations	1	0
Requests	230	0
Prerequest Scripts	234	0
Test Scripts	287	0
Assertions	478	0

Total run duration	7m 54.9s
Total data received	121.58KB (approx)
Average response time	16ms

Total Failures	0
----------------	---

### Requests

Pre-test-setup / Run-sample-quotes-transfers-to-prep-simulator

Send Quote

Send Transfer

p2p\_money\_transfer / p2p\_happy\_path SEND\_QUOTE

Add User - {{pathfinderMSISDN}} to payeefsp

Register Participant {{pathfinderMSISDN}} against MSISDN Simulator for PayeeFSP

Get Party Receiver

Send Quote\_SEND

Send Transfer

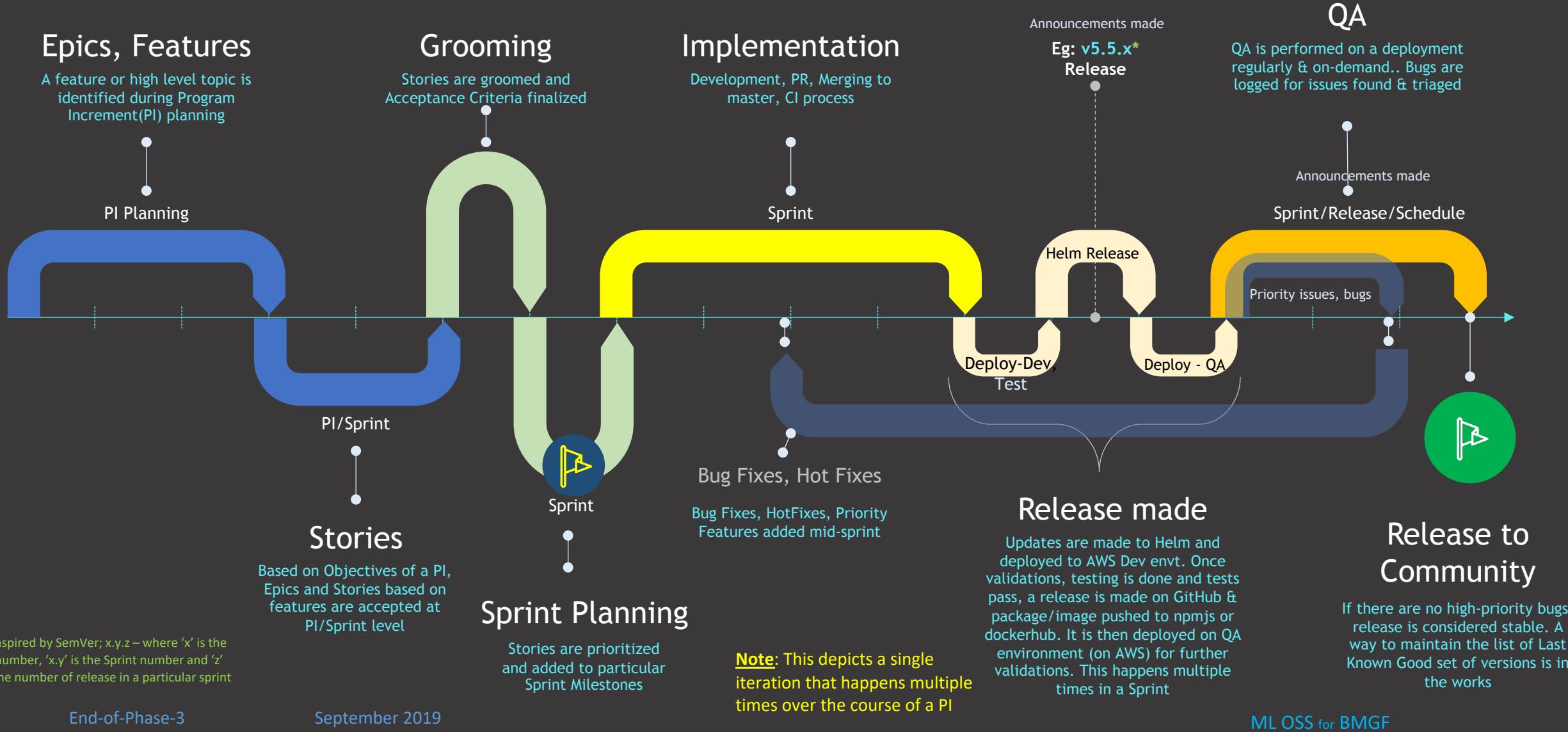
p2p\_money\_transfer / p2p\_happy\_path RECEIVE\_QUOTE

Add User - {{pathfinderMSISDN}} to payeefsp

## PI-7: Enhancements to Releases

1. OSS Release process (published earlier)
2. Releases being made after being tested on AWS Dev environment
3. Stable versions of Helm
  - a. v7.4.0
  - b. v7.4.3

# ML OSS: Release Mechanism



**v7.4.3 Release**

mdebarros released this 10 days ago

**Helm release changes**

- Added missing config items to ml-api-adapter config for:
  - MAX\_CALLBACK\_TIME\_LAG\_DILATION\_MILLISECONDS <-- default to 100ms
  - MAX\_FULFIL\_TIMEOUT\_DURATION\_SECONDS <-- default to 240s (4m)
- Added configs for the above config items in the values.yaml
- Bump to mojaloop chart to reflect the above changes
- Updated central-ledger MIGRATION init-container configs to include:
  - RUN\_DATA.Migrations: true <-- this will enable or disable the data migration to run as part of the DB migration scripts. By default, this is set to true.

**Application versions**

Application versions that are supported for this update:

- central-ledger: v7.4.3
- ml-api-adapter: v7.4.1
- central-settlement: v7.4.0
- central-event-processor: v7.1.1
- email-notifier: v7.3.0
- account-lookup-service: v7.4.0-snapshot
- quoting-service: v7.4.0-snapshot
- simulator: v7.2.1
- bulk-api-adapter: v7.1.1-snapshot

**Application release notes**

- central-ledger - <https://github.com/mojaloop/central-ledger/releases/tag/v7.4.1>
- ml-api-adapter - <https://github.com/mojaloop/ml-api-adapter/releases/tag/v7.4.3>
- central-settlement- <https://github.com/mojaloop/central-settlement/releases/tag/v7.4.0>
- central-event-processor - <https://github.com/mojaloop/central-event-processor/releases/tag/v7.1.1>
- email-notifier - <https://github.com/mojaloop/email-notifier/releases/tag/v7.3.0>
- account-lookup-service - <https://github.com/mojaloop/account-lookup-service/releases/tag/v7.4.0-snapshot>
- quoting-service - <https://github.com/mojaloop/quoting-service/releases/tag/v7.4.0-snapshot>
- simulator - <https://github.com/mojaloop/simulator/releases/tag/v7.2.1>
- bulk-api-adapter - <https://github.com/mojaloop/bulk-api-adapter/releases/tag/v7.1.1-snapshot>

**Operational Chart versions**

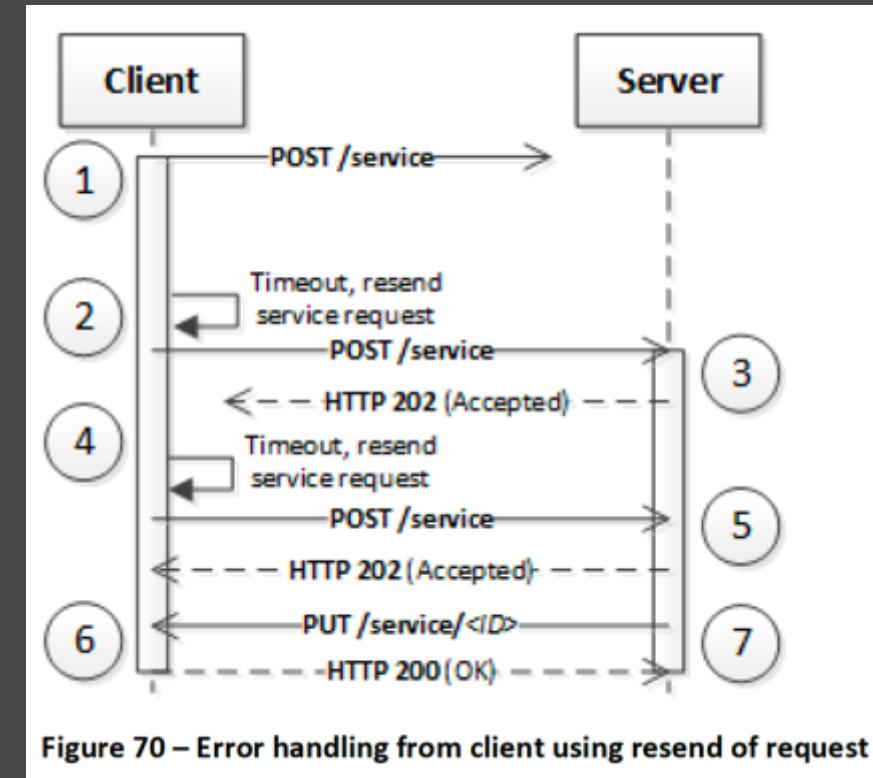
- EFK: v7.1.0
- Promfana: v4.4.0

# Idempotency: Definitions and general HTTP concept

1. Definition
  - a. Idempotence = (lat.) idem + (fr.) potence = same + power
  - b. Potential result of performing an action: idempotent vs. nullipotent
2. HTTP Concept
  - a. Null-potent (safe) HTTP methods: GET, HEAD, OPTIONS, TRACE
  - b. Idempotent methods: PUT (create **or** replace), DELETE
  - c. Non-idempotent: POST, PATCH, CONNECT
3. Fundamental difference between PUT and POST and fault-tolerant APIs

# Idempotency: Mojaloop API Specification

1. Idempotent Services in Server (3.2.5)
  - a. GET and POST must be idempotent
2. Client Missing Response from Server (9.4)
3. Duplicate Handling (3.2.5.1)
  - a. Ignored resend request
  - b. Resend and GET
  - c. Modified request



# Idempotency: Switch Implementation PI-7

1. Handling of Invalid Transfer Prepare Requests
  - a. Idempotent, but different response, due to transfer data rollback
2. Accepts Multiple Transfer Fulfil (Reject) Requests
  - a. Sending a fulfilment multiple times is possible
  - b. Different response on resend
  - c. ID/message hash logic replaced by overcomplicated handling
3. Absence of a common library for Duplicate Handling
  - a. Individual transfers, bulk transfers, quotes, transaction requests

# Idempotency: Duplicate Handling Enhancements

1. Handling of Invalid Transfer Prepare Requests
  - ✓ Stored to DB and same response
2. Accept **Single** Transfer Fulfil (Reject) Request from Payee FSP
  - ✓ Ensures that sending of fulfilment is not exploitable
  - ✓ Same response on resend
  - ✓ Simple code – easy to maintain, extend and test
3. Request Duplicate Check added to central-services-shared
  - ✓ Implemented for Individual transfers

# PI-7 Updates: Simulator

## API operations

- [●] Transfers
- [●] Parties
- [●] Participants
- [●] Quotes
- [●] Transaction Requests
- [●] Oracle API operations

## Bulk Transfers

- [●] POST - Create
- [●] PUT - Update
- [●] PUT - Update Error
- [●] GET - Requests ID

## PI-7 Changes / Enhancements

1. Added Bulk Transfer operations to mock a FSP supporting /bulkTransfers endpoints
2. Bug fix for incorrect PUT /parties error endpoint (/parties/<Type>/<ID>/error)

## Future Enhancement

1. Move bulkTransfers support to Mojaloop-simulator
2. Decision on oracle support for QA [Stand-alone or in to Mojaloop-simulator]
3. Consolidate Simulator capabilities to support QA framework

## Mojaloop Simulator [Mowali]

1. Initial Analysis done on the donated codebase
2. Collaboration with CrossLake, Community
3. Setting up Continuous Integration
4. Move towards having a single simulator in OSS

## Key

- [●] Added in PI
- [●] Fully implemented
- [●] Legacy Code
- [●] Partially implemented
- [●] Not implemented
- [○] Future Roadmap



mojaloop

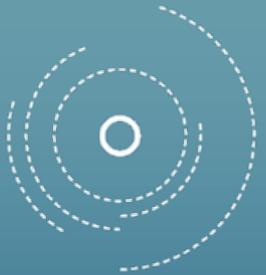
## PI-7: Updates

---

End-of-Phase3

# PI-7: Updates

1. Performance Testing Capabilities
2. Bug Fixes
3. On-boarding improvements
4. Deployment enhancements



mojaloop

## Performance Testing Capabilities

---

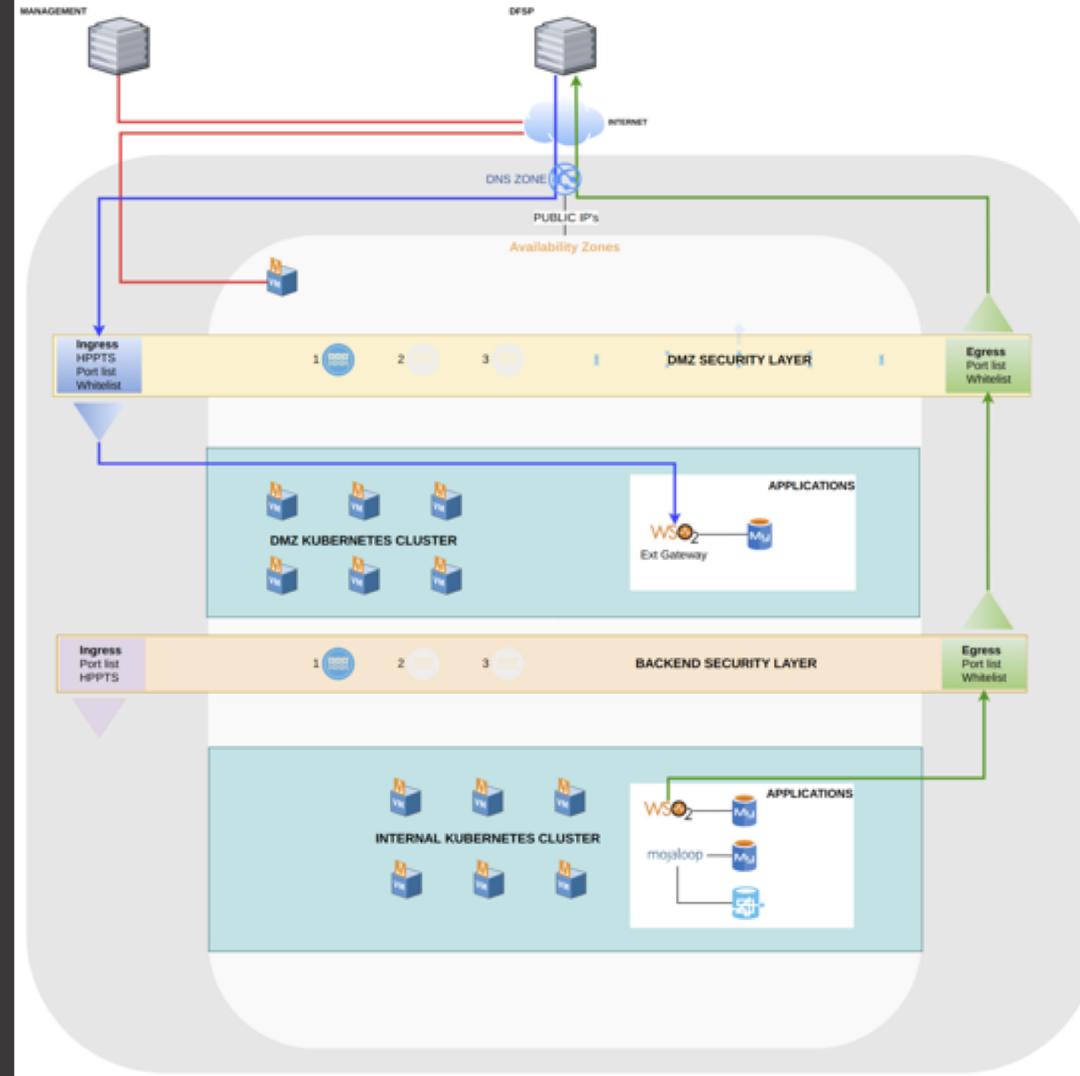
End-of-Phase-3

# PI-7: Updates

1. Environment setup Performance Testing (future scenarios)
  1. Mojaloop Core (regression)
  2. Full Lab (WSO2 + Security + Core, etc)
2. EFK support , Dashboards for viewing analyzing numbers
  - a. Instrumentation of services
  - b. Performance Dashboards (future)
3. Scripts for performance testing

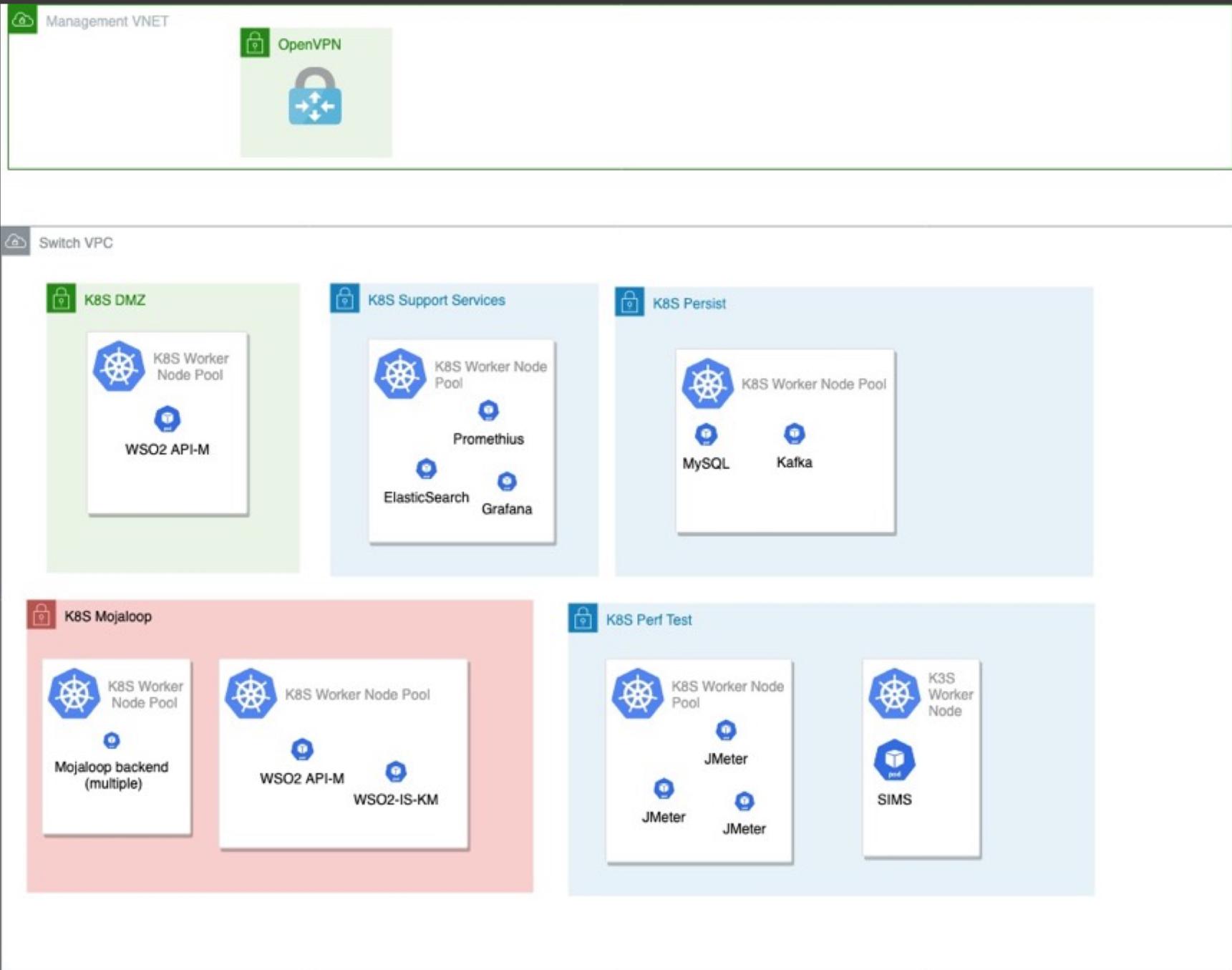
# IaC Environment

- Terraform IaC from Last Convening being published in Mojaloop Repo
- Builds 2 Clusters with WSO2 internal and external GWs with Identity Server



## Next Steps for IaC

- Add ability to specify VM AMIs for different server roles (k8s worker nodes, etc)
- K8s clusters to have additional parameterized features added:
  - Numbers of worker nodes
  - Node pool labeling for workload segregation
- Configurable number of k8s clusters for separating workloads for testing related tasks:
  - K8s cluster for jmeter workloads for driving tests
  - K8s cluster for running EFK, SIMS
  - K8s cluster for MySQL, Kafka
- Optimize IaC for fast spin up and ability to persist results before tear down



## PI-7: Bug Fixes - 1

1. Headers/Hapi issues
  - a. Standard libraries for headers transformation
  - b. Raw Payload Hapi plugin created in shared-library
2. Numbers processing – Precision change and QA
  - a. Number precision fix, changed from 2 to 4 decimal points in DB
  - b. Number library in progress

## PI-7: Bug Fixes - 2

1. Error codes issues
  - a. Standard libraries for Error Handling
  - b. Using consistent error messages (aligned to Spec v1.0)
2. Bug fixes in PI7 as part of packaging: Fixing, validating bugs for a quality release of Mojaloop
3. ALS, JWS issues addressed
4. 42 Bugs addressed in PI-7 (Closed)

## PI-7: Bug Fixes - 2

1. Onboarding docker commands, instructions fixed
  - a. ml-api-adapter
  - b. central-ledger
  - c. Scripts to populate test data
2. Postman collections for onboarding fixed



# Deployment Improvements for PI7

## Helm Release Charts

### Public releases for PI-7

- v7.4.3 (**latest - stable**)
- v7.4.1
- v7.4.0

### Mojaloop v7.4.3

- [●] ML-API-Adapter v7.4.1
- [●] Central-Ledger v7.4.3
- [●] Central-Settlements v7.4.0
- [●] Central-Event-Processor v7.1.1
- [●] Email-Notifier v7.3.0
- [●] Account-Lookup-Service v7.4.0-snapshot
- [●] Quoting-Service v7.4.0-snapshot
- [●] Simulator v7.2.1
- [●] Bulk-API-Adapter v7.1.1-snapshot

#### Key

- [●] Added in PI
- [●] Updates made in PI
- [●] In-progress
- [x] Deprecated

## Helm Maintenance

- Updated to support enhanced **Health-Endpoints**
- Added **debug** config to all charts
- Integrated the following components into the **Mojaloop Helm chart**
  - Bulk-API-Adapter
  - Upgraded EFK charts (*pending publish*)
  - Added APM to EFK chart deployments (*pending publish*)
  - Added Event-Stream-Processor (*pending publish*)
- Integrated **Event-Sidecar** into:
  - ML-API-Adapter (*pending publish*)
  - Central-Ledger (*pending publish*)

## Release Notes:

- <https://github.com/mojaloop/helm/releases/tag/v7.4.3> (**latest - stable**)
- <https://github.com/mojaloop/helm/releases/tag/v7.4.1>
- <https://github.com/mojaloop/helm/releases/tag/v7.4.0>

## Future Roadmap

- Integrate Mojaloop-Simulator
- Integrate Transaction-request-service

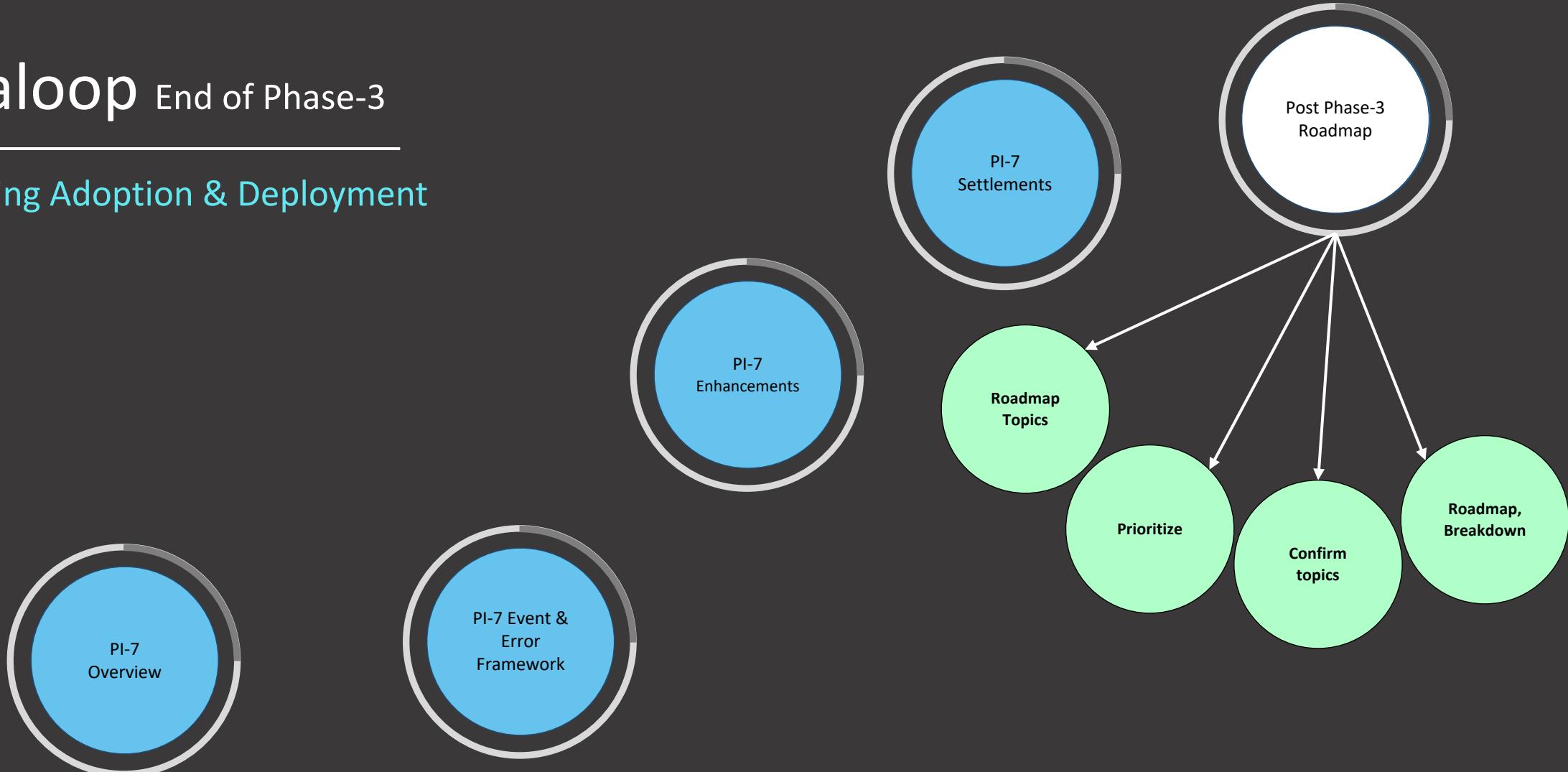
# Mojaloop End of Phase-3

## Supporting Adoption & Deployment



# Mojaloop End of Phase-3

## Supporting Adoption & Deployment



# Post-Phase3 Roadmap: Candidate Topics

- |   |   |
|---|---|
| <ol style="list-style-type: none"><li>1. OSS Settlement</li><li>2. Performance Testing, Tuning</li><li>3. Packaging, Bug Fixing</li><li>4. Community support</li><li>5. Planning, Backlog for Phase-4</li></ol> | <ol style="list-style-type: none"><li>6. Security</li><li>7. Bulk Transfers (Incorporating PoC)</li><li>8. Fraud &amp; AML</li><li>9. Auditing</li><li>10. Support Reporting mechanism</li><li>11. Central-Hub</li><li>12. Resilience [DB, Retries on notifications]</li><li>13. Operations API</li><li>14. Event enhancements:<ul style="list-style-type: none"><li>• Integration with ticketing for Ops</li><li>• Custom Dashboards</li></ul></li></ol> |
|---|---|

# Post-Phase3 Roadmap: Candidate Topics - 3

- 15. Merchant Payments
- 16. PoC for FXP and/or CXP
- 17. OSS Lab Refresh, IaC (Infrastructure as Code)
- 18. Implementation of Settlements v2.0
- 19. GitLabs Migration
- 20. Perf testing with edge cases

# Post-Phase3 Roadmap: Candidate Topics - 1

1. OSS Settlement
2. Performance Testing, Tuning
3. Packaging, Bug Fixing
4. Community support
5. Planning, Backlog for Phase-4
6. Cross Currency / Border (Coil)

# Post-Phase3 Roadmap: Discussed Topics

1. OSS Settlement
2. Cross Border / Network
3. Performance Testing, Tuning
4. Resilience
5. Central Hub / Portal

# PI-8 Objectives ?

1. Cross Network (Currency / Border) - Coil
2. ISO Adapter – Coil
3. Packaging OSS (readiness)
4. Performance testing and tuning
  - a. Perform Performance testing on a Mojaloop Switch
  - b. Tune performance to match performance at the end of PI3
5. OSS Settlement
6. Phase-4 Roadmap
  - a. Support Phase-3 packaging, community support
  - b. Prepare backlog for Phase-4

# Post-Phase-3 Schedule

PHASE THREE	Start	End	Weeks	Notes
PI 6 On-Site	4/16/19	4/18/19	3 days	Johannesburg
PI 7 On-Site	6/25/19	6/27/19	3 days	Arusha
End-of-Phase3 On-Site	9/10/19	9/12/19	3 days	<b>Abidjan</b>
8.1	9/16/19	9/27/19	2 weeks	Packaging, Perf
8.2	9/30/19	10/11/19	2 weeks	Performance
8.3	10/14/19	10/25/19	2 weeks	Performance
8.4	10/28/19	11/08/19	2 weeks	Performance
...	...	...	...	...
Phase-4 Kick-off	01/28/20	01/30/20	3 days	<b>Jo'burg</b>
Get Involved – Weekly Scrum of Scrum Meetings for the OSS community				

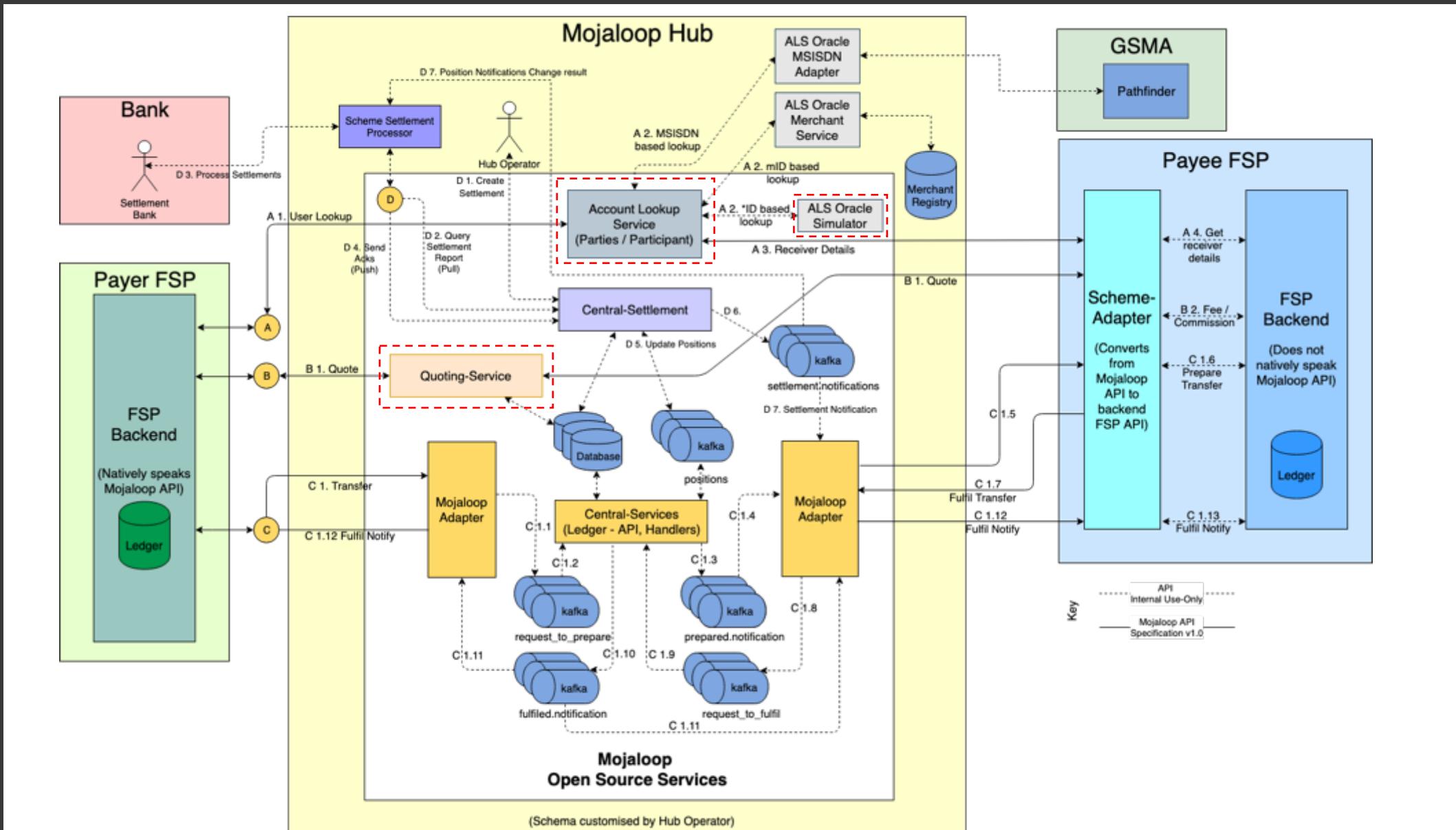


mojaloop

## Appendix

### End-of-Phase3

# PI7 – Architecture Overview



# Standardising Common Code – Functionality

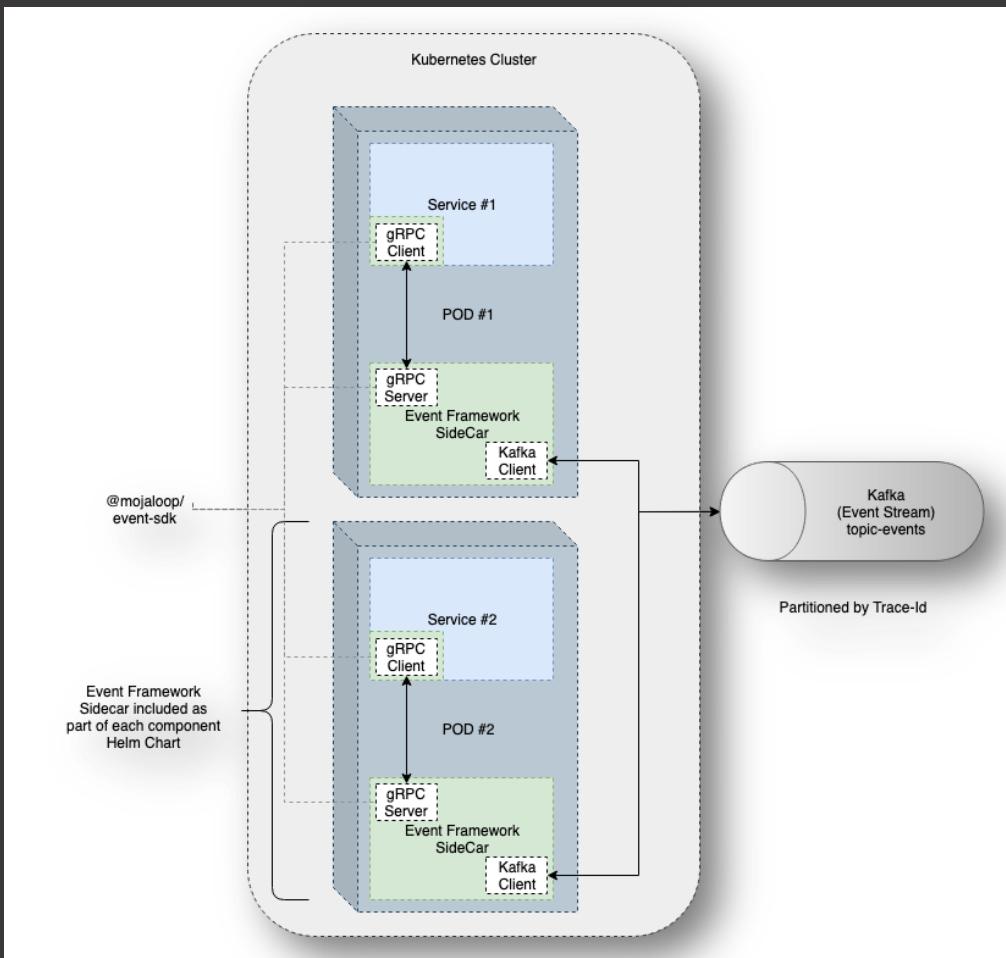
## Features:

1. Supported Functionality
  - a. Transform Headers
  - b. Send Requests
  - c. Streaming Protocol Utilities
  - d. Kafka Producer
  - e. Kafka Consumer
  - f. Kafka Utilities
  - g. Endpoint Cache
  - h. Time
2. Hapi Plugins:
  - a. rawPayloadToDataUri
  - b. eventPlugin
  - c. Fspiop-api-protocol-version-header-validator
3. Enums:
  - a. Accounts
  - b. Endpoint Templates
  - c. Events
  - d. Http
  - e. Kafka
  - f. Transfers

## Implementation

1. Enhanced the following components with common code library
  - a. MI-api-adapter
  - b. Central-Ledger
  - c. Account-lookup-service
  - d. Central-Settlements
  - e. Transaction-Request-Service

# Event Framework – Deployment Arch & Roadmap



Deployment Architecture

End-of-Phase-3

September 2019

## 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

ML OSS for BMGF

# Event Framework - Tracing Dashboard Example

