

# Mojaloop Core & Releases #PI-29

MojaCom29 Nairobi

Sam Kummary, Mojaloop Foundation



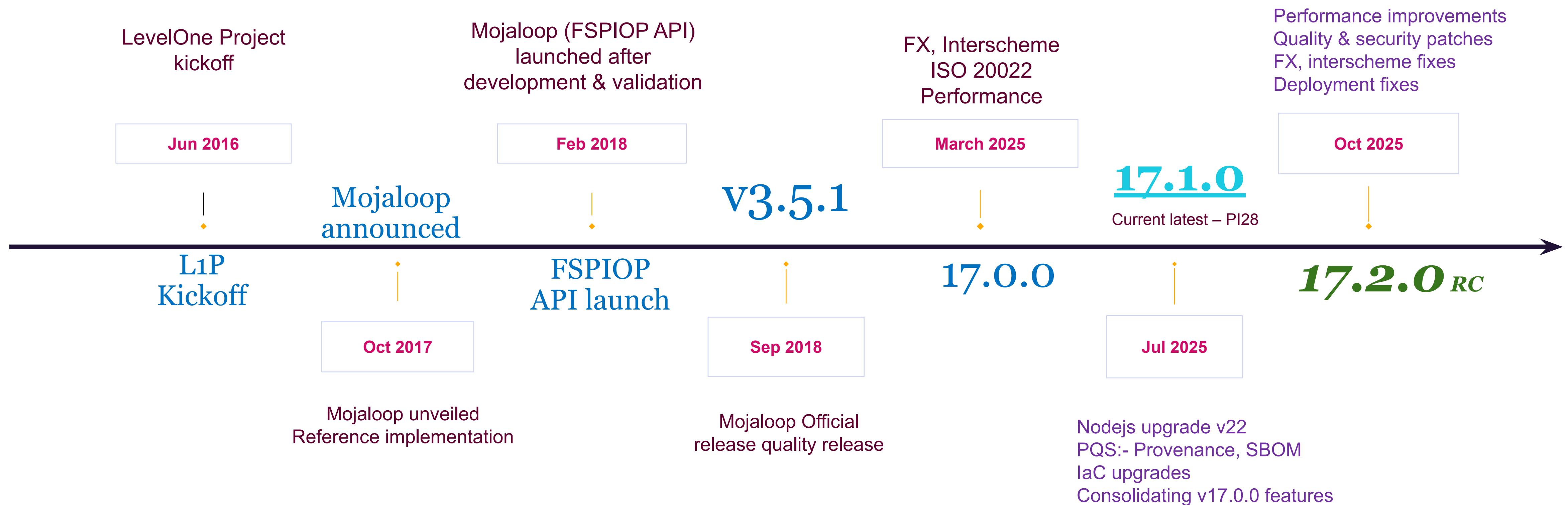
# Agenda



1. Mojaloop Release v17.1.0, v17.2.0
  - a. Features, updates
  - b. Release process
2. Platform quality and security
3. Roadmap

# Mojaloop

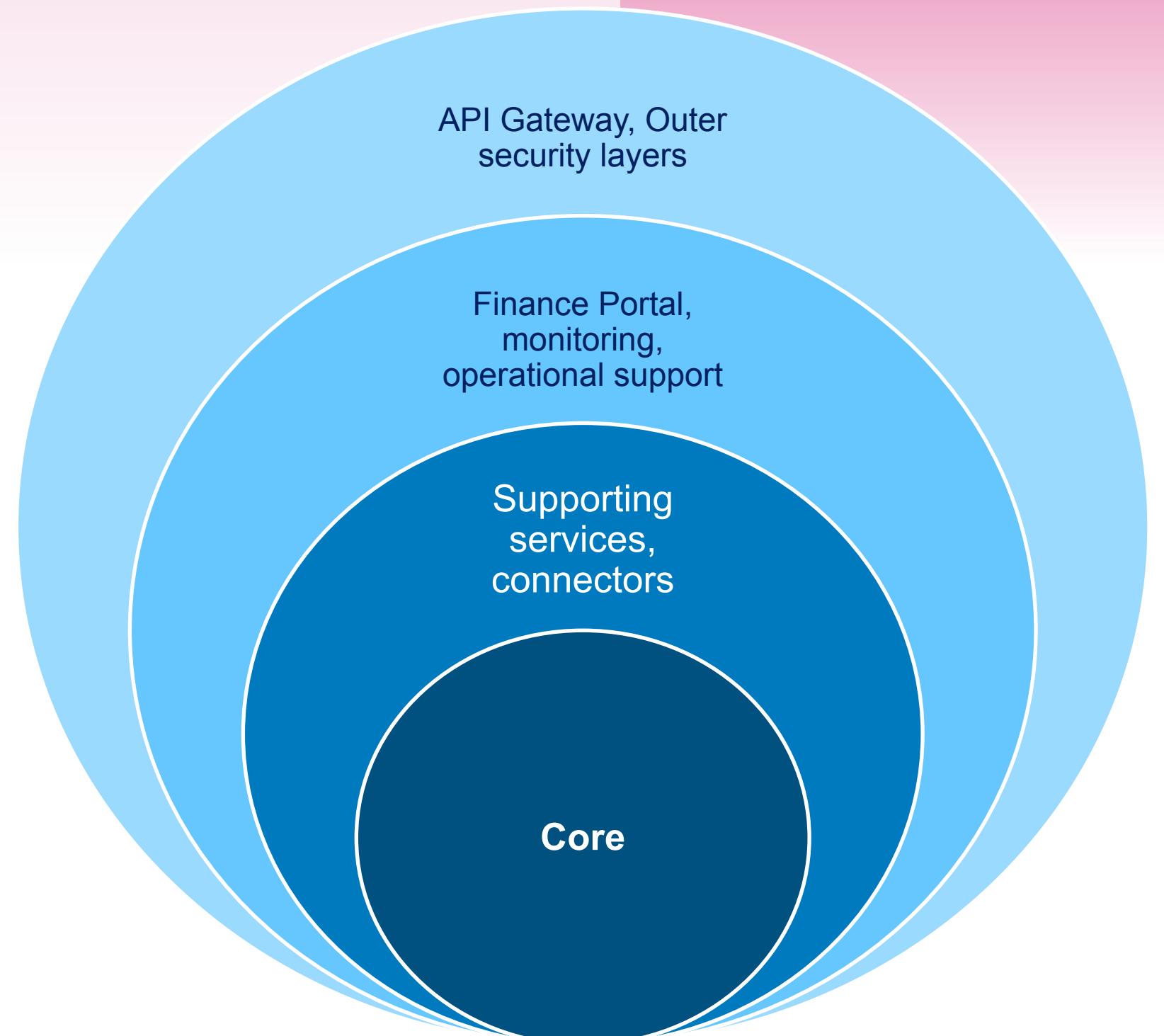
PI-28



\* Version numbers guided by SemVer

# Mojaloop Platform

1. **Mojaloop Core**  
Transactional hub components and settlement i.e. API layer, central ledger (including bulk), settlement, ALS, quoting, trx req, Admin
2. **Mojaloop Connector(s)**  
Best practice Mojaloop connectors. I.e. Third-party services, sdk adapter/ml-connectors
3. **Supporting services**  
Interop: Connect with settlement banks, Onboarding FSPs, integrations (connection management)
4. **Mojaloop Toolkit (TTK), ITK instances, Simulators, ALS support, CEP, Events processing**  
Oracles - MSISDN or Merchant Registries
5. **Finance portal (RBAC), Reporting**  
Hub operator UI & API experience layer that is Implemented by business operations framework.
6. **Monitoring support (Operational support)**  
Operational support & tracing (e.g. EFK / Prometheus, Grafana, Loki, etc) [IaC: Grafana, Prometheus & Loki]
7. **API Gateway**  
IP filtering, Ingress/Egress, Firewalls, Security support (JWS, mTLS) [IaC provides this via Istio (older versions used WSO2)]  
Adds in the rules and enforces the Mojaloop Security design
8. **Infrastructure**  
Kubernetes, Infrastructure as Code, Automated deployments (IaC)  
Deployment artifacts - Helm Charts, Docker Containers, npm packages



# Mojaloop v17.2.0



## Mojaloop Release: Core

1. Base use-cases
  1. P2P, R2P, Bulk, *PISP v1*
  2. FX, Inter-scheme
  3. ISO 20022 messaging
  4. *Performance, quality & security improvements*
2. Settlement: Deferred-net, Continuous Gross, FX support
3. Admin: Participant Lifecycle API
4. Third-party payments v1 (PISP v2 development in progress)

# Mojaloop IaC bundle



Bundled Mojaloop IaC Release (Mojaloop IaC Release 5.x.x / 6.x.x)

1. Gateway & Load Balancers
2. RBAC via Istio and Ory Stack
3. Mojaloop Security (mTLS, JWS, ILP)
4. Finance Portal (Business operations) – support for FX & inter-scheme
5. Mojaloop Connection Manager (MCM)
6. Reporting
7. Control Plane for CI/CD (GitLab) – CC v2
8. Technical Operational Dashboards (Monitoring)
9. Participant Onboarding Support (Connection Manager)
10. Optional Payment Manager deployment (for DFSPs)

# Mojaloop Switch Functionality – FSPIOP Use Cases (PI-28)

## Mojaloop FSPIOP API v1.1, v2.0 Use cases



### Payer-Initiated Transaction

- [●] P2P Transfers
- [●] Customer-Initiated Merchant Payment
- [●] Customer-Initiated Cash-out - Receive Amount
- [●] Customer-Initiated Cash-out - Send Amount
- [●] ATM-Initiated Cash-out
- [●] Refund
- [●] ***Foreign Exchange and Inter-scheme***

### 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

- [●] Supported with functional tests
- [●] Supported
- [●] Proof of Concept
- [●] Not implemented
- [○] Out of Scope

# Switch Functionality: Mojaloop End-points (PI-28)



## FSPIOP API v1.1, v2.0 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

### fxQuotes\*

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

### fxTransfers\*

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

### Key

- [●] Supported with functional tests
- [●] Supported
- [●] Proof of Concept
- [●] Not implemented
- [○] Out of Scope

\*Enhancements in PI-26,27

# Switch Functionality – Administration (PI-28)



## Admin API v1.0 Specification

### Participants

- [●] Manage Participants
- [●] Create Initial Values
- [●] Query
- [●] Update
- [●] Manage positions, limits
- [●] Create Initial Value
- [●] Query
- [●] Update
- [●] Manage Callback URLs
- [●] Create Initial Value
- [●] Query
- [●] Update

### Settlement Models

- [●] Settlement models
- [●] Configure
- [●] Query

### Participant Accounts

- [●] Manage accounts
- [●] Create
- [●] Query
- [●] Update
- [●] Funds In/Out

### Transactions

- [●] Transactions
- [●] Query

## ALS Oracle API v1.0 Specification

### Oracles (ALS)

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

### Services

- [○] Services FXP\*
- [○] Get (Query)
- [○] PUT

### Key

- [●] Supported with functional tests
- [●] Supported
- [●] Proof of Concept
- [●] Not implemented
- [○] Out of Scope

\* Enhancements in PI-26,27

# Switch Functionality – Settlement (PI-28)

## Settlement API v2.0 Specification



### 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

### Settlements v2

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

#### Key

- [ ] Supported with functional tests
- [●] Supported
- [●] Proof of Concept
- [●] Not implemented
- [○] Out of Scope

# Switch – Third-party (3p) API v1.0 (PI-28)

## Third-party API v1.0 Specification



### Linking

- [●] Pre-Linking
- [●] Discovery
- [●] Request consent
- [●] Authentication
- [●] Grant consent
- [●] Credential registration

### Un-Linking

- [●] Un-linking

### Third-party (3p) Transfers

- [●] Discovery
- [●] Agreement
  - [●] 3p transaction request
  - [●] 3p authorization
  - [●] Signed authorization
  - [●] Validate authorization
- [●] 3p Transfer
- [●] Query trxreq status

### PISP v2.0 (in progress)

- [●] Discovery
- [●] Agreement
  - [●] 3p transaction request
  - [●] 3p authorization
  - [●] Signed authorization
  - [●] Validate authorization
- [●] 3p Transfer

#### Key

- [●] Supported with functional tests
- [●] Supported
- [●] Implementation in progress
- [●] Not implemented
- [○] Out of Scope

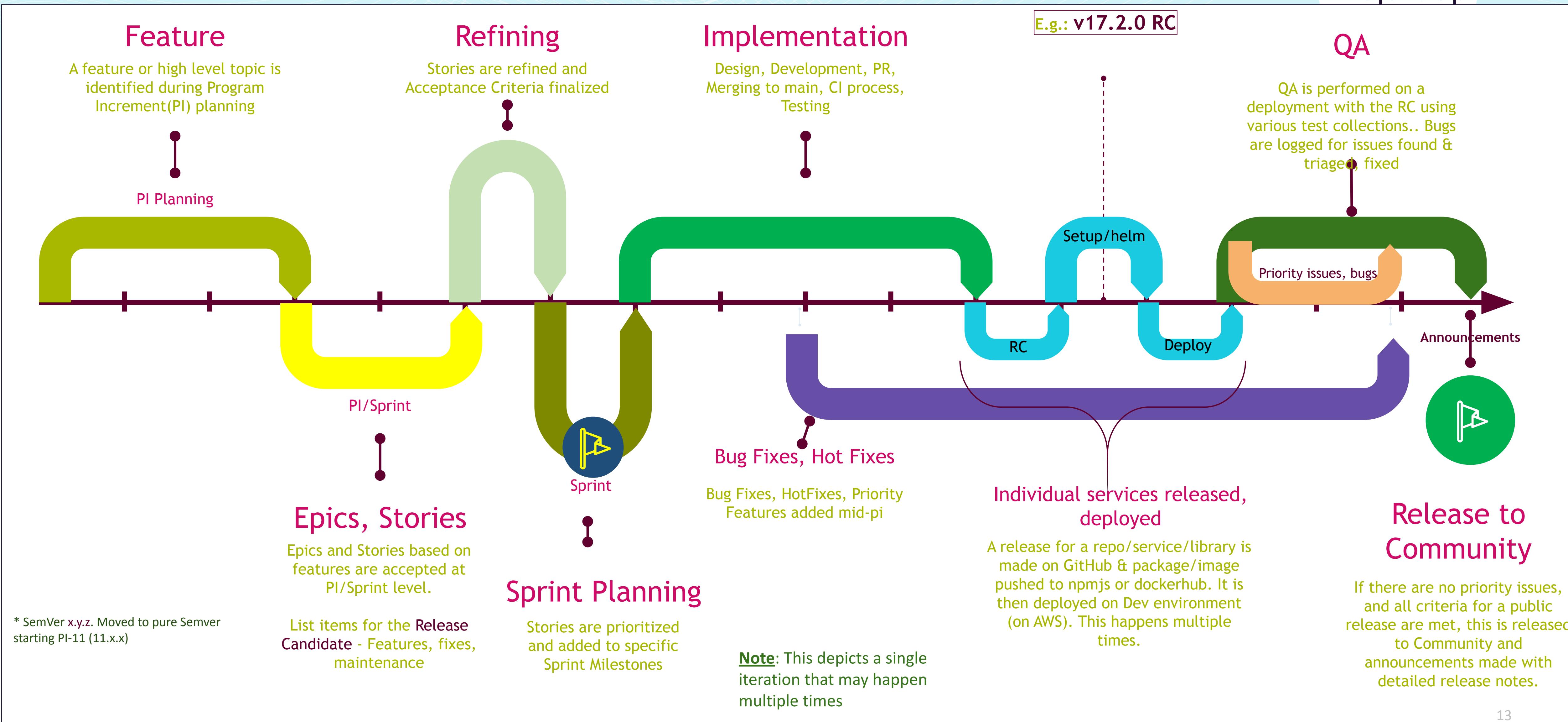
# Releases - WHY upgrade?



## Mojaloop v17.0.0 (v17.1.0 and v17.2.0 RC)

1. Improved Performance and Features
2. Enhanced Security (Patching, TLS support to DB)
3. Bug Fixes and Stability (FX, inter-scheme, lookup, logging)
4. Future-Proofing (Prepare for future releases)

# Mojaloop Release Process - overview



# ML Releases: PI-24/27 update for reference



v15.1.0  
(PI-21)

v15.2.0  
(PI-22/23)

v16.0.0  
(PI-23)

v17.0.0  
(PI-26)

1. Test against upgraded QA/Dev env & QA/Dev with K8s v1.26/1.27
2. Refactored QA/Dev env ease of maintenance with EKS
3. Enhanced SDK support for Request-to-Pay (R2P)
4. Follow-up bug-fixes
5. IaC v4.1.0 with support for Mojaloop v15.1.0

1. Upgraded QA/Dev env & Tested against K8s v1.28
2. General maintenance
3. Node upgrade (v18 LTS)
4. Performance setup, profiling and document bottlenecks (perf workstream)
5. Performance Stream improvements (enhancements, fixes, instrumentation, & contribs)
6. Automated, CI tests on ML releases with Helm snapshot support

1. General maintenance
2. Node upgrade for supporting services not updated in v15.2.0
3. Performance enhancements done by perf-workstream included
4. Automated CI steps on Mojaloop releases
5. IaC changes - new and refactored version

1. **FX**, Inter-scheme support
2. Support for **ISO 20022** messaging
3. Merchant payments features
4. **PQS** updates - security patches and upgrades
5. **Performance** enhancements
6. General maintenance

# Release timeline PI29: MLF Product & work-streams



1. Nodejs upgrade to v22
2. PQS - security patches, SBOM & vulnerability management
3. MySql2 upgrade (SSL support); DA issue [#131](#)
4. Cleanup unused charts
5. Documentation - release automation
6. Merchant services updates
7. General maintenance, bug fixes (knex)

1. Performance improvements
2. PQS - security patches & vulnerability management
3. Deployment improvements - Helm and IaC
4. Merchant services updates
5. General maintenance, bug fixes (FX, inter-scheme - improve discovery, logging, error-codes)

1. Deployment improvements
2. Settlements improvements
3. TigerBeetle integration into Mojaloop Core
4. PISP Bulk API support ([#107](#))
5. PISP v2 support
6. Enhance tracing components to use latest open-tracing standards
7. Automated CI testing on Helm PRs & Releases - completed
8. Migrate Charts (i.e. BoF) to Helm
9. Patch notification support for failures

# Mojaloop Core & Releases - Ongoing activities



1. RC testing for v17.2.0 in progress; GP tests passing
2. Vulnerabilities related updates - PQS
3. Performance whitepaper workstream support, performance improvements
4. Mojaloop deployment enhancements, documentation updates and related testing

# Mojaloop Releases - Enhancements



1. Release cycle cadence, Support discussion (LTS)
2. Performance profiling releases - in progress
3. Closer integration with downstream tooling – Finance Portal, IaC, Test cases
4. Deployment enhancements (helm, IaC)
5. Documentation updates



# Roadmap PI-29

1. v17.2.0 release & post-release activities
  - a. Performance profiling
  - b. IaC support
  - c. Finance portal enhancements
2. Mojaloop v17.x.y / v18.0.0 release planning and support for incoming changes
3. Vulnerabilities related updates - PQS
4. Performance whitepaper workstream support
5. Facilitate Mojaloop deployment enhancements
6. Documentation updates and related testing



# Mojaloop Platform Quality & Security (PQS) PI-29

# Mojaloop PQS: In addition to CI steps



1. GitHub tools – CodeQL, Dependabots, Synk, Co-pilot secret scans
2. Peer reviews of contributions
3. Security audits every PI
4. Review security, quality processes during releases
5. Sonarcloud scanning
6. Test coverage standards
7. End-to-end test suites
8. Deployment validation
9. Release checklist

# Sample Mojaloop CI pipeline



circleci MAINTENANCE

mojaloop

Dashboard Project Workflow

All Pipelines > central-services-shared > build\_and\_test

build\_and\_test ✓ Success

Organization Home

Pipelines Projects Deployments Insights Self-Hosted Runners Organization Settings Plan

Checkout method is changing  
Blobless checkout becomes the default - verify your builds still work or switch to full checkout by November 3

Duration / Finished Tag Commit Author

3m 15s / 10d ago v18.35.0-snapshot.0 2e176cd kirgene

Setup 35s

Deprecations 54s

Build 3s

Unit tests 49s

Vulnerability check 44s

Lint 42s

License audit 36s

Coverage 1m 11s

Functional tests 4s

Grype scan 46s

Integration tests 39s

License scan 3s

Docker snapshot 3s

NPM snapshot 33s

Insights Rerun ...

The diagram illustrates a CI pipeline structure. It starts with a 'Setup' step (35s) which branches into 'Deprecations' (54s) and 'Build' (3s). The 'Build' step further branches into several sub-tasks: 'Unit tests' (49s), 'Vulnerability check' (44s), 'Lint' (42s), 'License audit' (36s), 'Coverage' (1m 11s), 'Functional tests' (4s), 'Grype scan' (46s), 'Integration tests' (39s), and 'License scan' (3s). A callout arrow points from the text 'Grype scan 46s' to the 'Grype scan' step in the pipeline diagram.

# Mojaloop PQS: Mojaloop v17.2.0 RC



Review core and related repositories released with Mojaloop for

1. Gype scanning for sources
2. Dependabot alerts are addressed (moderate, high and critical)
3. *SBOMs created, published for all repositories published with a Mojaloop release*
4. *Aggregated SBOM for Mojaloop Helm attached to release notes*
5. *Provenance of Mojaloop helm charts*
6. Ensure main branch is protected and collaborator list is up-to-date
7. Ensure open PRs are addressed, closing stale PRs
8. Update audit exceptions json file to remove exceptions
9. Ensure codeowners file is current

Reviewing updates and vulnerabilities (GitHub, NVD, other related media)

1. Take necessary action for Mojaloop services as required
2. Provide guidance and mitigate exposure to Mojaloop following the Mojaloop CVD and Cybersecurity policies

# Mojaloop PQS: artefacts provenance



## Provenance for helm charts

1. Helm charts signed during publishing
2. Signed charts can be verified during install or pre-install steps
3. Helm charts commands tested and verified
4. Options to include in scripts / CI tested
5. Available for both standard releases and dev releases
6. Introduced with v17.1.0

# Mojaloop PQS: Roadmap PI-29



1. Vulnerability management in Mojaloop
2. Hardening of Mojaloop releases - continuation
3. Github code scanning - address issues
4. SonarCloud – analyze issues in detail
5. OpenSSF badge – self evaluation
6. Review tooling used in CI/CD - CI tools, license checkers
7. Documentation enhancements regarding patching, vulnerability management
8. Provenance of Mojaloop artefacts (apart from helm charts)
9. Finance portal: refactor older react components; improve quality standards

# Mojaloop PQS: General Objectives



1. Hardening of Mojaloop releases
2. Security testing of Mojaloop
  - Assess feedback from testing done by security experts, internally
  - Prioritize issues based on criticality and severity
  - Update backlog with stories to address issues
3. Review core and related repositories released with Mojaloop for
  - Ensure dependabot alerts are addressed
  - Ensure Snyk issues are addressed
4. Review the codebase for quality issues
5. Review PRs in critical repositories and address (close, approve / reject)
6. Update maintenance tooling (license-scanner, CI tools)
7. GitHub maintenance of Mojaloop repositories (used in the platform)
8. Hardening Mojaloop core and addressing any known and reported quality, security issues
9. SBOM of dependencies in Mojaloop to better manage situations
10. Provenance of Mojaloop artefacts - (*support for helm charts in v17.1.0*)



# Thank you!

Resources:

1. Mojaloop docs: <https://docs.mojaloop.io>
2. Mojaloop training program: <https://learn.mojaloop.io/>
3. Mojaloop core & releases slack channel (pi28): <https://mojaloop.slack.com/archives/C06VB2FHUER>
4. Mojaloop platform quality & security slack channel (pi28): <https://mojaloop.slack.com/archives/C06UW0E2KBN>
5. Appendix following this

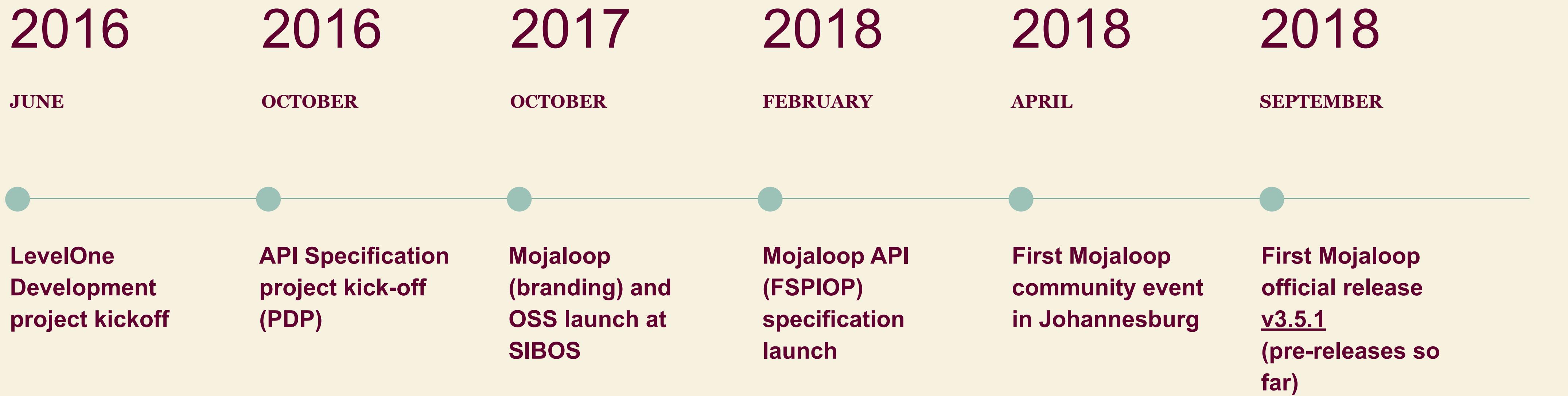
# How we work



1. Messaging: [Short-term - Slack](#); [Community central](#) - announcements
2. GitHub: [Issues](#) for longer form discussion; Issue tracking, Project
3. Documentation
  - a. Markdown on GitHub for specifications and documents with long life.
  - b. Sequence diagrams: Plant UML, DrawIO, etc...
4. Our [Git process](#) (forking/branching WF), the CLA
5. Quality gateways
6. Apache 2.0 License for code
7. API specifications: [docs.mojaloop.io/api](https://docs.mojaloop.io/api), Change requests, Solution Proposals
8. Scaled Agile Framework (SAFe)

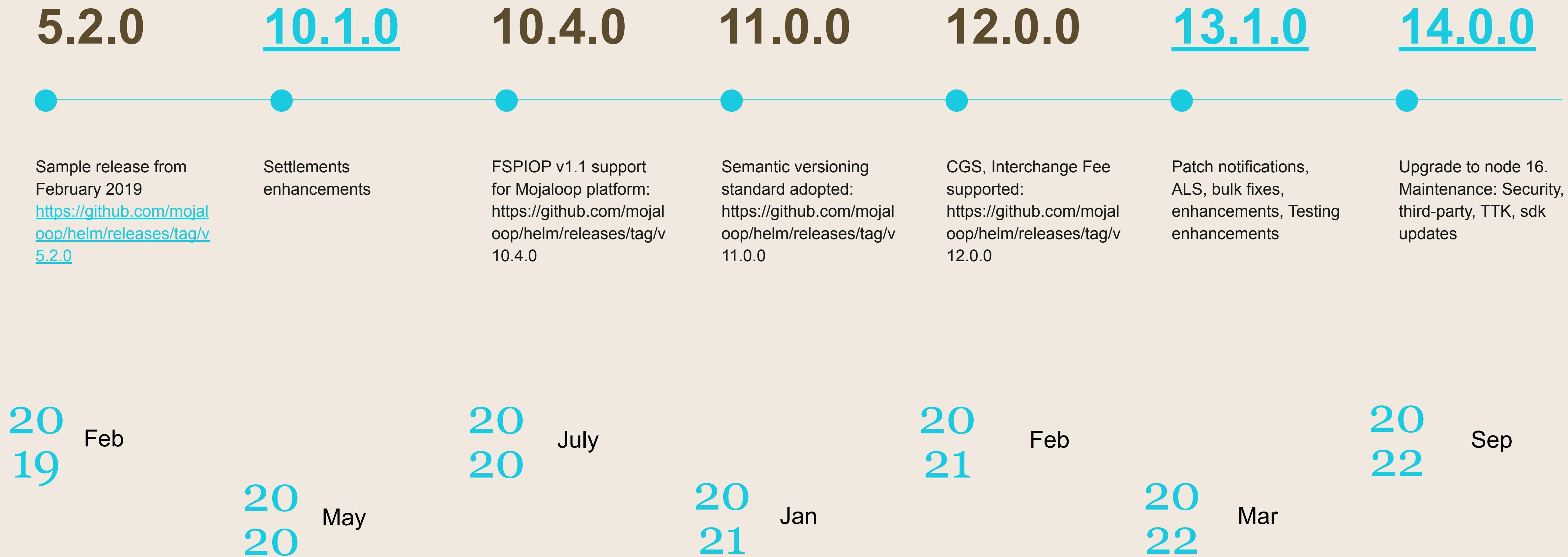
# Mojaloop

## PHASES 1-2



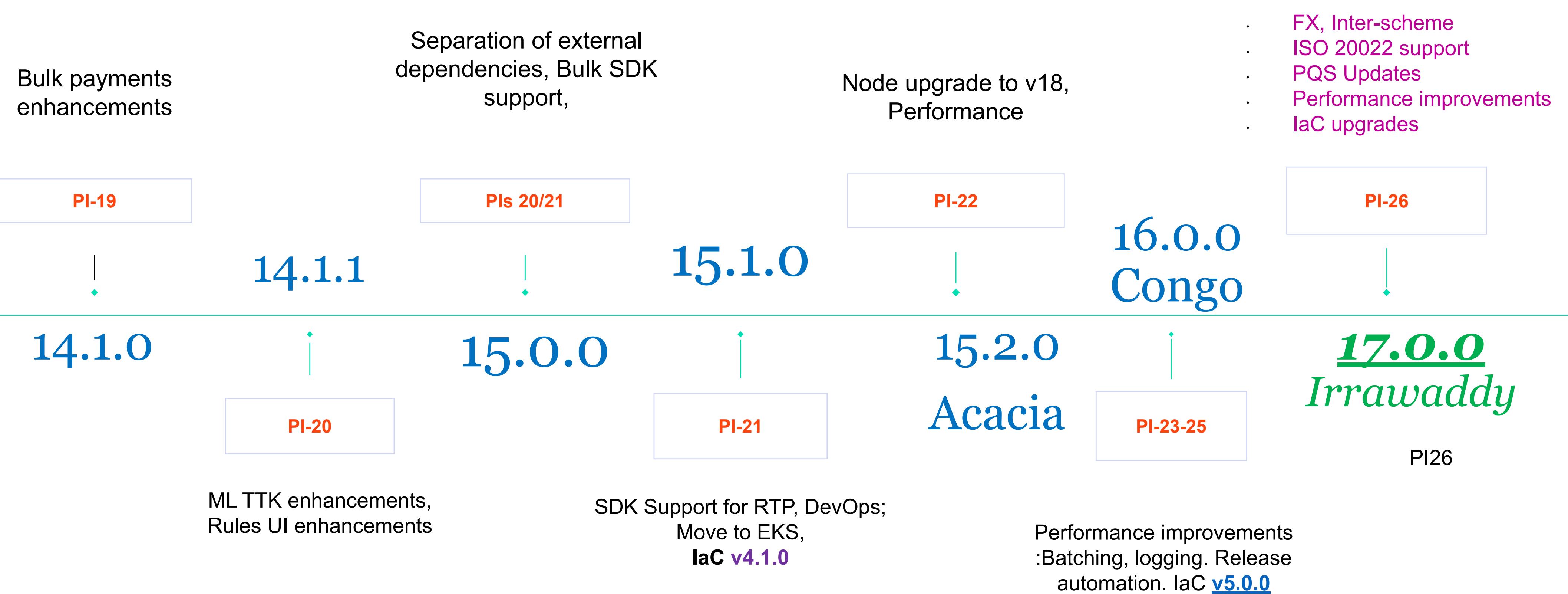
# Mojaloop

PHASES 3-6



# Mojaloop

PI-25

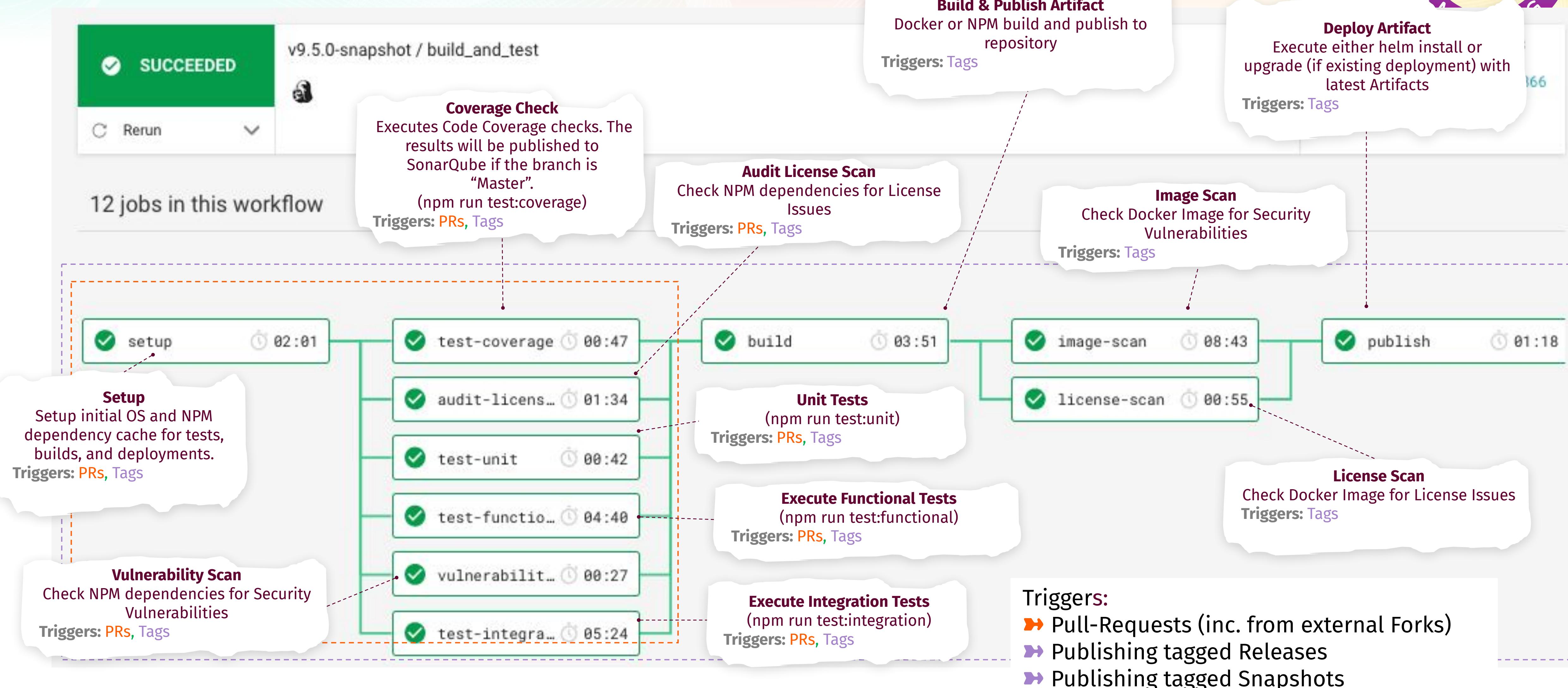


\* Version numbers guided by SemVer

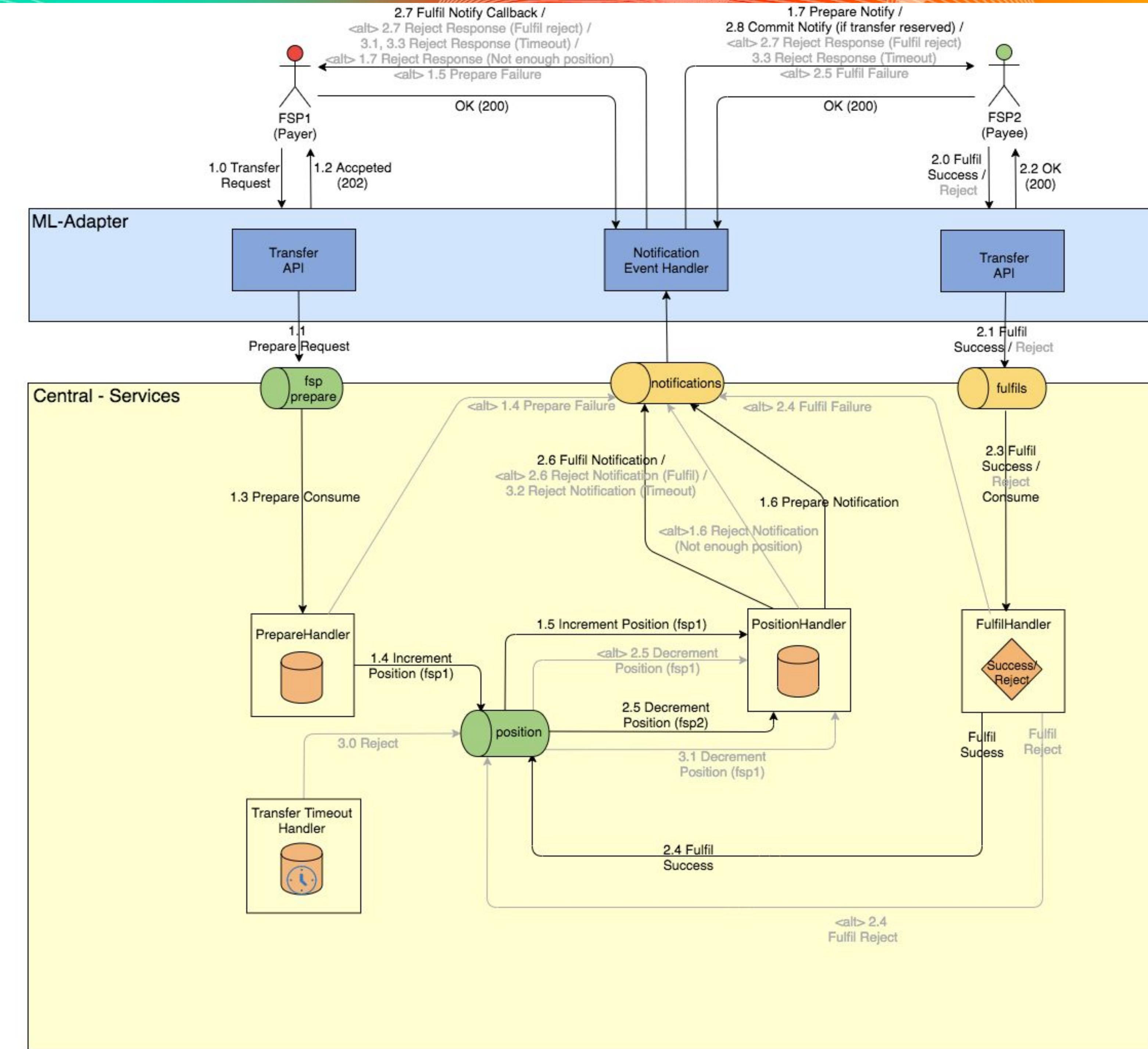
# Deployment Architecture: CI/CD Pipeline -



Ref <https://app.circleci.com/projects/project-dashboard/github/mojaloop>



# Mojaloop Transfers Architecture v1.1



# Release Automation: validation, usage



1. Automated Update of Helm Chart Packages with [updatecli](#)
2. Detection of Breaking Changes - DB Migrations, Default Configuration Changes
3. Generation of Changelog
4. Determination of Next Release Version Number
5. Generation of Release Note
6. Deployment of RC to OSS Test Environment
7. Release Candidate Validation with TTK
8. Creation of Release Candidate Pull Request
9. Bug Fixes, PR Reviews, Github Release

*The automation scripts can be found here:*

<https://github.com/mojaloop/helm/tree/main/.github/workflows>

v16.0.0 - Congo Latest

oderayi released this 2 weeks ago ↳ v16.0.0 ↳ e34bcf5

---

### Helm Release Notes

Date	Revision	Description
2024-02-06	0	Initial draft
2024-03-13	0	Initial release

---

#### 0. Summary

Enhancements and breaking changes to the [v15.2.0 Release](#), which includes:

1. Performance improvement in Central Ledger via batching of position prepare and position fulfil messages
2. Performance improvement in ALS with the addition of participant requests caching, and logging fixes
3. Refactoring of Quoting Service into an event-driven service to improve performance
4. Helm chart: external K8s secret are now used for JWS signing key source if present
5. General maintenance, bug fixes, and Nodejs upgrades

---

#### 1. New Features

- mojaloop/#3426 add participant req caching, enable cache metrics, log fixes ([mojaloop/#465](#)), closes [mojaloop/#3426](#)
- mojaloop/#3427 add oracle endpoint db caching and oracle request caching ([mojaloop/#467](#)), closes [mojaloop/#3427](#)
- mojaloop/#3441 nodejs upgrade ([mojaloop/#252](#)), closes [mojaloop/#3441](#)
- mojaloop/#3636 batching implementation for position prepare messages ([mojaloop/#968](#)), closes [mojaloop/#3636](#)
- mojaloop/#3524 add position fulfil to binprocessor ([mojaloop/#990](#)), closes [mojaloop/#3524](#)
- mojaloop/#3524 add reserve action to fulfil logic ([mojaloop/#992](#)), closes [mojaloop/#3524](#)
- mojaloop/#3666 refactor quoting service into an event driven solution to improve performance ([mojaloop/#321](#)), closes [mojaloop/#3666](#)