

Onboarding FinTechs and External Financial Institutions to Participate in a Mojaloop Payment Network.

Introduction:

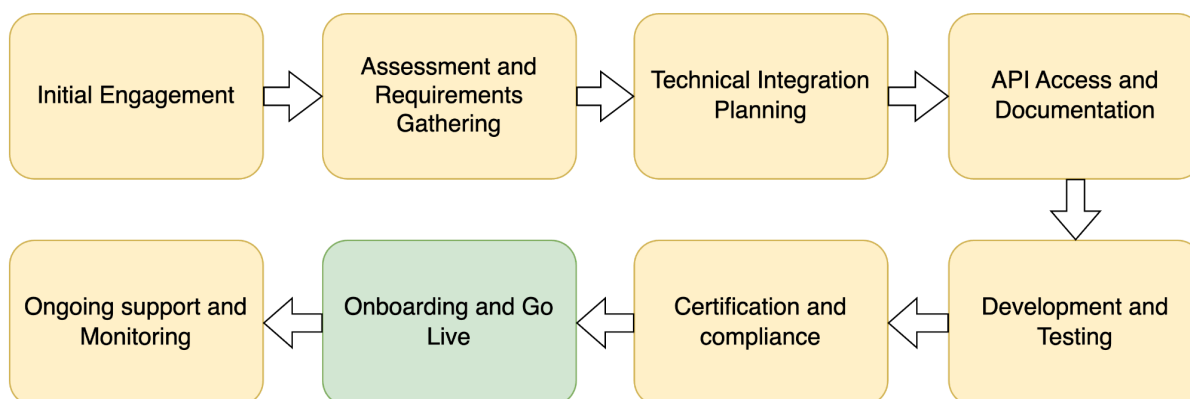
This document outlines the process for FinTechs (financial technology companies) and External Financial Institutions (FIs) to onboard and integrate with a Mojaloop Payment Network. INFITX follows a tried and tested standardised approach to building integrations into FinTech financial backend systems. This approach leverages participation tools that simplify integration using standard design patterns. These tools and patterns provide a best-in-class onboarding and participation experience for your connected FIs and FinTechs.

Onboarding requires less than ten days of effort and can be completed within four weeks. The integration solutions, designed to be Mojaloop enabled, will result in minimal ongoing reconciliation costs.

The proposed lifecycle for onboarding and integration will be tailored to each Mojaloop Payment Network, ensuring compliance with scheme rules, local regulations, and internal compliance requirements necessary for DFSPs to connect.

FinTech Integration Lifecycle:

The onboarding process is divided into distinct stages, each crucial for establishing a secure and functional connection:



Initial Engagement

The process begins with initial discussions between the Payment Network team and interested FinTechs/FIs to explore Mojaloop's capabilities. During these meetings, presentations and detailed information about the platform's features, benefits, and functionalities are shared. This allows Financial Institutions or FinTechs to understand and select the appropriate offerings.

Mojaloop Scheme rules and operating guidelines are discussed, and participating organisations must agree to these terms to operate within the Scheme Rules.

Assessment and Requirements Gathering

A thorough assessment is conducted to understand the specific needs of the FinTech/FI, covering:

- Technical capabilities and training requirements, such as scheme training and tool training.
- Integration requirements and necessary use cases.
- Systems overview and deployment architecture choices, like self-hosted vs. scheme-hosted options.
- Level of participation tools use that is deemed appropriate.

Information about APIs, security protocols, and compliance standards is also gathered during this phase.

Technical Integration Planning

The FinTechs/FIs and their System Integrators (SI)/Vendors must choose an integration approach and develop an implementation plan that outlines the following:

- Steps involved in integration.
- Timelines for completion.
- Required resources and technical specifications for compatibility, including scheme participant type, use cases, core banking resources, and selected integration flow patterns.
- Data Exchange Protocols, API endpoints, and system compatibility considerations.

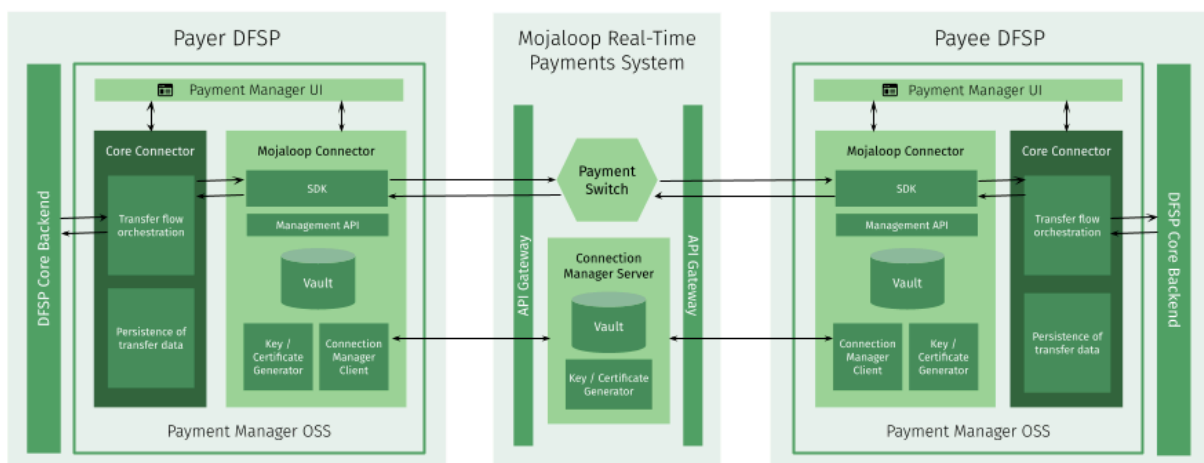
All stakeholders must agree that the design meets the business requirements.

API Access and Documentation

The self-help Developer Portal provides access to Mojaloop platform APIs and comprehensive documentation. The documentation details API endpoints, functionalities, and relevant data formats. Tools like the Mojaloop Testing Toolkit and local deployment of Payment Manager are offered for exploration and testing. These resources enable the simulation of financial and API flows, with tutorials and dedicated support channels to assist in understanding APIs and tools.

Development and Testing

FinTechs/FIs develop their integrations using the provided tools and mock responses. The Developer Portal facilitates development with resources such as code samples, Software Development Kits (SDKs), and developer forums for collaboration. A testing sandbox environment is provided to simulate real operations with features like example, payment managers, test currency transactions, and comprehensive test scenarios.



Certification and Compliance

FinTechs/FIs validate their integration's functionality, security, and compliance with Mojaloop standards, performing necessary audits before going live.

Onboarding and Go Live

FinTechs/FIs are onboarded to the Mojaloop Payment Network upon successful testing and certification. Their integrations are deployed to the production environment, enabling live payments. The go-live process involves participant creation, new payment manager deployment, and establishing secure communication channels, with a thorough checklist to ensure a smooth transition.

Ongoing Support and Monitoring

Mojaloop and the participation tools are designed to ease participation efforts and ongoing operational costs. Ongoing support includes troubleshooting, performance optimisation, and updates to meet evolving business needs. FinTechs/FIs also receive access to payment manager portals for credential management, Role-Based Access Control (RBAC), and transfer history monitoring, with robust security measures like multi-factor authentication to safeguard sensitive data. Comprehensive documentation and training materials guide portal usage, ensuring smooth operation.

Addendum A: Technical Collaborator Skills

Skill Set	Description
Node.js and TypeScript:	The project involves building a microservice using Node.js and TypeScript. Proficiency in both of these technologies is essential for developing robust and scalable server-side applications.
RESTful API Design:	Understanding and implementing API design principles is crucial for building the custom component that connects with the core banking systems and other standard integration components. This includes designing endpoints, handling HTTP methods, and ensuring proper data exchange formats like JSON.
Microservices Architecture:	Familiarity with microservices architecture is important for designing the system in a modular and scalable way. This involves breaking down the application into smaller, loosely coupled services that can be independently developed, deployed, and maintained.
Financial Domain Knowledge	Understanding financial concepts, such as transactions, accounts, balances, and authentication mechanisms like OAuth2, is essential for building integrations with core banking systems. Knowledge of financial regulations and compliance standards may also be necessary, depending on the specific requirements.
Security Best Practices	Given the sensitive nature of financial data, implementing security best practices is critical.
Error Handling and Logging	Implementing robust error handling and logging mechanisms is essential for diagnosing issues and ensuring the reliability of the system. This includes handling exceptions gracefully, logging relevant information for troubleshooting, and monitoring system health.
Testing and Quality Assurance	Writing comprehensive unit tests, integration tests, and end-to-end tests is crucial for ensuring the reliability and correctness of the software. Knowledge of testing frameworks like Jest or TTK, as well as techniques such as mocking and stubbing, is valuable for this purpose.
Continuous Integration/Continuous Deployment (CI/CD)	Implementing CI/CD pipelines automates the process of building, testing, and deploying code changes, leading to faster release cycles and improved software quality. Familiarity with tools like CircleCI, or GitHub Actions is beneficial for setting up these pipelines.
Version Control	Proficiency in version control systems like Git is essential for collaborating with team members, managing code changes, and tracking the history of the project.

Addendum B: Technical Skill Summary by Role

Skill Set	Description
System Integration Engineer	Sequence Diagrams; Microservices; Programing Languages Typescript; NodeJS; API Design OpenAPI; Infrastructure as Code Terraform; Containerised deployment Helm; Docker; Kubernetes; Security PKI; CI/CD GitHub; CI; Linux;
QA Engineer	HTTP API testing; Postman; Test Automation Frameworks, Jest; TestCafe, C, GitActions/ CircleCI; Docker; Text Design and documentation; Web technologies

Addendum C: Mojaloop Training Program Course Recommendations

The [Mojaloop Training Program](#) is a useful resource for building technical skills on Mojaloop. The following courses are recommended for the following roles:

System Integrator Business / Product Expert

- The [MOJA101](#) and [DFSP101](#) Mojaloop Training Courses should be taken.
- Optional Courses that would help understanding, but are not required immediately are: [SCHEME101](#)

System Integrator Technical Architect

- The [MOJA101](#), [MOJA102](#), [DFSP101](#) and [TTK101](#) Mojaloop Training Courses should be taken.
- Optional Courses that would help understanding, but are not required immediately are: [MOJA104](#), [DFSP209](#), [MOJA103](#)

System Integrator Software Engineer

- The [MOJA101](#), [MOJA102](#), and [TTK101](#) Mojaloop Training Courses should be taken.

Addendum C: Delivery Stories

Story	Description
Execute DFSP Technical and Business Assessment for Connecting to Mojaloop	<p>As a Digital Financial Services Provider (DFSP) Integration Manager</p> <p>I want to perform a comprehensive technical and business assessment for connecting our DFSP to Mojaloop</p> <p>So that I can ensure seamless integration, compliance with technical requirements, and alignment with business objectives.</p>
Design of the Financial Mojaloop Core Connection via Payment Manager	<p>As a Payment System Architect</p> <p>I want to design the core connection between the Financial System and Mojaloop via the Payment Manager.</p> <p>So that transactions can be processed seamlessly and efficiently, ensuring secure and reliable payment flows.</p>
Development of a Financial Core Connector for Mojaloop Payment Manager	<p>As a Digital Financial Services Provider (DFSP)</p> <p>I want to be able to use the "PM4ML/SDK Core Connector."</p> <p>So that our financial system can process transactions seamlessly with the Mojaloop network, ensuring secure, efficient, and reliable payment flows.</p>
Onboard DFSP to UAT Testing	<p>As a Mojaloop Implementation Manager</p> <p>I want to onboard a Digital Financial Services Provider (DFSP) to the User Acceptance Testing (UAT) phase of Mojaloop integration.</p> <p>So that the DFSP can validate the integration in a controlled environment before going live.</p>
Design of the FRMS Connection via Payment Manager (Optional)	<p>As a Payment System Architect</p> <p>I want to design the connection between the Financial Risk Management System (FRMS) and the Payment Manager.</p> <p>So that transactions can be monitored and analyzed for fraud risk in real-time, ensuring secure and reliable payment flows.</p>
Development of a Tazama FRMS Connector for Mojaloop Payment Manager (Optional)	<p>As a Payments System Integrator</p> <p>I want to develop a connector for the Tazama Fraud Risk Management System (FRMS) that integrates with the Mojaloop Payment Manager.</p> <p>So that I can detect and mitigate fraudulent transactions in real time within the Mojaloop payment ecosystem.</p>

Addendum C: References

1. Mojaloop Business Operations Documentation - Technical onboarding of DFSPs
<https://docs.mojaloop.io/mojaloop-business-docs/HubOperations/Onboarding/technical-onboarding.html>
2. infitx.com - The Journey to Become a Mojaloop System Integrator
<https://www.infitx.com/the-journey-to-become-a-mojaloop-system-integrator/>
3. infitx.com - Developing a Core Connector for Mifos X
<https://www.infitx.com/developing-a-core-connector-for-mifos-x/>
4. Payment Manager OSS Documentation
https://pm4ml.github.io/documents/payment_manager_oss/latest/index.html
5. GitHub - Payment Manager for Mojaloop
<https://github.com/pm4ml>