

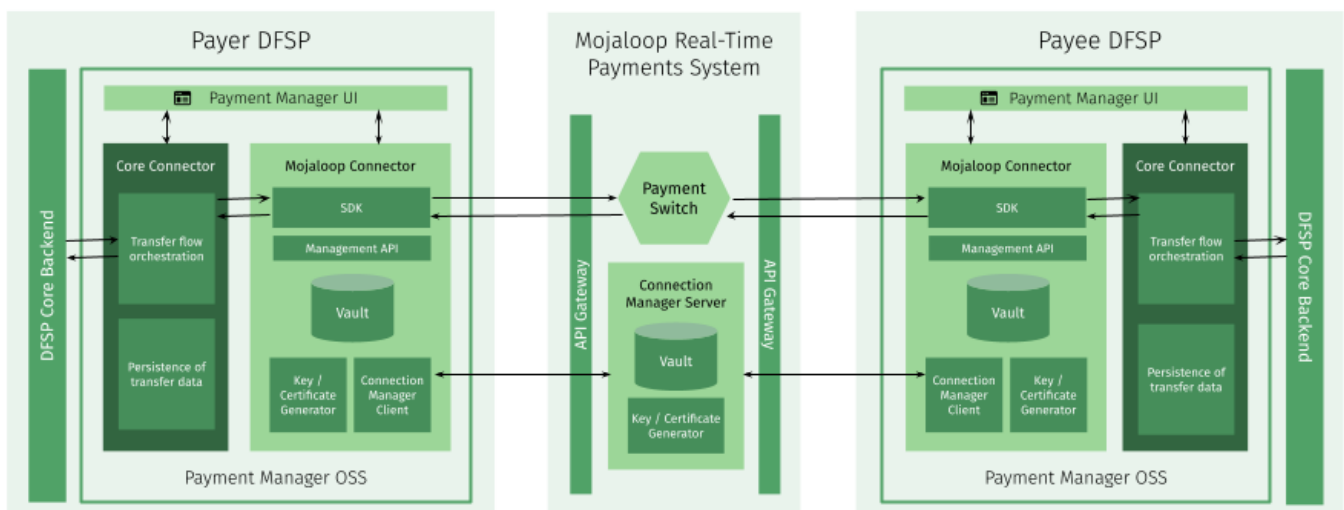
Payment Manager OSS

Building, validating, and operating a service that connects a Digital Financial Service Provider's (DFSP's) core systems to a Mojaloop real-time payments network requires a detailed understanding of the business, technical, and operational requirements of that network. Payment Manager makes the integration path to Mojaloop easier.

Payment Manager provides tools to manage connections, services, and transaction monitoring, and it also provides a single point to configure the DFSP environment to go from a test to a real-time one.

Payment Manager is made up of the following major components:

- **Core Connector:** Integrates a DFSP's Core Backend to Payment Manager as an "adapter" for both parts so communication is possible between them. It uses standard templates, the majority of them written in Apache Camel, a declarative language for integration engineers that does not require writing code from scratch.
- **Mojaloop Connector:** Comes with the following key components:
 - A Mojaloop-SDK, which provides Mojaloop-compliant security components and HTTP header processing capabilities, as well as a simplified version of the Mojaloop API.
 - A Connection Manager Client, which simplifies and automates certificate creation, signing and exchange, as well as the configuration of the connections required to different environments.
- **Business and Technical Portals:** Provide dashboards for monitoring transactions and service status. They also allow DFSPs to manage their security keys, certificates and endpoint configuration required for connecting to Mojaloop-based schemes.



Payment Manager abstracts away the complexity of the Mojaloop API and helps the DFSP understand their relationship with the scheme, focusing on the key business and technical areas.

Payment Manager can reduce technical risk, as well as ongoing operational capability needs, thereby reducing costs for development and integration with a Mojaloop Hub.

A few words about Payment Manager technology:

- Payment Manager is provided as a set of Linux container images (docker) and may be hosted on-premise using commodity server infrastructure or in appropriate cloud infrastructure where available.
- Payment Manager scales from a very small footprint to handling many hundreds of transactions per second (dependent on the performance of the DFSP's Core Backend).
- Payment Manager comes with turn-key integration options.

[Here](#) is a link to get more information.

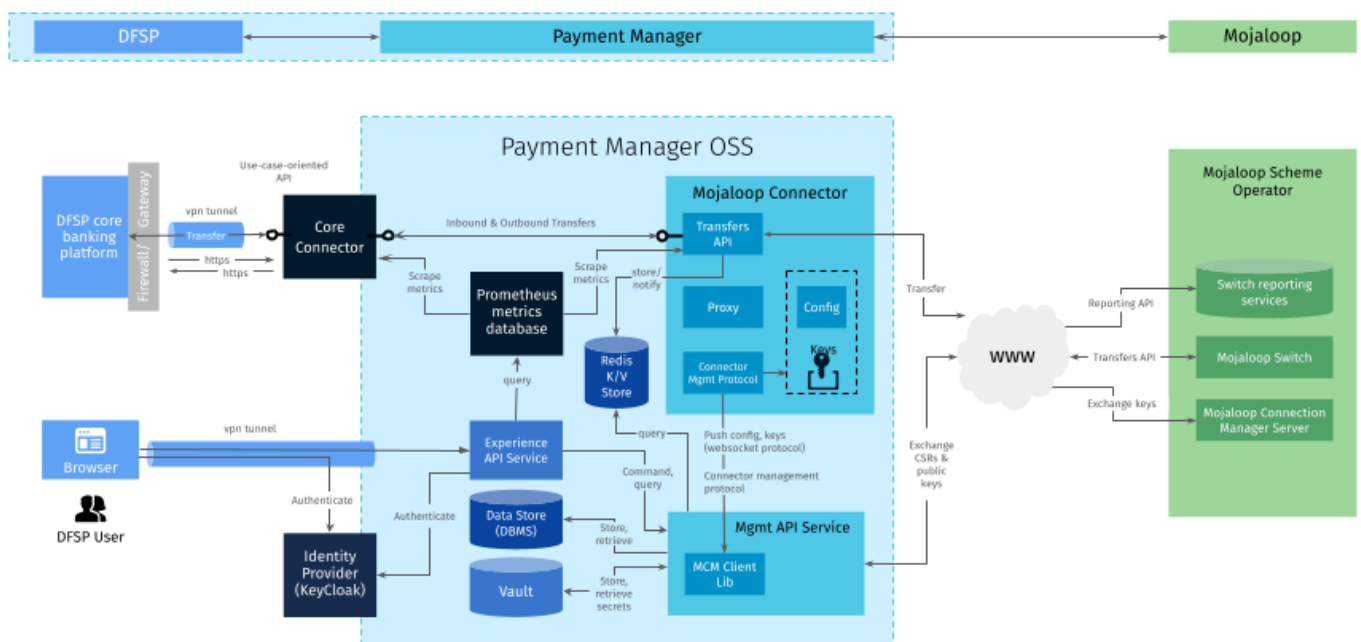
Technical Introduction

At a high level, onboarding to a Mojaloop-based scheme requires a Digital Financial Service Provider (DFSP) to focus their efforts around the following major milestones:

- Integration of their Core Backend with the Mojaloop Switch on the API level (this involves both coding and testing).
- Connecting to Mojaloop pre-production and production environments following rigorous Mojaloop security requirements.

Payment Manager for Mojaloop provides functionality to simplify both of these steps. This document provides details about API-level integration.

The following diagram provides a high-level view of the integration between a Mojaloop Real-Time Payment System and a DFSP's Core Backend.



Payment Manager provides the following components:

- **Core Connector:** Integrates a DFSP's Core Backend to Payment Manager as an "adapter" for both parts so communication is possible between them. It uses standard templates, the majority of them written in Apache Camel, a declarative language for integration engineers that does not require writing code from scratch. There is a ready-made Core Connector template available for DFSPs to simplify their development effort.
- **Mojaloop Connector:** Comes with the following key components:

- A Mojaloop-SDK, which provides:
 - Mojaloop-compliant security components
 - HTTP header processing capabilities
 - A simplified use-case-oriented version of the Mojaloop FSPIOP API. DFSPs will be talking to this API, leveraging Core Connector.
- A Connection Manager Client, which simplifies and automates certificate creation, signing and exchange, as well as the configuration of the connections required to different environments.
- **Portals:** Provide dashboards for monitoring transactions and service status. They also allow DFSPs to manage their security keys, certificates, and endpoint configuration required for connecting to a Mojaloop-based scheme in a guided way.