OpenLegacy
Digital to the Core

Datasheet

OpenLegacy IMS DC Connector

Overview

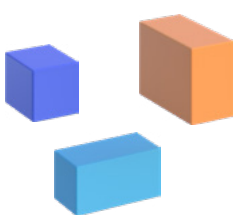
The OpenLegacy IMS DC Connector generates microservice-based APIs to deliver and extend legacy IMS DC functionality through the OpenLegacy platform. Using the OpenLegacy IMS DC Connector does not require any changes to existing applications or the installation of any additional software.

Key benefits

- Enables access to all legacy IMS DC transactions using technology supported by IBM
- Supports transaction-specific data models so transaction data can be easily retrieved
- Automated IDE and use of a standard Java stack saves time, effort and money
- Code-first approach allows source control and versioning, CI/CD pipelines, and the use of standardized development methodologies
- Convenient sandbox testing using Junit and Swagger
- Complies with security standards

Features

- Automatic generation of Java data structures from IMS DC Transaction copybooks
- Direct access to IMS transactions from APIs and selfcontained microservices
- Multiple simultaneous IMS DC transactions handled through dependency-based orchestration
- Wizard and template driven connector definition and configuration using Spring Boot framework
- Generation of connector based on pre-defined structures, with IMS DC-specific configurations



IMS DC connector architecture

SDK

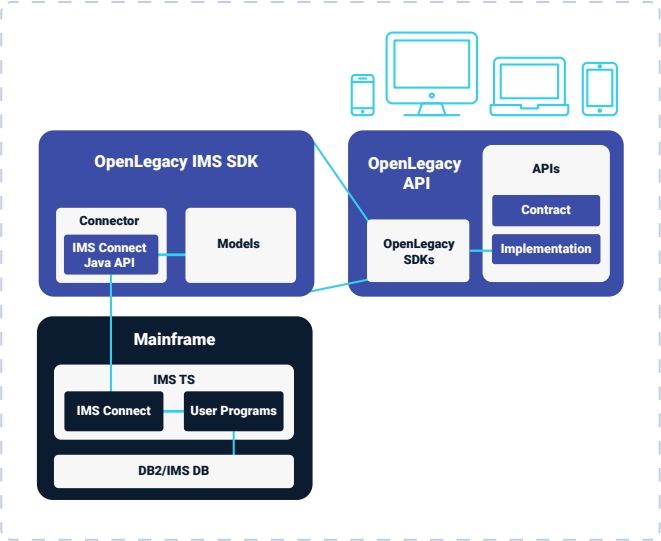
- Models are automatically generated from source copybooks
- Models represent transactions
- Models contain I/O fields that reflect transaction I/O fields
- Connector receives a model with input parameters, invokes the transaction and populates the model output fields with the response
- Connector is based on IBM's standard IMS Connect library

Service

- Exposes a Java service and allows customization beyond the IMS fields' inputs and outputs
- Utilizes the SDK to implement customized service
- Java Implementation can be customized to apply new business logic and cross-cutting concerns

Contract

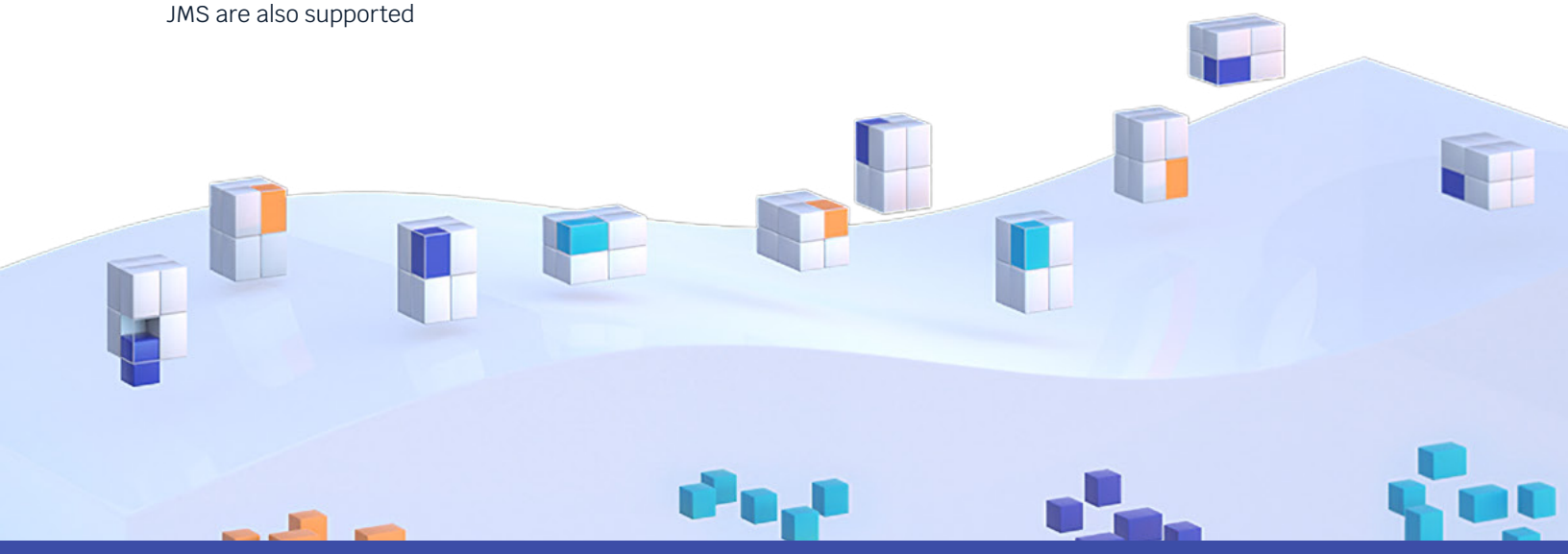
- Standard (REST) contracts are exposed using the Swagger API catalog
- Additional endpoints, such as SOAP, RabbitMQ and JMS are also supported



OpenLegacy IMS DC Connector Technical Details

Technical Prerequisites

- IMS installed on mainframe, configured to use one of the default user message exit routines (HWSSMPL0, HWSSMPL1)
- Full IDE installation on development machine
- The OpenLegacy IMS DC Connector has been tested and officially verified on IMS v12 and v14



Connector Configuration

In addition to common Mainframe RPC project properties, there are also configuration properties specific to IMS DC, such as:

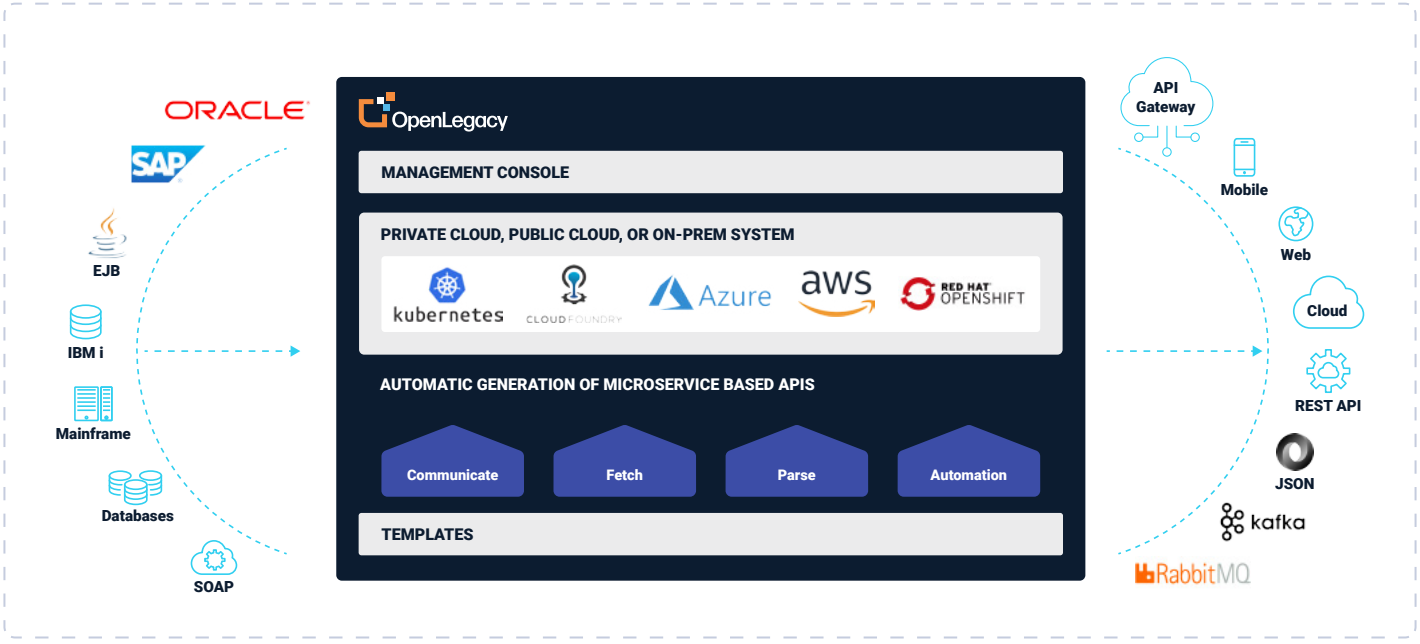
- **host:** IP address of the IMS host
- **port:** the port used by the IMS host
- **username:** RACF username (if RACF is used)
- **codepage:** The code page for translating buffers
- **data-store-name:** The name of the IMS host data store
- **connection-timeout:** connection has timed out
- **transaction-timeout:** transaction has timed out
- **password:** RACF password (if RACF is used)
- **group-name:** RACF group (if RACF is used)

SDK Usage

OpenLegacy uses the IBM IMS Connect API to connect to IMS mainframe and invoke IMS transactions. Like all other OpenLegacy Java APIs, the IMS SDK uses the RpcSession interface to execute a program represented by an RpcEntity.

OpenLegacy Microservice Architecture

The OpenLegacy platform serves as a microservices enabler. It allows easy decoupling of existing monolithic applications into small independent microservices. The OpenLegacy integration process creates each API integrated with the monolith in a separate Java project. By default, the OpenLegacy Java Project template contains all of the complex configurations needed to allow each API project to be deployed separately, to communicate with the various microservices ecosystem components, to scale out easily and behave as a complete microservice.



Automatically and quickly create microservice APIs

About OpenLegacy

OpenLegacy's Digital-Driven Integration enables organizations with legacy systems to release new digital services faster and more efficiently than ever before. It connects directly to even the most complex legacy systems, bypassing the need for extra layers of technology. It then automatically generates APIs in minutes, rapidly integrating those assets into exciting new innovations. Finally, it deploys them as standard microservices or serverless functions, giving organizations speed and flexibility while drastically cutting costs and resources. With OpenLegacy, industry-leading companies release new apps, features, and updates in days instead of months, enabling them to truly become digital to the core.