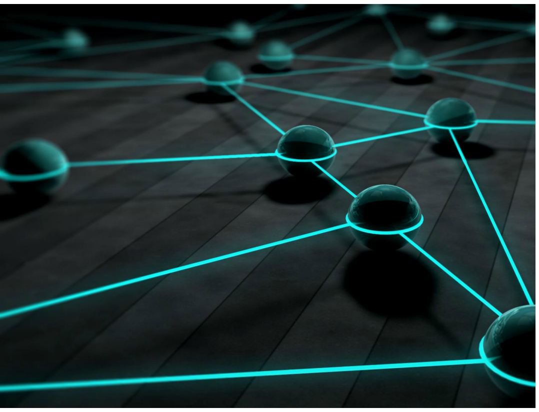
Mifos X Software Platform

Technical Overview



SACCOs, Co-Operatives, Banks,
Credit Unions, Digital Platforms

Mifos®
Technology for Financial Inclusion

Introduction

Mifos X Software Platform (simply "the Platform") is built around a multitenant, service oriented, and tiered architecture, and can be deployed in a SaaS[1] (Software as a Service) model or on-premises.

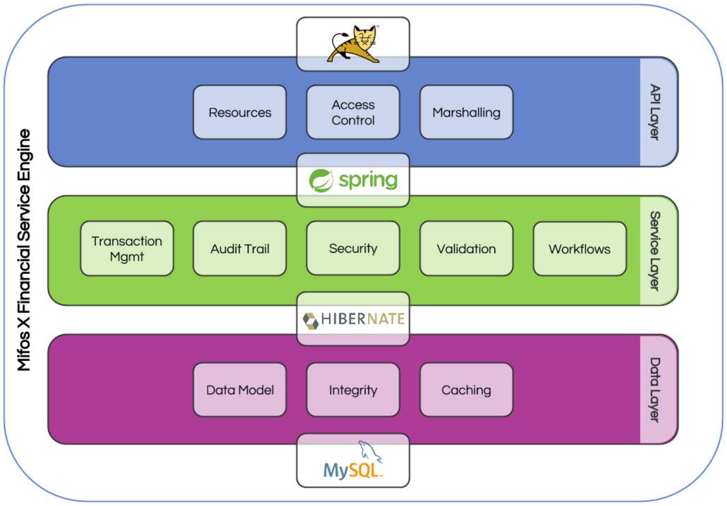
The foundation forms a robust but flexible data model which is ready-made for extensions and customizations. An API provides access to all basic functions grouped in modules.

Because the Platform needs to be accessible to users at remote locations, a browser-based solution was essential.

Java was selected as a suitable high level language allowing the rapid customization of code required for each individual circumstance yet has a large number of powerful libraries available.

Mifos X Financial Service Engine

The Mifos X Financial Service Engine (simply "the Engine") is a scalable and extensible framework written in Java, and licensed under the Mozilla Public License 2.0[2].



The main design principle of the Engine is CQRS[3] (Command Query Respon- sibility Segregation), a pattern that separates commands and queries into different models and services. This approach has multiple benefits: 1) State changes (commands) are persisted, providing an audit of all changes, 2) fine grained control and extension of state changes, and 3) scalability based on consumer behavior and real system load.

Based on a tiered architecture the Engine provides multiple layers to separate concerns and allow reusability.

API LAYER

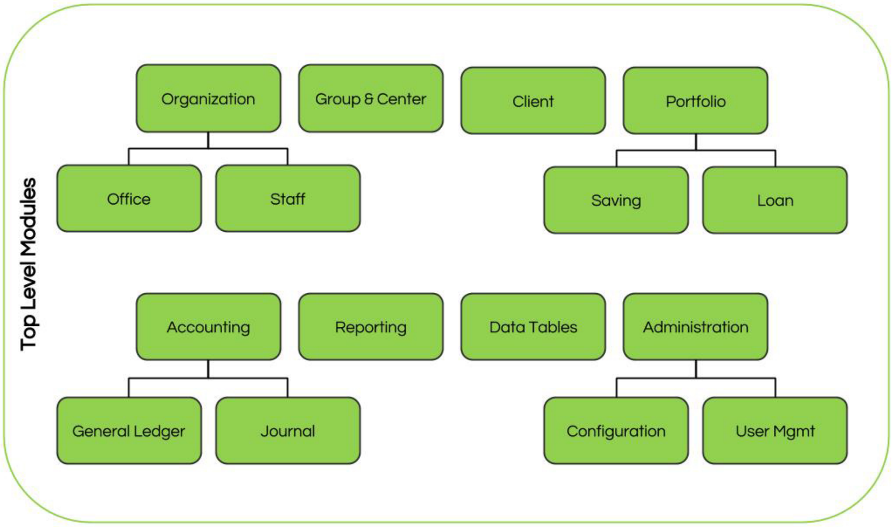
The API Layer is built entirely as a RESTful[4] Webservice, using JSON[5] to transmit data, and utilizes standard HTTP Methods for interactions. The main components of this layer are:

- Resources: Every module is exposed via an URI[5] (Unique Resource Identifier).
- Access Control: Early exit if a consumer lacks authorization for a queried Resource.
- Data marshaling: Any data transmitted to and from the API is de-/serialized using JSON.

The API is exposed via HTTPS to encrypt all data and secure the communication. OAuth2 is used for authorization.

Stateless by nature the API Layer can be scaled based on the actual load and even used in high availability scenarios.

SERVICE LAYER



The Service Layer provides module specific business logic and rules, role based access control[7] and a xBRL[8] compliant reporting engine. Transaction awareness and data validity is encapsulated, and extension points are available to enhance built-in workflows.

Cross concerns like Security or Transactions are part of the core framework and provided via AOP[9] (Aspect Oriented Programming).

The Service Layer has horizontal and vertical layers.

Services, command, and event handlers form horizontal layers; functional modules form vertical layers.

The top level modules are shown in the diagram.

Data Layer

The Data Layer provides module specific access to data stores. Data integrity is reached by using the ACID[10] principal and well defined data relations. Write and read operations are encapsulated in different repositories to define a clean responsibility separation.

The Data Layer uses two cache strategies to cache data efficiently; database and 2nd level caching.

- Database caching stores records on the database level and is useful for reporting, batch jobs, and low level SQL queries.
- 2nd Level caching stores data objects on the application level and keeps database turnarounds on a very low level allowing fast in-memory access of frequent data.

ARCHITECTURE

The Engine's architecture was designed with a focus on modern, scalable and extensible technologies ready to run in the cloud. All components where selected based on their maturity, proven reliability and cost efficiency.

It is our goal to provide a software that is deployable with effectively no cost that can grow with the customer need over time using well known best practices and support from a wide range of communities.

The basic Mifos X Financial Service Engine architecture is as follows:

Resource	Component	Comment
Operating System	Linux , Windows, Mac OS	Amazon AWS image and Debian package available
Runtime	Java 7.x	
Application Server	Tomcat 7.x	Embedded Tomcat for development environments
Database	MySQL 5.x	Embedded MariaDB for development environments
Application Framework	Spring Framework 4.x	
Persistence Framework	Hibernate 4.x	
Reporting Engine	Pentaho Reporting Engine 3.x	
Test Framework	JUnit 4.x Mockito 1.x REST-assured 2.x	

Mifos X Community App

Mifos X Community App (simply "the Community App") is a browser based Single Page Application[11] (SPA) built entirely with HTML, JavaScript and CSS. Resources are loaded dynamically and added to the page as necessary.

All user interactions, workflows, and application logic is taking place in the browser and does not create load on the Engine. Ajax[12] is used to send data to and receive data from the Engine asynchronous, and bidirectional model binding update views automatically.

Based on AngularJS the Community App is built on the MVC[13] (Model-View-Controller) pattern and uses templates to render dynamic content. To provide a responsive and modern look and feel Bootstrap is part of the application stack.

The basic Mifos X Community App components are as follows:

Resource	Component
Application Framework	AngularJS 1.x
Dependency Management	Bower 1.x
Module Management	RequireJS 2.x
Chart Library	D3 3.x
Look and Feel	Bootstrap 3.x
Test Framework	Jasmine 1.x

References

1. http://en.wikipedia.org/wiki/Software_as_a_service
2. https://www.mozilla.org/MPL/2.0/
3. http://martinfowler.com/bliki/CQRS.html
4. http://en.wikipedia.org/wiki/Representational_state_transfer
5. http://json.org/
6. http://en.wikipedia.org/wiki/Uniform_resource_identifier
7. http://en.wikipedia.org/wiki/Role-based_access_control
8. https://www.xbrl.org/
9. http://en.wikipedia.org/wiki/Aspect-oriented_programming
10. http://en.wikipedia.org/wiki/ACID
11. http://en.wikipedia.org/wiki/Single-page_application
12. http://en.wikipedia.org/wiki/Ajax_(programming)
13. http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller

Disclaimer

Mifos Initiative is approved by the Internal Revenue Service as a 501 (C) (3) tax-exempt organization, and all donations are tax deductible to the extent provided by law. The Mifos Initiative's Federal Identification Number (EIN) is 45-3613178.