6/8/2022

Yogesh Chandra

GIKS INDIA PVT LTD

NBP Liferay Platform Architecture Specification &

Project Plan

Table of Contents

[Overview 2](#_Toc105858360)

[Liferay Core Technologies and Standards 2](#_Toc105858361)

[Liferay platform Architecture 3](#_Toc105858362)

[Liferay User Interface Technologies Overview 4](#_Toc105858363)

[NBP Liferay Platform Architecture (To-BE) 6](#_Toc105858364)

# Overview

The main objective of the document is to specify Architecture of Liferay Platform to develop and deploy National Business Portal (NBP).

# Liferay Core Technologies and Standards

The Liferay platform core build on following technologies

* **JAVA EE:** the runtime and programming environment
* **OSGI:** modular application runtime and development framework
* **Spring:** transaction and dependency injection in the core
* **Hibernate:** for database access and (along with direct JDBC access for optimize queries)
* **Ehcache:** caching
* **Elasticsearch:** indexing and searching

Liferay is compliant with many industry-proven standards. For example, the following standards, are supported:

* Portlets 1.0 (JSR-168) and Portlets 2.0 (JSR-286) Liferay Portal can run any portlets that follow these two portlet specifications
* JSF (JSR-127, JSR-314, JSR-344) The Java standard for building component-based web applications. Liferay is an active contributor to the standard and lead of the JSF-Portlet Bridge specification.
* EcmaScript 2015 (ES6) Liferay's tooling supports the 6th edition of ECMAScript specification and provides the ability to use it in all modern browsers with the integration of the Babel JS transpiler.
* JAX-WS and JAX-RS Java API for XML Web Services (JAX-WS) and for building RESTful services (JAX-RS) are incorporated as the preferred tooling to create web services.

Liferay supports the OSGi family of standards through its own implementations and also integrates implementations of the Apache Felix and Eclipse Equinox projects. Here are some of the most relevant supported standards:

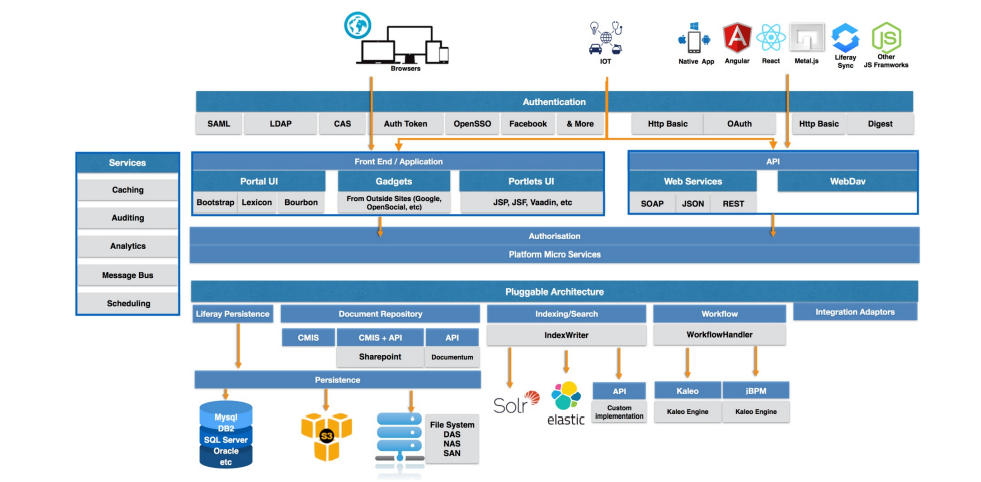
* **OSGI Runtime:** Allowing any module to run in Liferay
* **Declarative services:** Supports a dynamic component model for Liferay Development
* **Configuration Admin:** Lets you create configurable applications that can be reconfiguration on the fly. Liferay provides an auto generated UI to change the configuration of any component that leverages this feature.



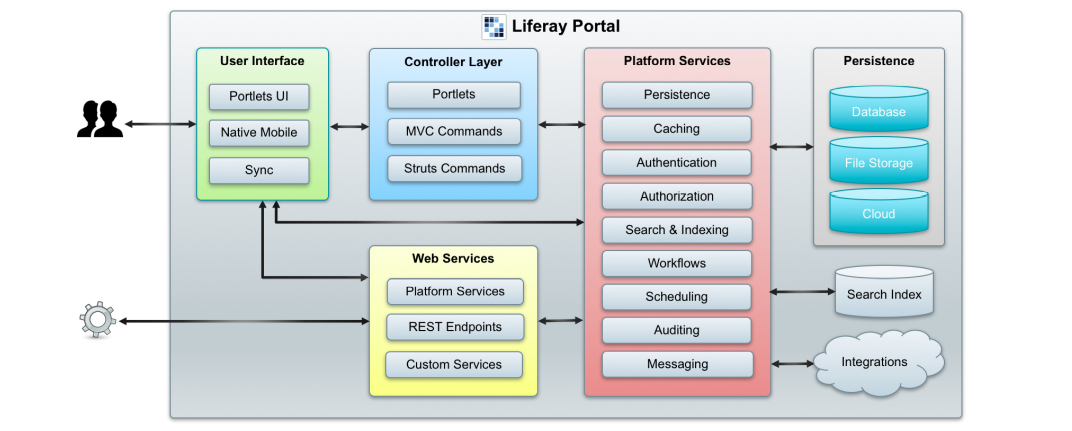
## Liferay platform Architecture

The platform architecture can be presented in different ways. Below are two examples. The first one focuses on the technologies and the second one on the logical architecture

**Liferay Architecture, technology view**



**Liferay architecture, logical view**



# Liferay User Interface Technologies Overview

**Liferay User Interface Types**

Because of the comprehensive web services APIs, the Liferay platform can be accessed from different kind of user interfaces. In addition to the portal web user interface, Liferay provides desktop and mobile Sync clients for accessing the document repository and provides tools like Liferay Screens and Mobile SDK for creating native mobile applications. In the following sections, we will discuss the basic concepts and means of customizing the portal web user interface.

**Web User Interface Technologies Overview**

How is the portal web interface implemented? The web user interface is built mainly on JSP technology. Styling and user interface responsiveness is accomplished with Liferay Lexicon and Clay frameworks. Three main JavaScript frameworks are used as well: Alloy UI, jQuery\_\_ and the emerging Metal.js\_\_. Additionally, the JSP files use Liferay custom tag libraries.

**Liferay Lexicon**

Liferay Lexicon is an abstract user interface design language. It doesn't dictate the implementation but provides principles, patterns, and tools to produce a common design framework and a consistent style and user experience.

**Liferay Clay**

Liferay Clay is an implementation of the Lexicon Experience Design Language. It is an extension of the Twitter Bootstrap framework and is built with HTML, CSS, and JavaScript.

The dependency of Twitter Bootstrap means that jQuery is always available in Liferay by default

**Alloy UI**

AlloyUI is an extensive UI framework incorporating HTML, CSS, and JavaScript and built on Yahoo YUI3. It has over 350 YUI and 150 modules. As Yahoo decided to stop maintaining the YUI, AlloyUI was already deprecated in Liferay DXP. Although legacy, it is still commonly used in the portal native user interface and will be supported throughout the whole Liferay 7.x product lifecycle.

**Metal.js**

The development of Liferay-developed lightweight JavaScript framework Metal.js started after Yahoo YUI deprecation in 2014. Metal.js integrates with Google Closures (SOY) and Facebook (JSX) templating languages, supporting ECMAScript 2015 and 6 (ES 2015, ES6). Metal.js also has an isomorphic, Server-side rendering (SSR) and AMD loader support. Below is an example of a Metal.js component with a SOY template. When compiled and transpiled, they are merged into a single JavaScript file

**Tag Libraries**

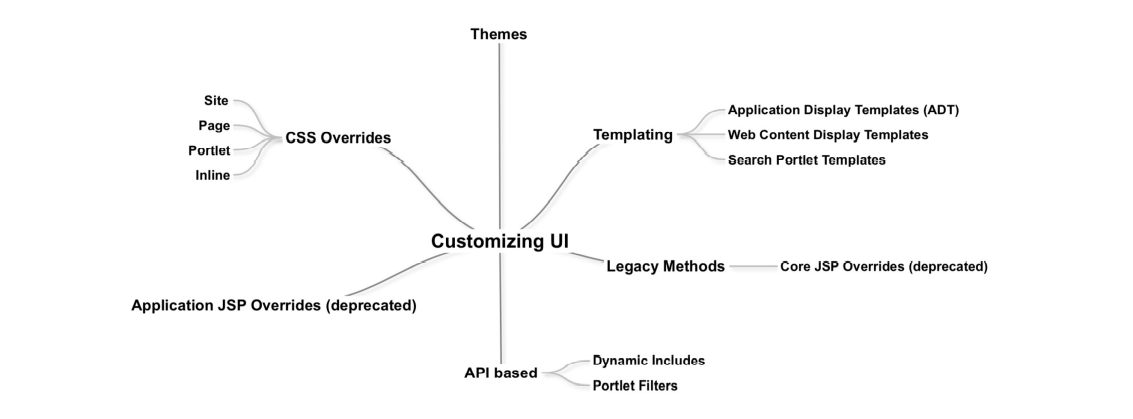
In addition to JavaScript and CSS libraries, Liferay offers a set of fully integrated taglibs for use in JSP files. In addition to reducing boilerplate code, taglibs provide other advantages, too, like a consistent and responsive user interface. The documents for most of the Liferay taglibs can be accessed on [https://docs.liferay.com/ce/portal/7.1- latest/taglibs/util-taglib/](https://docs.liferay.com/ce/portal/7.1-%20latest/taglibs/util-taglib/).

**Summary**

The Liferay portal web user interface still relies on Alloy UI, but Metal.js is increasingly being introduced in the new native applications.

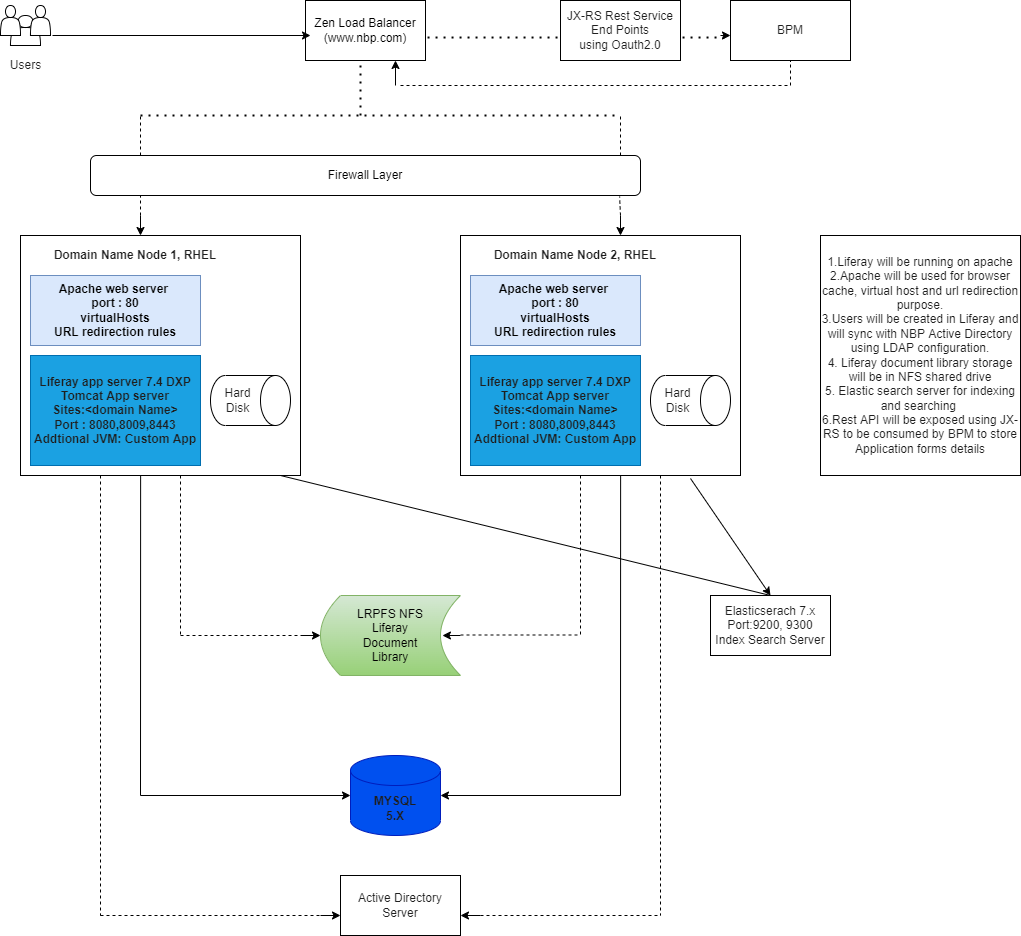
As a development platform, Liferay is framework-agnostic. Developers can use any preferred JavaScript libraries.

Mind map summarizes the typical areas of user interface customization in Liferay:



# NBP Liferay Platform Architecture (To-BE)

The NBP platform will be using Liferay DXP 7.4. The diagram below outlines the system architecture of the NBP Landscape.



1. NBP will run on two main Liferay nodes that will be clustered to replicate the cache and provide high availability
2. Internet traffic is directed to both the servers using Load Balancer
3. Apache Web server will be installed on both nodes and provides the necessary configuration to
4. Provide virtual host settings
5. Managing Browser Caches
6. Redirect URLS to specific resources withing Liferay or outside if necessary
7. Serve any static content reducing the overall load on the application server
8. Tomcat Application server will used to serve Liferay dynamic pages (Liferay will be hosted within the Tomcat application

server for the main website with domain name)

1. Liferay Portal database will be hosted on MYSQL 5.X
2. Liferay Document Library will be stored and mounted on NFS file server
3. Users will be created in Liferay but also synced with external Active Directory (LDAP) server
4. Elastic Search will be used for searching and indexing for better search experience.
5. Rest service end point will be exposed using JX-RS to be consumed by BPM to store user application details.

Proposed Project Development Plan

