h



Manulife SPI

Application Architecture Phase 1

Version 0.9

Prepared By

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 30-Jul-2013 | V 0.7 | Initial Draft | iGATE |
| 16-Sep-2013 | V 0.8 | Amended Section 10.13, 10.14, 13.1 | iGATE |
| 15-Nov-2013 | V0.9 | Amended Section 4.13, 13.5, 13.6 | iGATE |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[Revision History 2](#_Toc372526877)

[1 Executive Summary 6](#_Toc372526878)

[1.1 Project description and background 6](#_Toc372526879)

[1.2 Business Context 6](#_Toc372526880)

[1.3 Business Objectives 7](#_Toc372526881)

[1.4 Business Requirements 7](#_Toc372526882)

[1.5 Technical Requirements 8](#_Toc372526883)

[1.6 Business Benefits 8](#_Toc372526884)

[1.7 Scope 8](#_Toc372526885)

[1.8 Inclusions 8](#_Toc372526886)

[1.9 Exclusions 9](#_Toc372526887)

[1.10 Key Risks 9](#_Toc372526888)

[1.11 Technologies 9](#_Toc372526889)

[1.12 Implementation Considerations 11](#_Toc372526890)

[1.13 Timelines 11](#_Toc372526891)

[1.14 Cost Guidelines 11](#_Toc372526892)

[2 Introduction 12](#_Toc372526893)

[2.1 Objective 12](#_Toc372526894)

[2.2 Definitions, Acronyms and Abbreviations 12](#_Toc372526895)

[2.3 References 13](#_Toc372526896)

[2.4 Key Stakeholder Summary 13](#_Toc372526897)

[2.5 Contributors 13](#_Toc372526898)

[3 Current System 14](#_Toc372526899)

[3.1 Websites 14](#_Toc372526900)

[3.2 Software Infrastructure 14](#_Toc372526901)

[3.3 Hardware Infrastructure 15](#_Toc372526902)

[3.4 Security 16](#_Toc372526903)

[3.4.1 Authentication and Authorization 16](#_Toc372526904)

[3.4.2 Data Privacy 16](#_Toc372526905)

[3.4.3 IP address Check 16](#_Toc372526906)

[4 User Descriptions 17](#_Toc372526907)

[4.1 Business User Summary 17](#_Toc372526908)

[4.2 Users and Roles 17](#_Toc372526909)

[4.3 Internal Users 17](#_Toc372526910)

[4.4 Administrator Users 17](#_Toc372526911)

[4.5 External Organization Users 17](#_Toc372526912)

[4.6 Individual Agents (Plan Advisories) 18](#_Toc372526913)

[4.7 Policy Holders 18](#_Toc372526914)

[4.8 Number of Users 18](#_Toc372526915)

[4.9 User Management and Administration 18](#_Toc372526916)

[4.10 Authentication & Authorization 18](#_Toc372526917)

[4.11 Federated User Registry approach 18](#_Toc372526918)

[4.12 Phase 2 extension Strategy (PA channel expansion) 19](#_Toc372526919)

[4.13 Alternative to Custom User Repository Implementation 19](#_Toc372526920)

[5 Functional Requirements 21](#_Toc372526921)

[5.1 Portal Requirements 21](#_Toc372526922)

[5.2 Personalization 21](#_Toc372526923)

[5.3 Dashboards 21](#_Toc372526924)

[5.4 Knowledge Reporting 21](#_Toc372526925)

[5.5 Manulife News and Announcements 21](#_Toc372526926)

[5.6 Social Collaboration 21](#_Toc372526927)

[5.7 Portalizing Internet/Intranet Applications 21](#_Toc372526928)

[5.8 Integrating Internet/Intranet Applications 21](#_Toc372526929)

[5.9 Products packaging summary 22](#_Toc372526930)

[5.10 Out of the box Features 22](#_Toc372526931)

[6 Interface Specification 23](#_Toc372526932)

[6.1 Internal Communication 23](#_Toc372526933)

[6.2 External Communication 23](#_Toc372526934)

[7 Supplementary Specification 24](#_Toc372526935)

[7.1 Application and Software Standards 24](#_Toc372526936)

[7.2 Mobile Enablement 24](#_Toc372526937)

[7.3 Offline Access Requirement 24](#_Toc372526938)

[7.4 Hardware Standards 24](#_Toc372526939)

[7.5 Access and Security Standards 24](#_Toc372526940)

[7.6 Environmental and Location Factors 25](#_Toc372526941)

[7.7 Usability Standards 25](#_Toc372526942)

[7.8 Performance, Service Level and System Availability 25](#_Toc372526943)

[7.9 Audit Control, Archival and Data Retention 26](#_Toc372526944)

[7.10 Caching 26](#_Toc372526945)

[7.11 Maintenance and Support 26](#_Toc372526946)

[7.12 Licensing and Implementation 26](#_Toc372526947)

[7.13 Documentation and Online Help 27](#_Toc372526948)

[7.14 Testing and Training Environment 27](#_Toc372526949)

[7.15 Legal and Regulatory Requirement 27](#_Toc372526950)

[8 Solution Options 28](#_Toc372526951)

[8.1 Portal Solution 28](#_Toc372526952)

[8.2 Client Side Rendering 29](#_Toc372526953)

[8.3 Pros and Cons Client Side Rendering (Portal) 29](#_Toc372526954)

[8.4 Server Side Rendering (Portal) 30](#_Toc372526955)

[8.5 Pros and Cons Server Side Rendering (Portal) 31](#_Toc372526956)

[8.6 Single Page Client (WEB) – Client Centric Solution 31](#_Toc372526957)

[8.7 Pros and Cons – Single Page Client (Web) 32](#_Toc372526958)

[8.8 Proposed Solution 32](#_Toc372526959)

[8.9 Solution Highlights 33](#_Toc372526960)

[8.10 Technology Stack 33](#_Toc372526961)

[9 Architectural Decisions 34](#_Toc372526962)

[9.1 Presentation Layer Framework 34](#_Toc372526963)

[9.2 UI Layer JavaScript Library 35](#_Toc372526964)

[9.3 Deployment Consideration (WPS and WAS versions) 35](#_Toc372526965)

[10 SPI Application Architecture Definition 37](#_Toc372526966)

[10.1 Conceptual View 37](#_Toc372526967)

[10.2 Logical View 37](#_Toc372526968)

[10.3 Layered Architecture 38](#_Toc372526969)

[10.4 Client Layer 38](#_Toc372526970)

[10.5 Presentation Layer 38](#_Toc372526971)

[10.6 Business Layer 38](#_Toc372526972)

[10.7 Data Access Layer 38](#_Toc372526973)

[10.8 Data Layer 39](#_Toc372526974)

[10.9 Service Enablement 39](#_Toc372526975)

[10.10 Packaging & deployment 39](#_Toc372526976)

[10.11 Conversion Guidelines 39](#_Toc372526977)

[10.12 Reusable components design 39](#_Toc372526978)

[10.13 Use case View 41](#_Toc372526979)

[10.13.1 Functional Use case View 42](#_Toc372526980)

[10.14 Data View 43](#_Toc372526981)

[10.15 Security View 43](#_Toc372526982)

[10.16 Physical View 43](#_Toc372526983)

[10.17 Development and Test Environment 45](#_Toc372526984)

[11 Risks, Assumptions, Issues, Dependencies and Constraints 46](#_Toc372526985)

[11.1 Risks 46](#_Toc372526986)

[11.2 Assumptions 46](#_Toc372526987)

[11.3 Issues and Unresolved Requirements 47](#_Toc372526988)

[11.4 Dependencies 47](#_Toc372526989)

[11.5 Constraints 47](#_Toc372526990)

[12 References 48](#_Toc372526991)

[12.1 Standard Compliance References 48](#_Toc372526992)

[12.2 Best Practices References 48](#_Toc372526993)

[13 Appendices 49](#_Toc372526994)

[13.1 Offline Mode Options 49](#_Toc372526995)

[13.1.1 Considerations for Offline Mode Access 49](#_Toc372526996)

[13.1.2 Conceptual Solution Model 49](#_Toc372526997)

[13.2 Capacity Planning 50](#_Toc372526998)

[13.3 Testing Strategy 50](#_Toc372526999)

[13.4 Batch Mode 50](#_Toc372527000)

[13.5 WODM integration with services layer 50](#_Toc372527001)

[13.6 Payment Gateway Interface and WODM Integration 51](#_Toc372527002)

# Executive Summary

### Project description and background

Manulife Japan is facing constant challenge in coping up with the highly agile requirements of the sales and agency users and channel partners who are involved in selling Manulife products. A recent study conducted in Manulife Japan by a consulting firm (KPMG Cummins) also proposed a shift in the system outlook towards a customer centric approach than a product centric approach for enhancing the sales and better customer satisfaction. The study also recommended considering the products as packaged offerings to suite the diverse requirements of retail and corporate customers.

There are challenges due to browser compatibility in the current sales applications. It has always been a daunting task to enhance the features in the current applications and at the same time provide backward compatibility to older browsers. With the advent of mobile devices, the demand for supporting mobile devices and tablets for people on the field is also in the rise. There is also an increasing demand to provide offline mode functionality to the Sales applications for field users with limited connectivity.

Previous studies and feedback from the Agency users also brought to light the need for enhancing the user experience and giving a fresh new look to the sales channel applications. Architectural assessments conducted on Agent Web application also revealed the need to modernize and revamp the existing application layer by layer to overcome the performance related pain points.

Sales Process Innovations was setup to take the assessment recommendations to the next level and analyze the options that can meet the end user requirements without having any impact on the business. There is an ongoing effort in other Manulife divisions to provide a self service capabilities and personalized content through a portal. MLJ also has adopted this trend and has done a preliminary POC to portalize the existing Agent Web application in collaboration with iGATE. The success of the POC has given the confidence to pursue the portal journey in a phased manner ensuring that there is no disruption of the services.

In Phase 1 Day 1.0, the portal is intended to be rolled out for Agency Users for retail business. Agency users will access the AS-IS AGW application for corporate proposals through the portal using SSO implementation between the portal and AGW. The Bank channels will continue to access the AS-IS system through the NTT Gateway – iWIN integration mechanism. There will be no change in the HQ User’s access as well to the existing system.

In Phase 1 Day 1.1, the portal will be rolled out for corporate business and will be enabled for Bank Channels through tablets.

### Business Context

The illustration below provides a conceptual model of the SPI Portal. As depicted, the portal will have support for multiple devices (Tablets and Desktops) and multiple browsers. The portal will provide a highly responsive User Interface using AJAX and JSON with minimal screen refreshes. The user repositories of the existing application will be federated in to the portal to avoid migration effort. The business functions of the existing applications will be revamped to support the packaged product offerings and exposed as reusable RESTful WebServices. The Data Layer also will be revamped to accommodate the new requirements.

Fig 1: Conceptual Model

### Business Objectives

The objective of the SPI Portal project is to realize a portal platform that will enable Manulife Japan to

* Enhance the sales by providing packaged product offerings
* Make the applications more agile and configurable to changing business needs
* Transform the sales applications to a provide highly responsive and consistent User Interface across all channels
* Provide support for multiple devices and offline mode access
* Ensure better user experience and better customer satisfaction
* Single application for multiple sales channels
* Multi product illustration & application reports ( grouped by plan details)
* Time-to-market for new product developments
* Product sales improvements
* Process time improvements
* Competitive differentiation

### Business Requirements

High level business requirements of the SPI Portal are summarized below:

* Customer Centric business functions and processes
* Packaged product offerings in place of individual products
* Centralize the customer management from illustration & application modules
* Enhanced User Experience
* Consistent Look and Feel across channels
* Personalized and customized content delivery for partners and individuals
* Mobile Enablement – Multi Devices
* Support Offline Mode
* Phased Rollout for Retail and Corporate business
* Self Service capabilities
* Paper less process
* Storing PDF files for NB reports
* E-self disclosure ( auto underwriting)
* 1P & 2P billing process
* E-signature
* 1P bank transfer
* NB enablement for other distribution channels & extensions to PA & FID channels
* Self-registration & unlock features( create new user & unlock password etc)

### Technical Requirements

High level technical requirements of the SPI Portal are summarized below:

* Re-use existing Assets (Software / Hardware / User base)
* Code once, Use anywhere - any browser, any device
* Address browser compatibility related pain points in the current system
* Service Enablement of Business Processes
* Resolve Performance related pain points
* Ease of maintenance

### Business Benefits

The SPI Portal project intends on reaping the following benefits:

| **Benefit** | **Approach** |
| --- | --- |
| Enhances Sales | * Packaged product offerings and ability to meet diverse requirements of the customers |
| Better customer satisfaction | * Ability to illustrate various plans with different products and allow the customer to choose the plans best suited to his requirements |
| Increased transparency | * The packaged offering provides lot of flexibility and freedom for the customers and agents and enables them to have a holistic view of all the products and policies from the customer perspective. |
| Process Efficiency and increased productivity | * The SPI Portal will enable reuse of information and reports across different users of the system. Duplication of efforts in the existing process can also be identified and eliminated. |
| Information consistency and Accuracy | * Service enablement of the business functions will ensure that there is consistency on obtaining and processing the information. |
| Efficiency in data maintenance | * Self service capabilities exposed to a wider audience will make the information update process more efficient and reliable. |
| Expedition of customer query processing | * Enhance user interface, will enable the users to get a 360 Degree view of the customers and the capacity to provide answers to ad-hoc queries of the customers. |
| Predictive Modeling | * Trends in business can be derived from the available data and can be used for sales forecast. |

Table 1: Business Benefits

### Scope

The SPI Portal scope inclusions and exclusions are outlined in this section.

### Inclusions

The Architecture document applies to the proposed SPI Portal system. The portal is intended to be rolled out in a phased manner for Retail and Corporate business and to mitigate any risks in a full-fledged rollout. Features in scope are categorized in to Phase 1 (Day 1) and Phase 2 (Day 2) as below:

**Phase 1 Scope**

* Provide a new Portal platform for Agency users to access the retail business functionality
* Agent Web (AGW) application’s Retail functions for Agency Users is the primary scope
* UI and Portal Presentation Layer implementation of the SPI Portal
* New Data Layer implementation to accommodate packaged product offering
* New Data Access Layer Implementation
* New Business Layer Implementation
* Service enabling the new Business Layer
* Custom User Repository Implementation of the AGW application in the portal
* Rollout the Sales Applications to Agency Users in all devices including tablets
* Application server migration from WAS 7 to WAS 8
* NBOS integration
* WODM integration for sales conditions execution
* User base migration from M\_USER to TDS
* Daily feed from M\_USER to TDS
* Plan & customer record lock to prevent the updates from the other agents

**Day 1.1 Scope**

* Rollout the portal for corporate business and bank channels.
* Implement the offline capability
* E-Signature and Paper Less office capability
* Bank channel & HQ users enablement through portal
* Batch process

### Exclusions

Important exclusions include:

* Detailed consideration and enhancement of any systems that collaborate with the SPI Portal.
* Data Migration of the AS-IS Application Data to the newly designed DB.
* Bank channel & HQ users enablement through portal
* Batch process
* Support for IE6 browser
* Customer group associations
* Existing inquiry screen’s migration to SPI system

### Key Risks

Listed below are some of the key risks identified:

* Enhancements to the existing applications can proliferate due to the necessary changes to data structures.
* Custom User Repository implementation for reusing the existing user repository may be very involved and cumbersome due to the gaps in the user repository
* Co-existence of the AS-IS Infrastructure/Applications and the new Portal Infrastructure/Applications can introduce more maintenance overheads and integration issues
* Phased rollout of the portal to corporate business would necessitate data migration efforts later.
* Shifting the business logic embedded in the Stored Procedure to new the business layer can be challenging.

**Note**: The [Risks](#_Risks) section of this document provides an impact analysis and the possible mitigation actions for the risks.

### Technologies

The technology stacks for implementing the different layers of the portal application are outlined below:

| **Product / Software** | **Version** | **Comments** |
| --- | --- | --- |
| **UI Layer** | | |
| HTML | 5 | New UI components the Local Store option provided by HTML 5 will be leveraged |
| CSS | 3 | CSS 3 will be used to control the style of the web pages |
| JQuery Mobile | 1.3.1 | JQuery Mobile is a JavaScript framework allows us to design a web application that will work on all devices (Smartphone, Tablet and Desktop Platforms). It also provides a rich set of predefined UI components that can be leveraged |
| Angular JS | 1.0.7 | Angular JS is JavaScript framework that extends the MVC capability to the browsers and also helps in binding the HTML form data to JSON data easily. |
| JSON |  | JSON is a Data Interchange structure used mainly for AJAX based communication and REST Based Web Services |
|  |  |  |
| **Portal Layer** | | |
| JSR 286 | Portlet 2.0 | JSR 286 is portlet specification standard. All portlets implemented for SPI Portal will follow JSR 286 standards |
| Spring MVC Portlet | 3.2.3 | Spring MVC Portlet framework will enable us to make the code more manageable and modular in the server side. |
|  |  |  |
| **Business Services** | | |
| Spring | 3.2.3 | The service layer implementation will follow interface/implementation pattern and will be annotation driven. |
| REST |  | RESTful webservices are based on HTTP methods and typically provides a Base URI, Supported Mime Types and a set of operations (POST, GET, PUT, DELETE) supported. Using JAX-RS annotations, we can easily implement RESTful web services out of the Business Services. IBM provided JAX-RS will be used for REST service implementation. |
|  |  |  |
| **Data Access Layer** | | |
| JPA | 2.0 | The Entity Model corresponding to the Relational Tables can be derived using JPA. The relationships among the entities (One to One, One to Many etc) are resolved using annotations. The entity model will form the basis for Data Access. |
| Hibernate | 4.2.3 | Hibernate is an implementation of JPA specification and will be configured in the Spring configuration files as the JPAVendor. |
|  |  |  |
| **General** | | |
| JEE | 6 | JEE 6 supports Dependency Injection, Servlet 3.0, JPA 2.0, JAX-RS 1.1, JAX-WS 2.2 and JAX B 2.2. SPI Portal and the Services application will be JEE 6 based. |
| Log4J | 1.2.15 | Log4J will be configured to log the Errors, Warnings, Info and Debug messages. |
| Acrobat Reader | 6.1 |  |
|  |  |  |
| **Software Infrastructure** | | |
| Client |  | Windows XP / Vista with IE 8.0 /9.0 , Chrome, Mozilla, Safari |
| IBM HTTP Server | 8.0 | For serving the static contents |
| IBM WebSphere Portal Server | 8.0 | IBM WebSphere Portal Server is a JSR 286 based portlet container. It provides several Out of the Box Portlets and other features like SSO that can be leveraged for the SPI Portal |
| IBM WebSphere Application Server | 8.0 | The services component of the applications that are portalized will be deployed in the WebSphere Application Server. |
| DB2 | V9.7.5 EE | The new application data and portal data will be defined in the DB2 database |
|  |  |  |
|  |  |  |

Table 2: Technologies

### Implementation Considerations

The SPI Portal will be implemented in a Phased manner. The portal will be rolled out for Retail business first and the AS-IS application will be retained for corporate business in the interim. The portal solution will retain most of the existing functionality and introduce additional functionality for packaged product offerings. Introduction of the new functionality will be to ensure better user experience and minimal impact on existing applications.

The SPI portal project comprises of the following independent implementation modules, based on the layered architecture and separation of concerns. This segregation allows us to comprehend the functions and use the specialized skills at appropriate layers.

1. UI Layer Implementation
2. Portal Implementation
3. Custom Login Module Implementation
4. Service Layer Implementation
5. Service Enablement
6. Data Access Layer Implementation

There will be 2 major deployable, one for the Portal (Portlet component) and another for the Application Server (Service Component). These components can be independently developed and tested and integrated later.

### Timelines

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Functions | Q3 2013 | Q4 2013 | Q1 2014 | Q2 2014 | Q3 2014 | Q4 2014 |
| Portal POC |  | √ |  |  |  |  |
| Retail Rollout |  |  | √ |  |  |  |
| Corporate Rollout |  |  |  | √ |  |  |
|  |  |  |  |  |  |  |

Table 3: Timelines

### Cost Guidelines

<TBD>

# Introduction

### Objective

This document explains the Application Architecture adopted for SPI Portal project. This document has to be referred for any design decisions. This document details the high level technical design details to be followed in the SPI Portal.

The targeted audience for this document is technical personnel, business analysts, developers who are involved in the SPI Portal implementation. The reader of the document is assumed to have understanding of background and requirements of the SPI Portal project.

The purpose of this document is to provide a complete, accurate and unambiguous description of the functional and technical requirements of the SPI Portal and the Architecture and Design guidelines to be followed for the implementation. As a result this document will provide a baseline agreement, appropriately signed off confirming that:

* The nominated business and technical representatives have confirmed that the portal requirements are correctly and completely specified
* Manulife Infrastructure personnel have confirmed compliance with technology stack, strategy, security and quality standards
* The SCV, SSW, AGW, Unisys, Vantage and Ingenium system owners have confirmed their understanding of the interfacing requirements.
* The team responsible for Bank Channels through iWIN have confirmed their understanding of the interfacing requirements

### Definitions, Acronyms and Abbreviations

|  |  |  |
| --- | --- | --- |
| Acronym / Abbreviation | Name | Description |
| AGW | Agent Web |  |
| AJAX | Asynchronous JavaScript and XML |  |
| API | Application Program Interface |  |
| HTML | Hyper Text Markup Language |  |
| JEE | Java Enterprise Edition |  |
| JPA | Java Persistence API |  |
| JSON | JavaScript Object Notation |  |
| JSR | Java Specification Request |  |
| LDAP | Light Weight Directory Access Protocol |  |
| MVC | Model View Controller |  |
| POC | Proof of Concept |  |
| REST | Representational State Transfer |  |
| SCV | Single Customer View |  |
| SPI | Sales Process Innovations |  |
| UI | User Interface |  |
| SSW | Sales Support Web |  |
| XML | Extensible Markup Language |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table 4: Definitions, Acronyms and Abbreviations

### References

This section provides the list of documents referred:

|  |  |  |
| --- | --- | --- |
| Document Name | Version | Location |
| SPI\_Application\_Architecture\_Proposal.ppt |  |  |
| Manulife-AGWPortalSolutionApproach\_V1.0.pptx |  |  |
| Manulife - Agent WEB Application Architecture Assessment Report v1.0.docx |  |  |
| SPI Infrastructure View 0.1.pptx |  |  |
| SPI Design Guidelines.1.0.pptx |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table 5: References

### Key Stakeholder Summary

There are a number of stakeholders with an interest in the development of the SPI Portal. The Table below presents a summary list of key stakeholders

|  |  |  |
| --- | --- | --- |
| Stakeholder Type | Name | Project Responsibilities |
| Project Sponsor |  |  |
| Program Manager |  |  |
| Project Manager |  |  |
| SME – Business |  |  |
| SME – Technical |  |  |
| SME - Infrastructure |  |  |
|  |  |  |
|  |  |  |

Table 6: Stakeholder Summary

### Contributors

The following individuals have contributed to this document and project:

|  |  |  |
| --- | --- | --- |
| Name | Role | Business Unit / Department |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table 7: Contributors

# Current System

### Websites

The lists of Internet and intranet applications accessible to different user categories are as in the table below:

| **Application** | **User Type** | **App Type**  **(Internet / Intranet)** | **URL** |
| --- | --- | --- | --- |
| AGW | MGA / Agency Users |  |  |
| HQ Users |  |  |
| Bank Users |  |  |
| PAW | PA Users |  |  |
| HQ Users |  |  |
|  |  |  |  |

Table 8: Websites

### Software Infrastructure

The AS-IS applications are JEE based 3-Tier application hosted on WebSphere Application Server. The Architecture and Design of these applications follows standard JEE principles and clear separation of concerns. The Presentation tier is implemented using Struts, the Business Logic layer is implemented using Spring and the Data Access Layer is implemented using Hibernate. The Business and Data Access layer follows the Interface/Implementation pattern. Although Hibernate is used, the ORM features of Hibernate is not exploited to its potential in the AS-IS applications. Additionally, most of the business logic is implemented in the Stored Procedures, causing heavy utilization of DB Server resources.

The Technology stack of the AS-IS application are as listed below:

|  |  |  |
| --- | --- | --- |
| **Product** | **Version** | **Comments** |
| Java2SDK | 1.5 |  |
| J2EE | 1.4 |  |
| Log4J | 1.2.15 |  |
| Spring | 2.0.2 | Business layer implementation |
| DB2 | DB2 V9.5 Enterprise Server Edition | Some of the business logic lies in the Database causing heavy utilization of CPU and Memory |
| WebSphere | 7.0 | Application Server where the applications are hosted |
| Struts | 1.2.9 | Presentation Layer implementation |
| Client | Windows XP / Vista with IE 6.0, IE 7.0 browser |  |
| Acrobat Reader | 6.1 and above |  |
| Hibernate | 3.2.6 | Data Access Layer |

Table 9: AS-IS Technology Stack

### Hardware Infrastructure

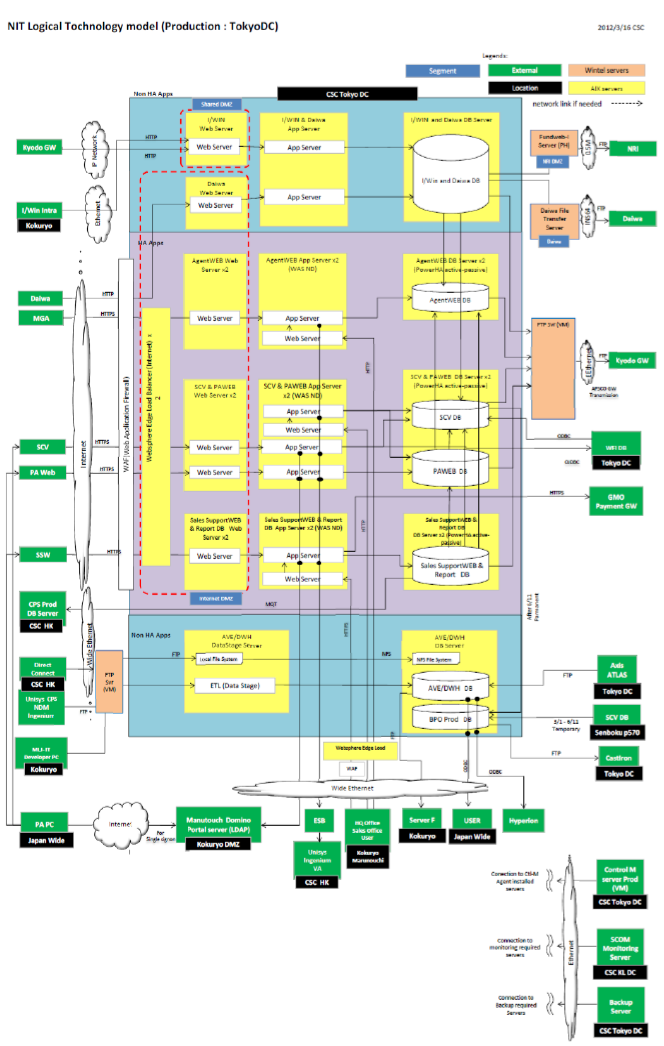


Fig 2: AS-IS Infrastructure

### Security

The following are the general application security aspects that are followed in AS-IS AGW application

* Authentication and Authorization
* Data Privacy
* IP Address Check

### Authentication and Authorization

The AS-IS AGW application is accessed through different channels, MGA Users, Bank Users and HQ users. The AS-IS applications maintains its own user repository. Authentication and Authorization are more of application responsibility than the infrastructure responsibility. The application uses Role Based authorization. Application role to User/Group mapping is maintained in the DB and is mapped for each user. This mapping is employed for coarse grained (declarative) security.

Bank uses coming through NTT Gateway are authenticated in iWIN WEB Plug-in. There is a trust association established between iWIN and AgentWEB Bank ear module in the WAS server. The credentials of the Bank User authenticated by iWIN is propagated to AGW Bank for controlled access.

### Data Privacy

All the User Interface pages served by MGA system are over HTTPS. It is assumed that there are no special requirements for data privacy above and beyond the use of HTTPS to serve HTML.

### IP address Check

AGW verifies the client IP address against the IP which are maintained in property file before allowing download of search results in CSV files.

# User Descriptions

### Business User Summary

|  |  |  |  |
| --- | --- | --- | --- |
| Role | Description of Role | Responsibilities | Represented By |
| Agent |  |  |  |
| Bank User |  |  |  |
| HQ User |  |  |  |
| SPI Portal Admin |  |  |  |
| AGW Admin |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table 10: Business User Summary

### Users and Roles

Important User Roles identified in the AGW application are:

|  |  |  |  |
| --- | --- | --- | --- |
| Role Id | Role Name | Description of Role | Represented By |
| 11 | MGA administrator |  |  |
| 12 | MGA users |  |  |
| 21 | MGA agent Admin |  |  |
| 22 | MGA Agent user |  |  |
| 42 | WMD (FID) users |  |  |
| 51 | WMD (FID) agency users |  |  |
| 61 | MLJ administrator |  |  |
| 62 | MLJ users |  |  |
| 72 | DSC users |  |  |
|  |  |  |  |
|  |  |  |  |

Table 11: Users and Roles

### Internal Users

The following are the MLJ users who can access AGW application internally.

* MGA users
* MLJ users
* MLJ administrator

### Administrator Users

The following are the MLJ administrator users who have administrator privileges to AGW application:

* MGA administrator
* DSC users

### External Organization Users

The following are the external organization users:

* MGA agent Admin
* WMD (FID) users
* MGA Agent user

### Individual Agents (Plan Advisories)

### Policy Holders

### Number of Users

The total number of users that could use the SPI Portal is not limited. There are approximately 9999 agent users across Japan who will access the SPI Portal for retail business. Estimated number of Bank users who will access the SPI portal once it is rolled out to corporate business are 9999. At a peak load it is estimated that 80% of these users will be accessing the portal for various transactions. The user base is expected to grow by approximately at the rate of 10% per annum.

### User Management and Administration

In Phase1 we will to continue use the as is system for user Management and administration. Currently HQ users perform the following user management tasks:

* Creating a new user.
* Edit the user information.
* Revoke/reset the password.

In Phase2 we will leverage the WebSphere Portal out of the box features for self registration to register the user. Portal Administrator will perform the user management and administration with the built in UI provided by the WebSphere portal. The portal will use the federated repository to store the information and communicate via the PUMA API & VMM SPI.

**Note:** Implementation details are covered in SPI – Design Guidelines document.

### Authentication & Authorization

**Phase1:**

* MGA / Agency users are authenticated via portal.
* Bank Users are authenticated by NTT Gateway and iWIN Plugin, as in the AS-IS system. SSO will be implemented to connect to portal.
* PAW users are authenticated via existing form based login.

**In Phase2:**

* PAW users are authenticated via portal along with MGA/Agency users.

### Federated User Registry approach

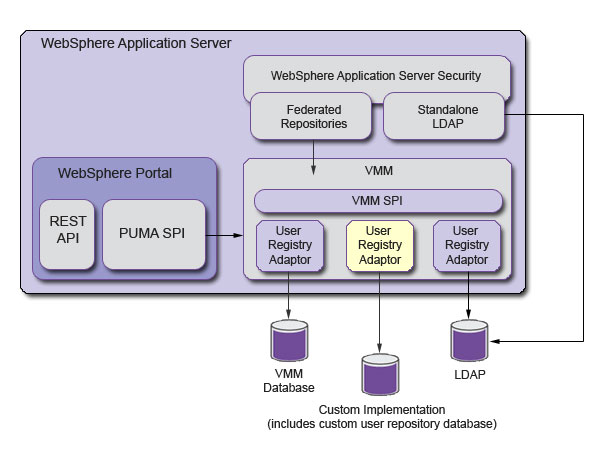


Fig 3: Federated User Registry Approach

The Federated User Registry provided by IBM enables us to reuse the existing applications User base, with out the need to migrating to the portal’s LDAP repository. The approach requires us to provide implementation for UserRegistry interface of Service Provider Interface (SPI) provided by IBM and configure Federated WebSphere Portal to use the Custom User Registry.

**Note:** Implementation details are covered in SPI - Design Guidelines document.

### Phase 2 extension Strategy (PA channel expansion)

Fig 4: PA Channel Expansion in Phase 2

As new applications are portalized in Phase 2, we can incrementally add the User Repositories of these applications to the Portal Federation by implementing the Custom User Registry.

### Alternative to Custom User Repository Implementation

An alternative to Custom User Repository implementation is to configure the Tivoli Directory Server to the Federated Repository. As TDS is a LDAP repository, there is no need for Custom User Registry implementation. Once TDS is configured to the portal, a onetime migration of the User Data from the existing applications (AGW, PAW etc) to Tivoli Directory Server will be performed. The users’ login to the portal will now be authenticated against TDS. The user data migrated from the existing user repositories (Eg: M\_USER table of AGW) include USER\_ID, PASSWORD, and GROUP information. Additional information available in the relational tables like AGENCY\_CD, BRANCH\_CD, SALES\_DEP\_CD etc are not migrated to TDS.

The Application Server administration console will be used to perform USER/GROUP mapping to the portlet ROLES. High level access control (ROLE-BASED) using PUMA API will be performed based on the above ROLE mapping. For example personalization of content based on Logged in user’s channel will be achieved by the ROLE mapping approach.

For more granular authorization, the portlet application will refer the additional information available in the existing relational DB User tables (Eg M\_USER). For example any personalization or access restrictions based on which AGENCY, BRANCH etc of a logged on user will be controlled using the additional information from the relational tables.

The additional information will be made available to the portlets by service enabling the User information.

The relational DB USER tables are administered using an external feed from CPS application. The updates (INSERT/UPDATE/DELETE) on the USER tables will have to be synchronized with the TDS in batch mode. Tivoli Directory Integrator will be used for this purpose.

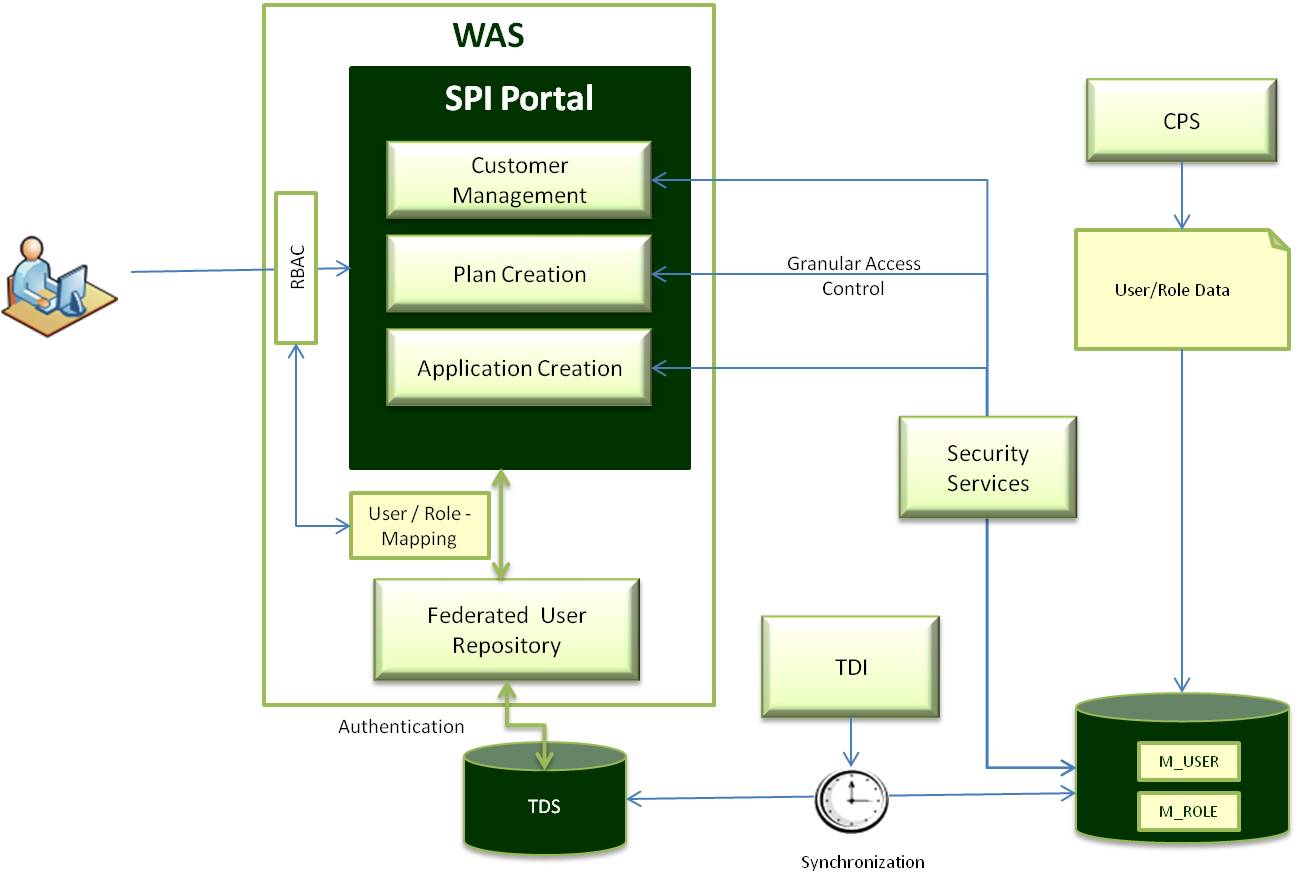


Fig ##: TDS Implementation

# Functional Requirements

### Portal Requirements

Develop a portal that provides the AGW application with the modernization of running the business using the following:

* Packaged solutions instead of existing individual products.
* The portal should be device independent
* Enhanced User Experience
* Consistent Look and Feel across channels
* Mobile Enablement – Multi Devices
* Support Offline Mode
* Self Service capabilities

### Personalization

The Portal should provide Personalized and customized content delivery for partners and individuals

### Dashboards

<<TBD>>

### Knowledge Reporting

<<TBD>>

### Manulife News and Announcements

The SPI Portal should enable Manulife Japan to publish news and announcements on the portal

### Social Collaboration

<<TBD>>

### Portalizing Internet/Intranet Applications

The applications that should be brought into the portal in phase1 are:

* AGW application
* AGW Bank application

The following applications should be brought into the portal in phase2:

* PAW application

### Integrating Internet/Intranet Applications

**Day 1.0:**

To allow the agency users to create the corporate proposals from the portal context, existing system (MGA) needs to be integrated to leverage the corporate functionalities. In this case, product packaging for corporate proposals is not applicable. Behavior of the integrated application is identical to existing application.

While integrating the existing system with portal, authentication & authorization parameters will be propagated from the portal context.

Below are the solution options that can be considered for the integration.

* In a new Browser window by means of a hyperlink in the portlet page.
* In an iFrame with in the portal page itself
* As a clipped portlet using WebClipping feature of WebSphere portal

### Products packaging summary

In case of existing system, agency users can create one product at a time for illustration or application. If the customer requests to create multiple products a time, agency users have to create all products (one by one). Along with that, if the customer wants to compare the premium & simulation values between products, agency users will have to create multiple products and print the illustration report for all selected products (one by one) and then compare the results. This process takes considerable amount of time to complete the task.

To address the above issue, MLJ has come up with the solution to package multiple products into a single plan. This means, agency users can create multiple products & multiple plans at a time (without navigating to other screens). In accordance with phase1 scope, one plan can contain four products a time (same product can be attached multiple times). One illustration can accommodate up to five plans in a single window. This gives the flexibility in comparing the premiums and simulations across plans and products.

Day 1.0 scope also includes the consolidated reports printing for illustration & application proposals. Illustration & application print modules organize the reports in plan wise rather than the product wise.

### Out of the box Features

Some of the out of the box portal features of WebSphere portal that can be leveraged are listed below:

* **Banner Ad Portlet** – This portlet can be used for the ad campaign or sales promotion of Manulife Japan
* **Email** **Portlet** – Using this feature the email can be integrated with the portal.
* **Article portlet** –Manulife Japan will make use of this feature to publish news and announcements to its Agents, Partners and Internal employees in its portal as well as any notifications to everyone or a specific group.
* **Web Content Management** - Using this feature Manulife Japan can create the content and display to its Agents, Partners and Internal employees based on their roles. The content creation can also undergo a workflow based process for review and approval.

# Interface Specification

### Internal Communication

The SPI Portal will interface with the following back-end and enterprise systems:

* Single Customer View
* AS-IS Agent Web
* Unisys
* Vantage
* Ingenium

The table below describes the communication From/To the above mentioned interfacing systems.

| **From / To System** | **Data being sent** | **Purpose** | **Protocol** | **Security Implications** |
| --- | --- | --- | --- | --- |
| SCV | Existing Policy Information | Customer Validation | 1. WebServices 2. Back-end DB through Federated Tables | Authenticated AGW User |
| Unisys |  |  | FTP in Batch Mode | SSO |
| Vantage |  |  | FTP in Batch Mode  SOAP Web Services | SSO |
| Ingenium |  |  | FTP in Batch Mode  SOAP Web Services | SSO |
|  |  |  |  |  |

Table 12: Internal Interfaces

### External Communication

| **From / To System** | **Data being sent** | **Purpose** | **Protocol** | **Security Implications** |
| --- | --- | --- | --- | --- |
| Bank Channels | Customer Data, Policy Data. | Access AGW through NTT Gateway to create Corporate Proposals. | HTTP | Through Trust Association between iWin and AGW |
|  |  |  |  |  |
|  |  |  |  |  |

Table 13: External Interfaces

# Supplementary Specification

### Application and Software Standards

| **Requirement Description** | **Priority**  **(M/D/U)\*** |
| --- | --- |
| **Compliance:** The SPI Portal must be compliant to the software standards and guidelines of MLJ | M |
|  |  |
|  |  |

\*Mandatory / Desired / Useful

Table 14: Application and Software Standards

### Mobile Enablement

<<TBD>>

### Offline Access Requirement

<<TBD>>

### Hardware Standards

| **Requirement Description** | **Priority**  **(M/D/U)\*** |
| --- | --- |
| **Compliance:** The SPI Portal must be compliant to the hardware standards and guidelines of MLJ | M |
|  |  |
|  |  |

\*Mandatory / Desired / Useful

Table 15: Hardware Standards

### Access and Security Standards

The SPI Portal should be available to all existing and new users (Agency, Bank and Internal). The Agent Web user repository and other user repositories that are brought under the SPI portal federation will form the basis for user registry. Application access rights should be defined through the respective application's administrative module.

| **Requirement Description** | **Priority**  **(M/D/U)\*** |
| --- | --- |
| **Authentication –** All users connecting to the Portal will be authenticated against the federated user repository. The credential information is expected to be propagated from SPI Portal to the application that is accessed. In case of internal users (HQ Users) the Windows Logon credentials are propagated to the portal. | M |
| **Authorization** – SPI Portal will have a designated Portal Administrator user, capable of granting access rights and privileges to Agents, internal and other end users. Any user accessing the Portal should have the necessary authorization to perform such an operation. Authorization decision will be role based. Authenticated user will be verified against application roles before allowing an operation. | M |
| **Encryption –** All information movement (entering or leaving) is classified as “Sensitive” and must be secure at all times. The data must be encrypted and adhere to the Manulife security policy / encryption standards. | M |
| **Host Security –** The SPI Portal architecture encompasses firewall in front of any interaction with it. All communication will be handled through https (Web Services). | M |

\*Mandatory / Desired / Useful

Table 16: Access and Security Standards

### Environmental and Location Factors

| **Requirement Description** | **Priority**  **(M/D/U)\*** |
| --- | --- |
| The SPI Portal application should be accessible for any user from anywhere in Japan. | M |
| The SPI Portal should support all devices such as Tablets, Deskops and Laptops | M |
| The SPI Portal should work on all popular browsers without any additional plug-in requirement | M |

Table 17: Environmental and Location Factors

### Usability Standards

The usability of the SPI Portal is primarily driven by the Agency users who are accessing the portal. However the Bank users have also confirmed their consent to adopt the new look and feel adopted for the portal.

| **Requirement Description** | **Priority**  **(M/D/U)\*** |
| --- | --- |
| **Language Support** – SPI Portal will have support for Japanese and English (US), provisioning for other languages. | M |
| **Multi Browser Support** – The SPI portal should support all the major browsers:  Internet Explorer 8.0 and above  Mozilla Firefox  Apple Safari  Google Chrome | M |
| **Multi Device Support** – The SPI portal should at the minimum support Microsoft Windows XP and above Client PCs, Android 3.0 and above Tablets, iOS 4.0 and above iPads | M |
| **Navigation** - Simple in approach, user friendly and straightforward navigation. The application should maintain predictability at all times (i.e. common standards in naming conventions should be applied and the user should know what happens next according to the action selected). | M |
| **Intuitive UI** - The User Interface should clearly highlight to the user the Mandatory / Optional / Conditional information to be captured when performing a specific transaction. | M |
| **Look and Feel** - The SPI Portal should maintain consistent look and feel across all Portal Pages, confirming to Manulife standards and guidelines (Logos, Fonts, Colors and so on) | M |
| **Display Resolution** - Display settings are adaptable and support at least 800 X 600 screen resolution. | M |
| **Authentication** - User authentication will be through Portal Authentication mechanism for external users and Windows Logon for internal users. SPI Portal should not impose any additional password challenge for any other application / functionality accessed through the portal. | D |

Table 18: Usability Standards

### Performance, Service Level and System Availability

| **Requirement Description** | **Priority**  **(M/D/U)\*** |
| --- | --- |
| **Performance** - The system must meet the performance requirements and scalable to meet the evolving user base as specified in Section x.x of the document | M |
| **Availability** – The system should be available 99% of the time during office hours allowing for scheduled maintenance downtime | M |
| **Time zone** – All timing will be based on JST (Japan Standard Time) | M |
| **Volumes** – Existing throughput/volumes as defined in Section x.x will be extrapolated to obtain expected future volumes, assuming a 10% annual growth. | M |
|  |  |

Table 19: Performance, Service and System Availability

### Audit Control, Archival and Data Retention

The system should maintain audit log for all transactions.

| **Requirement Description** | **Priority**  **(M/D/U)\*** |
| --- | --- |
| **Audit Control –** The system must leave a digital trail of all the transactions indicating who did, what and when. A complete audit trail of the system must be available for the Portal and Application Administrator to create control reports. | M |
| **Archive** – All information in the system are classified “Sensitive” and must be archived as per the archival policy of Manulife | M |
| **Data Retention** - Housekeeping of unused data to keep the database to performing size will be required. Unused data can be archived and cleared from the active system. Data for closed policies will be retained in the active system for a period of XX days. | M |
|  |  |
|  |  |

Table 20: Audit Control, Archival and Data Retention

### Caching

| **Requirement Description** | **Priority**  **(M/D/U)\*** |
| --- | --- |
| A Caching strategy will have to be formulated to support temporary persistence of intermediate calculations. | M |
| The approach has to be less invasive so as not to introduce any additional tool, cost and maintenance overheads. | M |
| The functionality provided by Caching mechanism will be equivalent of the functionality supported Temptable in the AS-IS Agent WEB application. | M |
|  |  |
|  |  |

Table 21: Caching

### Maintenance and Support

| **Requirement Description** | **Priority**  **(M/D/U)\*** |
| --- | --- |
| **Maintenance and Support** – The SPI Portal system will be supported during the office hours, 5 days a week, in line with the current Sales Channel Application SLAs. | M |
| **Business Continuity Plan** – SPI Portal is a critical system and should have disaster recovery and business continuity plan. Functionality required to tackle contingencies of the SPI Portal will be defined as a part of the SPI Project Production Deployment guideline. | M |
|  |  |

Table 22: Maintenance and Support

### Licensing and Implementation

| **Requirement Description** | **Priority**  **(M/D/U)\*** |
| --- | --- |
| **Licensing** – All Agent Users, Bank Users and HQ Users are customers are automatically entitled to use SPI Portal. The infrastructure/procurement team will ensure appropriate product licenses are procured to accommodate the current and future user volumes. | M |
| **Implementation** –Implementation activities must ensure each function within Manulife understands the impact of this solution on their business processes and organization. | M |
|  |  |

Table 23: Licensing and Implementation

### Documentation and Online Help

| **Requirement Description** | **Priority**  **(M/D/U)\*** |
| --- | --- |
| **Documentation** - A Technical Manual that describes frequent process and system issues that the users may encounter will be provided by the project manager. | D |
| **On-Line Help** - On-line help describing the usage of each of the screen will be provided by the project manager | D |
|  |  |

Table 24: Documentation and Help

### Testing and Training Environment

| **Requirement Description** | **Priority**  **(M/D/U)\*** |
| --- | --- |
| **Testing and Training** – Separate Development, Testing and Training environments will be required for the SPI Portal | M |
|  |  |
|  |  |

Table 25: Testing and Training Environment

### Legal and Regulatory Requirement

| **Requirement Description** | **Priority**  **(M/D/U)\*** |
| --- | --- |
| Sarbanes-Oxley Act |  |
| Data Protection Act |  |
| BS7799 Security Standard |  |
| Compliance with country legislation |  |
| Compliance with government & health and safety regulations |  |
| Provide fully accessible system for all types of users and cater for disabilities (hearing, cognitive, etc.) |  |
|  |  |
|  |  |

Table 26: Legal and Regulatory Requirement

# Solution Options

Manulife has existing investments on IBM based technology stack. Most of the web applications including AGW, SCV are hosted IBM WebSphere Application Server. The data repositories of these applications are on DB2. In addition there is a broader guideline across Manulife divisions to use IBM suite of products. The existing technology team is also well versed with Java/JEE based technologies. Manulife also has adopted open source frameworks and tools for better maintainability of code. Spring and Hibernate has been used in most of the existing application in Manulife

Manulife would want to consider the above aspects and prefer a solution based on Java/JEE technology and leverage the existing investments on IBM.

The solution options under consideration are as depicted in the diagram below:

Fig 5: SPI Portal Solution Options

Primarily, the solution options can be categorized as

* Server Centric – Portal: In order to leverage the Personalization, Customization and other Out of the Box features of the Portal
* Client Centric – Standard Web: In order to provide a highly rich User Interface, completely driven from the Client side (Browser).

The sections below provide detailed overview of the options along with the pros and cons of each of the approaches.

### Portal Solution

A portal based infrastructure would enable Manulife to aggregate all the applications at the presentation layer. The Personalization (ability of the portal to render content and regulate access based on the logged in user) feature of the portal can be very useful. The look and feel of the portal can be personalized for different Agencies, Bank Channels and internal users. Individual users can also customize the Portal Page and content in the potlets according to their preferences. Additionally, portal provides several Out of box Content management and collaboration (Web 2.0) features that can be used.

A portal solution is primarily Server Centric, in the sense that the content aggregation and all the navigation logic happens in the server side. However JSR 286 (Porlet 2.0) specification provides mechanisms for asynchronous calls and client side rendering capabilities.

Spring MVC Portlet API provides an abstraction over the standard Portlet API and relieves the developers from writing any plumbing code. All portlet development for SPI portal would follow Spring MVC portlet framework.

### Client Side Rendering

The illustration below provides the sequence of steps involved in implementing a Portlet page using Client Side Rendering mechanism.

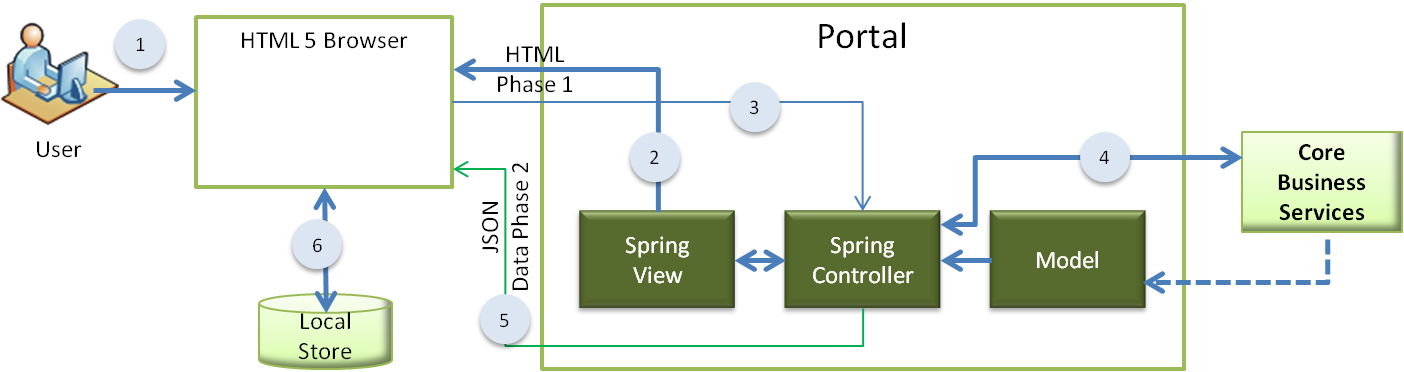


Fig 6: Client Side Rendering (Portal)

This is a 2 Phased approach. In Phase 1 the browser makes a Synchronous call (HTTP Request) and the server responds by sending the HTML mark up with the necessary JavaScript code for making subsequent calls and the JavaScript for DOM Manipulation to generate dynamic UI.

In Phase 2, the client browser makes Asynchronous call (XMLHttpRequest) to obtain JSON Response. The JSON data will be used by the callback mechanism in the browser to refresh only the data in the browser.

* User navigates to the page containing the portlet.
* The view rendering involves 2 phases
  + Phase 1 (HTML View):
    - The Portlet controller’s render method dispatches the request to appropriate view.
    - The view returned contains the standard UI controls that appear in the Header and Footer sections and the place holder for the content.
    - The scripts (JS Libraries) to call the services asynchronously are also sent as a part of the view.
  + Phase 2 (JSON Data):
    - The Script code in the browser will now make an asynchronous call to fetch the data.
    - The ResourceMapping method of the Portlet Controller receives the request and fetches the data from the appropriate Service.
    - The JSON Data is dispatched to the browser
    - The Callback mechanism of the scripts gets notified of the arrival of data, It will persist the data temporarily to HTML 5 local store.
    - The data is then shown in the UI place holders.

### Pros and Cons Client Side Rendering (Portal)

The pros and cons of the Client Side Rendering (Portal) approach are listed in the table below:

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| The solution will provide better user experience | Requires HTML 5 enabled browser to leverage local storage |
| Subsequent rendering of the screens will be faster | Mechanisms to refresh the local store will have to be built to ensure changes are reflected in the client side |
| Optimal use of network bandwidth | Client heavy may require more hardware resources in the client side |
| More oriented towards the evolution to Offline mode and mobile enablement | Learning curve for developers may affect productivity |
| Can leverage Portal features and Out of Box Portlets | Portal may become an overkill if the Customization / Personalization requirements are limited |
| Look and Feel customizations can be easy through themes | First time rendering of the pages may be slow |
| SSO and Re-use of user repository is supported inherently |  |

Table 27: Pros and Cons – Client Side Rendering (Portal)

### Server Side Rendering (Portal)

The Server Side Rendering approach is similar to Client Side Rendering, except that the UI navigation and rendering logic is controlled from the Server than the client. This is also 2 Phased approach. However the content aggregation happens at the server side.

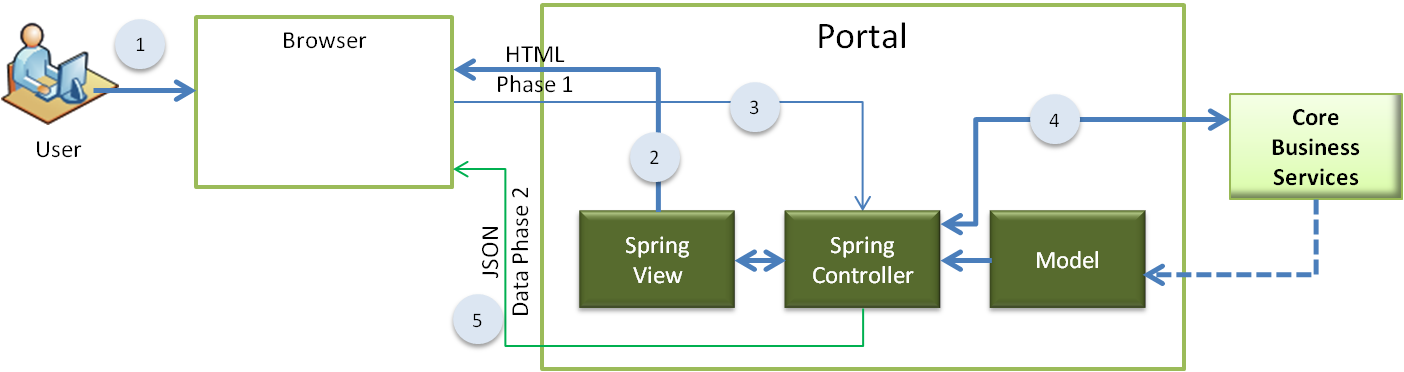


Fig 7: Server Side Rendering Portal

* User navigates to the page containing the portlet.
* The view rendering involves 2 phases
  + Phase 1 (HTML View):
    - The Portlet controller’s render method dispatches the request to appropriate view.
    - The view returned contains the standard UI controls that appear in the Header and Footer sections and the place holder for the content.
    - The scripts (JS Libraries) to call the services asynchronously are also sent as a part of the view.
  + Phase 2 (JSON Data):
    - The Script code in the browser will now make an asynchronous call to fetch the data.
    - The ResourceMapping method of the Portlet Controller receives the request and fetches the data from the appropriate Service.
    - The Json Data is dispatched to the browser
    - The Callback mechanism of the scripts gets notified of the arrival of data
    - The data is then rendered in the place holders.

### Pros and Cons Server Side Rendering (Portal)

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| The entire navigation/presentation logic is governed by the Portal in the server side | Frequent page refreshes may affect user experience |
| Support for older browsers (Does not require HTML 5) | Network bandwidth usage is not optimal |
| Can leverage Portal features and Out of Box Portlets | May not be leveraging modern browser features |
| Look and Feel customizations can be easy through themes | Portal may become an overkill if the Customization / Personalization requirements are limited |
| SSO and Re-use of user repository is supported inherently |  |

Table 28: Pros and Cons Server Side Rendering (Portal)

### Single Page Client (WEB) – Client Centric Solution

This solution option does not involve a portal and the entire navigation and UI rendering logic is client driven using JavaScript libraries. In this solution there is only one synchronous HTTP request from the browser to the server. The launch page along with all DOM manipulation and other JavaScript code is sent to the client browser. All subsequent requests from the browser is Asynchronous calls.

This approach uses HTML 5 standards and leverages the Local Store provided by HTML 5. This solution also uses MVVM framework on the client side.

As the solution does not involve a portal, the Personalization, Customization and other out of box features supported by the portal cannot be leveraged.

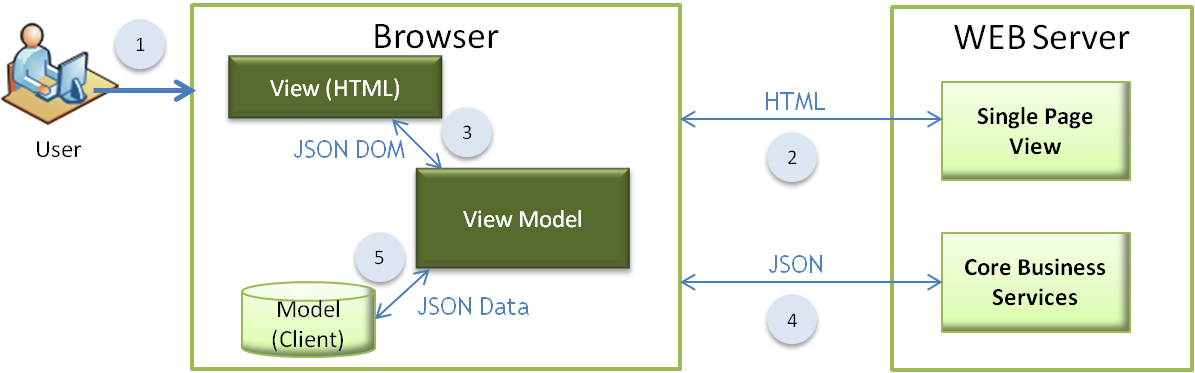


Fig 8: Single Page Client WEB

* This is a client centric solution with only a Single Page View
* This solution does not involve a Portal but a standard Web Container
* The User launches the application through a browser URL
* The Web Server receives the request and responds with the single page view along with all the necessary JavaScript code for Client Side Rendering
* JavaScript manipulates the DOM structure to create the dynamic View
* Client makes Asynchronous Requests for JSON Data from the Core Business Services to refresh content in the page
* Data to restore the view in the client is persisted in the Session Store of the browser

### Pros and Cons – Single Page Client (Web)

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| The entire navigation/presentation logic is governed by the Client Side MVVM Framework | The solution will not be able to address the Personalization / Customization requirements |
| Very rich user experience | The Out of Box portal / portlet features cannot be exploited |
| Better utilization of network bandwidth | Learning curve for developers may affect productivity |
| No latency or overheads due to portal | May not support older browsers |
| More inclined to mobile enablement and offline access |  |

Table 29: Pros and Cons – Single Page Client (Web)

### Proposed Solution

The Client Side Rendering (Portal) option is a hybrid solution providing the benefits of both the Portal and the rich user experience of the Client centric solution. This solution will enable us to leverage the Personalization, Customization, SSO and other out of the box features of the portal and at the same time provide a highly responsive UI of the Client Centric Solution.

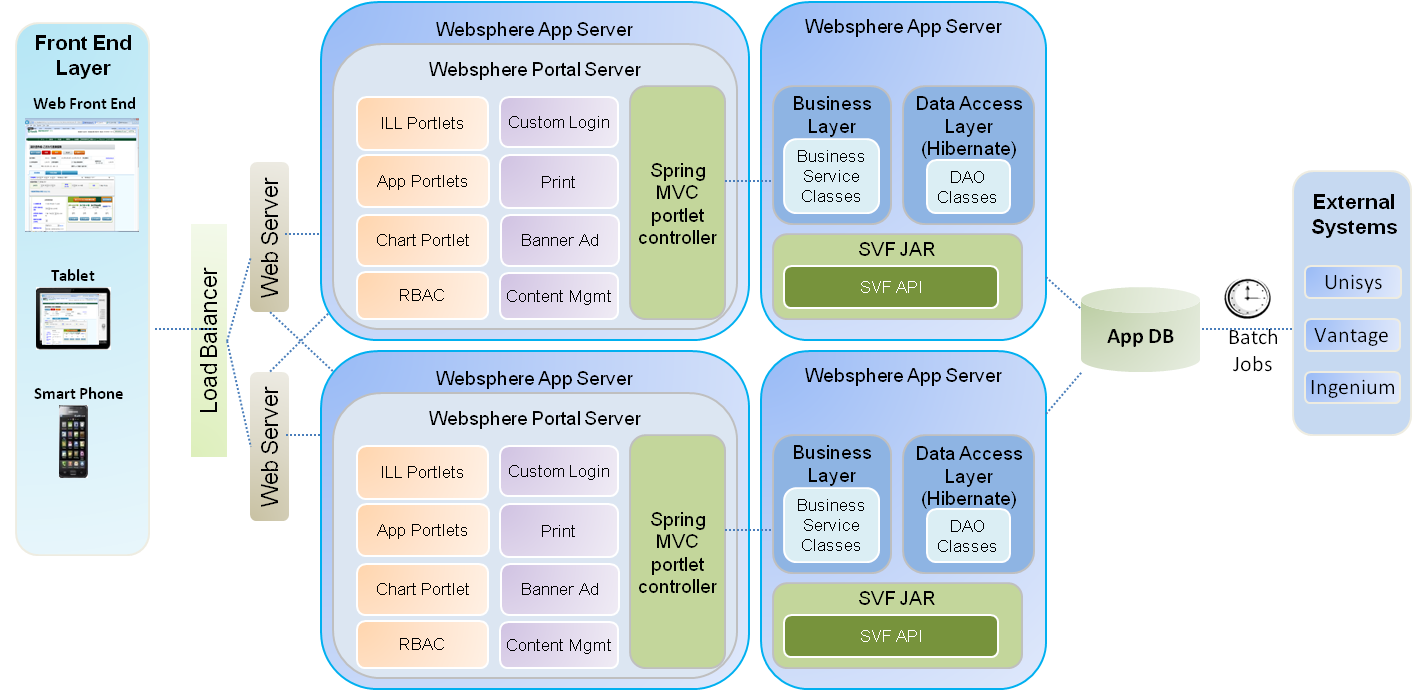


Fig 9: Proposed Solution (Hybrid) – Client Side Rendering Portal

Implementation of this solution involves a Single Page Client based UI for a given functionality. All requests within the context of the given functionality will be serviced using Asynchronous calls which returns JSON Data. Page refresh happens only when the user navigates from a given functionality to another. Any request that results in a screen refresh will be serviced by the Render Mapping method of the Portlet. The response for a Page refresh request will contain the HTML Markup for the new page and all the JavaScript functions to perform DOM manipulation within the page and to make asynchronous data (JSON) calls to the server.

### Solution Highlights

Important benefits of the proposed solution are:

* Multiple Device Support
* Cross Browser Compatibility:
* Responsive UI
* Mobile Enablement
* Customization / Personalization support for different categories of users and Channels
* Single Sign-on support
* Re-usability of existing user repository
* Maintainability

### Technology Stack

A quick summary of the technologies used in the SPI Portal project are provided in the table below. Please refer section [1.11 Technologies](#_Technologies) for a detailed layer wise description of the technologies that will be used in the SPI portal project

|  |  |
| --- | --- |
| **Product / Software** | **Version** |
| Acrobat Reader | 6.1 and above |
| Angular JS | 1.0.7 |
| Client | Windows XP / Vista with IE 8.0, 9.0, chrome, Mozilla  Safari |
| CSS | 3 |
| DB2 | DB2 V9.7.5 Enterprise Server Edition |
| HTML | 5 |
| IBM Http Server | 7.0 |
| J2EE | 1.6 |
| Java2SDK | 1.6 |
| JPA using Hibernate | 3.2.6 |
| jQuery mobile | 1.3.1 |
| Log4J | 1.2.15 |
| Spring | 3.2.3 |
| WebSphere App Server | 8.0 |
| WebSphere Portal Server | 8.0 |

Table 30: Technology Stack

# Architectural Decisions

### Presentation Layer Framework

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Decision | Presentation Layer Framework | | Category | Strategic |
| Design Issue | Should we use a Client Centric or Server Centric Solution | | Id. |  |
| Issue or Problem Statement | There are different Presentation layer implementations to choose from. The solution should take in to consideration the highly responsive UI requirements of the applications, Multi browser support, Mobile enablement and future requirement of Offline access. | | | |
| Assumptions | SPI portal requires a very rich user interface and asynchronous communication between client (browser) and server.  SPI portal has to leverage several OOB features of the portal and integrate several applications to the portal, also it has to leverage the Personalization and Customization features of the portal  Page refresh between two disparate function calls are acceptable. | | | |
| Motivation | Provide a futuristic portal solution that can be support multiple devices and browsers and provide enhanced user experience. | | | |
| Alternatives | 1. Server Centric Client Side Rendering Portal | Hybrid solution, involving the benefits of the portal and rich UI experience.  Spring MVC Portlets based on JSR 286 will be used in Server Side  JQuery Mobile and AngularJS will be used on the client side for providing rich UI  The Solution also leverages HTML 5 features of the browser | | |
| 1. Server Centric Server Side Rendering – Portal | Similar to option 1, but the content aggregation, UI rendering and navigation is server driven  Spring MVC Portlets based on JSR 286 will be used in Server Side | | |
| 1. Client Centric Single Page - Web | The solution does not involve the portal. But provides rich and responsive UI | | |
| Decision | Use Server Centric Client Side Rendering - Portal | | | |
| Justification | JQuery Mobile and Angular JS based Client Side framework along with Spring MVC Portlet meets the objective of having a highly rich and responsive UI and the benefits of the Portal. The solution offers a consistent configuration mechanism across all architectural layers and provides support for different devices and browsers.  The same combination of technology is proven in other divisions of Manulife. | | | |
| Implications | Not much of the existing Presentation Layer code can be reused. | | | |
| Decision Made by | MLJ, iGATE | | | |

### UI Layer JavaScript Library

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Decision | UI Layer JavaScript Library | | Category | Tactical |
| Design Issue | Should we use a IBM bundled DOJO libraries or user JQuery Mobile with Angular JS | | Id. |  |
| Issue or Problem Statement | WebSphere portal inherently supports DOJO as the JavaScript library for Rich Client interface. RAD also has tooling support for DOJO. JQuery Mobile is more popular than DOJO and is very well documented. JQuery Mobile is adopted in other Manulife divisions. | | | |
| Assumptions | There will be no conflict between the JQuery Mobile and DOJO libraries. DOJO can be disabled in WebSphere for the SPI portal pages without causing any impact. | | | |
| Motivation | Provide a highly responsive UI for all devices and browsers | | | |
| Alternatives | 1. JQuery Mobile with Angular JS |  | | |
| 1. DOJO |  | | |
| Decision | Use JQuery Mobile with Angular JS for the UI Layer implementation | | | |
| Justification | JQM is proven in Manulife environment. JQM is well documented and has relatively less learning curve for the developers. | | | |
| Implications | There may be unforeseen conflicts between JQM and DOJO | | | |
| Decision Made by | MLJ, iGATE | | | |

### Deployment Consideration (WPS and WAS versions)

|  |  |  |  |
| --- | --- | --- | --- |
| Decision | Deployment Consideration (WPS and WAS Versions) | Category | Strategic |
| Design Issue | The new portal is based on WPS 8.0 on WAS 8.0. AS-IS applications are deployed on WAS 7.0. The new system should allow for phased roll out and enable corporate channels to use the AS-IS system. There should be no disruption of services for Bank channels. | Id. |  |
| Issue or Problem Statement | Bank Channels and Agents who are involved in Corporate proposals should be able to access the AS-IS AgentWeb application. If the new business services are deployed on WAS 8 which also contains the Portal, it may not be able to reuse cost intensive utilities like SVF. | | |
| Assumptions | SSO from the portal to the AS-IS AGW application can be achieved through Federated Custom User Repository Implementation.  The new Business Services will be targeted for WAS 8.0 except the services involving SVF. Services catering to Reporting/Printing will be deployed on WAS 7.0 to leverage SVF licenses. | | |
| Motivation | To enable a TO-BE interim state for a phased rollout of Retail and Corporate functions of AGW through the portal. Retain the AS-IS WAS7.0 infrastructure for Bank Channels | | |
| Alternatives | 1. Launch the AS-IS AGW in a new browser window/tab by means of a hyperlink in the portal page | | |
| 1. Launch the AS-IS AGW in the Portal Page itself | | |
| 1. As a clipped portlet using WebClipping feature of WebSphere Portal | | |
| 1. By using IBM Web Application Integrator to embed the Portal Page Look and feel | | |
| Decision | Launch the AS-IS AGW in a new browser window/tab by means of a hyperlink in the portal page | | |
| Justification | There is no requirement for additional Hardware and Software. This approach serves as a risk mitigation and there is no disruption of AS-IS systems and allows for Phased Rollout of Retail and Corporate Functions. | | |
| Implications | There may be some additional effort involved in Migrating the Data from AS-IS system to the new system when the new system is rolled out for corporate functions and bank channels. | | |
| Decision Made by | MLJ, iGATE | | |

# SPI Application Architecture Definition

### Conceptual View

Manulife SPI portal solution is conceptualized as depicted in [section 1.2 Business Context](#_Business_Context).

The highlights of the solution are:

* The solution has to support all major browsers and also enable access via mobile devices.
* Rich Internet Application and Enhanced user experience by making use of AJAX based calls to the portal. The portal should only cause minimal screen refreshes there by optimizing the user experience and bandwidth utilization
* The portal will make use of federated user repositories for Authentication and Authorization. The user repositories of the existing applications that are brought under the portal will be reused by bringing them under a Federated User Repository. This approach will enable us to incorporate new and old applications in to the portal in a phased manner. There will be no user repository migration effort and end users will also not be prompted to change their password and other profile information.
* As a part of revamping exercise, the business layer of the application will be exposed as REST based WebServices.
* The data access layer of the applications will also be revamped to move the business logic entrenched in Stored Procedures to business services layer.

### Logical View

A tentative logical model allowing for phased roll out of the portal, with no disruption of services is depicted below:

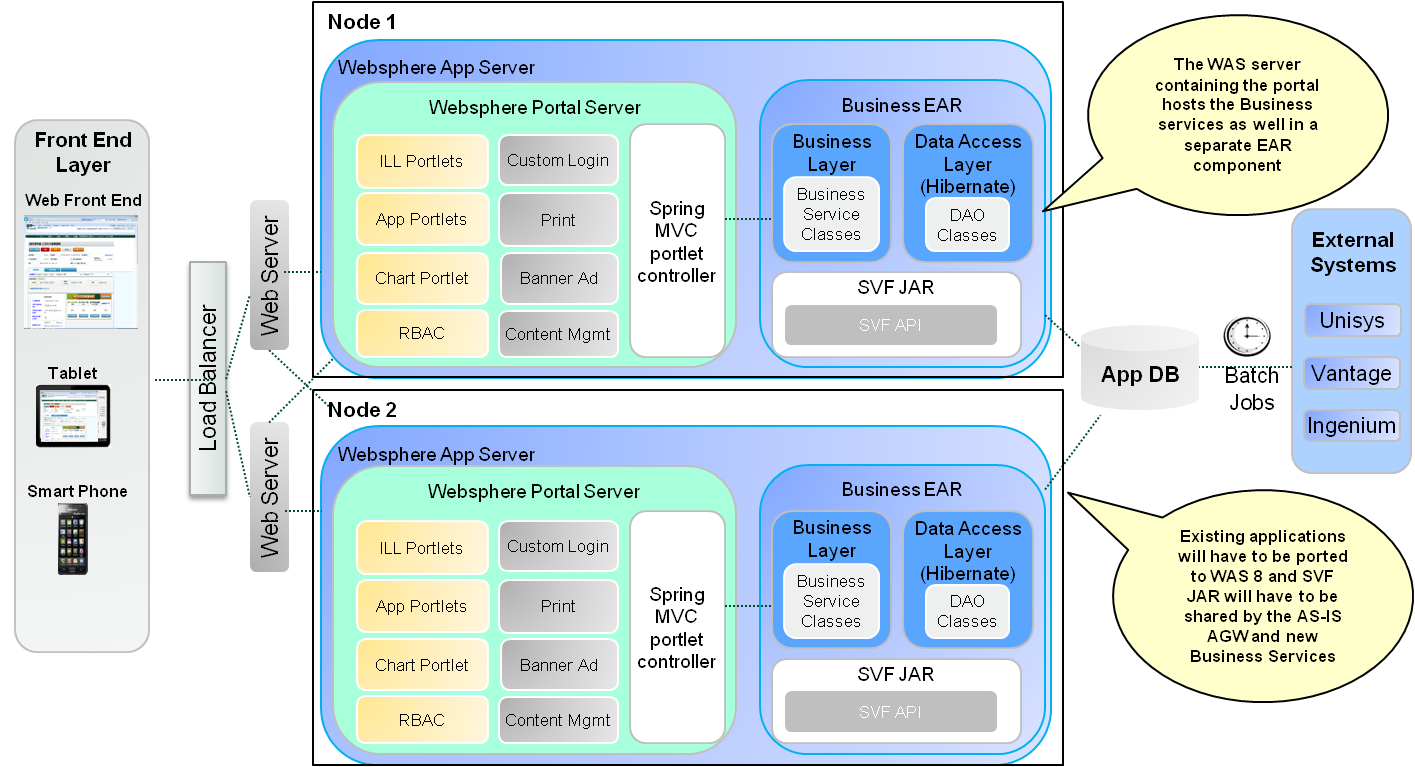


Fig 10: SPI Portal Logical Model WPS 8.0 / WAS 8.0

One of the key requirements is to allow a phased rollout of the portal for retail and corporate business. The AS-IS AgentWeb and other applications are deployed on WebSphere Application Server 7.0, where as the new SPI portal is intended to be deployed on WebSphere Portal Server 8.0 and WAS 8.0 infrastructure. The proposed solution requires service enablement of the Business Layer. At the same time, the existing applications should be retained for Bank channels and corporate business. This would mandate the co-existence of the AS-IS and TO-BE applications at least in the interim.

In addition there is a compulsion to re-use cost intensive tools such as SVF for reporting. As we are proposing a Phased rollout for the portal for corporate business, we will have to retain the AS-IS applications at least in the interim. There are also expensive software assets like SVF, which will have to be shared by the AS-IS applications and the new service enabled version of the applications. For this to be feasible, the AS-IS applications will have to be ported to the WAS 8.0 infrastructure.

The above logical model is perceived taking in to consideration the above aspects.

### Layered Architecture

The portlets in SPI Portal will be implemented using a layered architecture comprising of:

* Client Layer
* Presentation Layer (Portal)
* Business Layer
* Data Access Layer
* Data Layer

The diagram below depicts the high level components in each layer for Illustration Module.

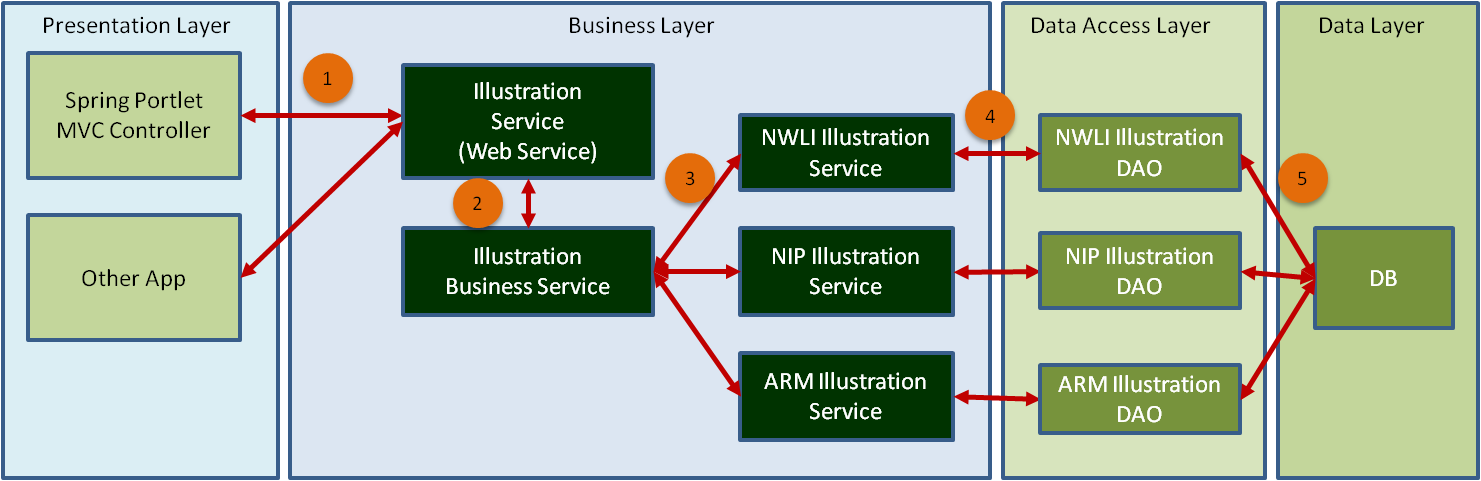


Fig 11: Layered Architecture (Illustration Module)

### Client Layer

The client layer corresponds to the Web browser and the UI screens. In the proposed solution, the client layer is implemented using HTML 5, JQuery Mobile and Angular JS.

### Presentation Layer

The presentation layer corresponds to the Portal and will be implemented using Spring MVC Portlet framework.

### Business Layer

Business layer exposes the core business functionality as REST based WebServices. The implementation will follow an interface/implementation approach based on Spring. A service may have dependency on other services. Coarse grained services which embed other services will act as façade to avoid chattiness.

### Data Access Layer

The DAO layer implementation also will follow an interface/implementation approach based on Spring. The Entity model will be derived from the Data Model using JPA tooling provided by the IDE (RAD/Eclipse). The Entity model will be further enriched by incorporating additional finder methods and Named Queries. An abstract GenericJPADAO class containing all the standard data manipulation methods will form the basis for all DAO classes.

### Data Layer

The data layer is revamped to accommodate the packaged product offerings. All the business logic embedded in stored procedures will be moved to the business layer.

### Service Enablement

Service Enabling the Business Layer implementation involves the following steps.

* Define a functional model by logically grouping the business functionalities
* For each functional component identified, derive fine grained interfaces by providing the well defined input / output structures
* The business method should implicitly convey its purpose and should have no affinity to the consuming application.
* Provide the implementation of the Business interfaces defined above
* Annotation driven approach will be followed to resolve dependencies of service.
* The next step is to expose the business interfaces as RESTful web services using web services annotations

### Packaging & deployment

The SPI Portlet applications will be layered as outlined in Section 10.3 Layered Architecture. The application code will be packaged to reflect this layering, with the intended responsibilities of each layer honored.

The business services layer along with service enablement component and the DAO layer will be packaged in to a separate project and deployed as a standard WAR component in the WebSphere Application Server. This service component has no affinity to the application that is consuming it. Spring configuration files and annotations will be used to resolve the dependencies.

The portlet war modules that form the presentation layer components will be deployed on WebSphere Portal Server. The portal layer implementation will follow Spring MVC Portlet framework based on JSR 286 standards.

### Conversion Guidelines

### Reusable components design

Business context for reusable components design is to avoid the duplicating logic between products and channels. Between channels, even though the standard components and core business validations are same, channel specific requirements will come into the account. To address this issue, we need to identify the channel specific requirements for current as well as future expansion channels.

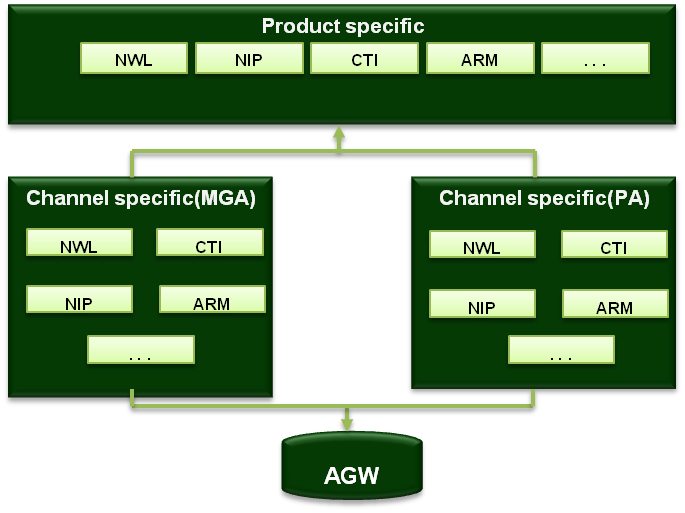


Fig 12: Reusable Components Design

In case of products, standard components & common business validations are replicated in multiple products to easy the development cycle. By refactoring the common functions/modules between products and create the common class, product specific implementation will be part of product specific requirements. This methodology addresses the maintainability issues in future.

Broad categories of reusable components are:

* Java classes ( Action , BO & DAO classes)
* Data base components ( stored procedures or complex query’s or static data )

**Java Classes**

**Existing system to SPI:** we can reuse the sales conditions, calculation & standard or business components associated to the product / application. Detailed guidelines will be identified at the time of implementation.

**Multiple products expansion:** common components between products will be extracted to the base classes to avoid duplication between products.

**Database components**

**Existing system to SPI:** Analyze the implementation details in the existing procedures / query’s to identify & implement the reusable components.

**Multiple products expansion:** Identify the common components like query’s or functions which are duplicated between various products.

**Multiple channel expansion:** In this case, we need to identify the product specific features between channels (not products). This gives us the channel specific requirements to take into the account while expanding to multiple channels.

### Use case View

The diagram below depicts the high level view of the SPI portal from business user’s perspective.

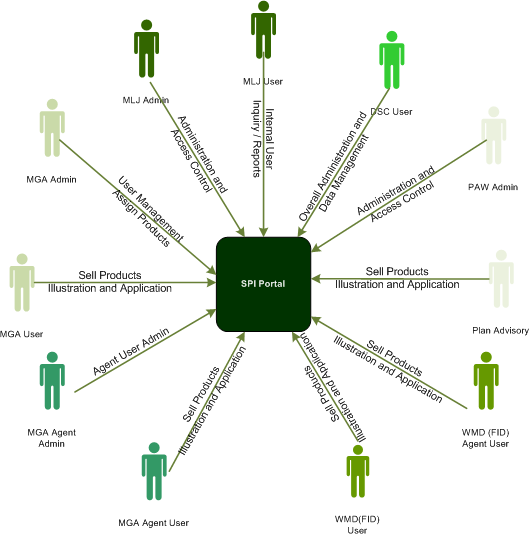


Fig 13: Use case View

Different categories of users and their functions are as below:

* **MGA Admin** – is a super user of MGA and responsible for MGA User Management and assign products to users
* **MGA Users** – are internal MLJ users who can sell products. They access the Illustration and Application modules through the SPI Portal
* **MGA Agent Admin** – are the administrator users for a Agency and responsible for Agent User management of the respective agency
* **MGA Agent User** – are the agency users who access the MGA Web application to sell policies and products. They use Illustration and Application modules for the same
* **WMD (FID) User** – are the Bank employees selling the Manulife products. They mainly deal with corporate policies. They access the SPI portal through the NTT Gateway.
* **WMD (FID) Agent User** – are the agents appointed by the Bank and access SPI portal to sell the Policies and Products. They also access the SPI portal through the NTT Gateway. They can perform illustration and application functions in the Portal
* **MLJ Admin** – are internal administration users and has overall administration rights on the portal. They mainly can monitor Applications and Illustration created through different channels
* **MLJ User** – are internal MLJ employees who access the SPI Portal for inquiry and reporting
* **DSC User** – is responsible for overall administration and maintenance. May use manual overrides to fix errors in the data.
* **PAW Admin** – Plan Advisory administrators is responsible for PA User Management and assign products to PA Users. PA users are maintained in the Manutouch LDAP user repository
* **PA User** – Plan Advisories are agents appointed by Manulife and access the Portal to sell plans and products. They use illustration and application modules of PA Web.

### Functional Use case View

The high level functions of the SPI portal are:

* Customer Management
* Plan Creation

The illustration below provides the functional use case model of the SPI portal

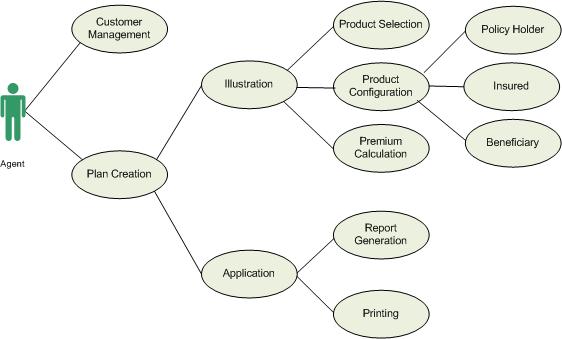


Fig 14: Functional Use Cases

As depicted, important use cases that form the SPI portal from an end-user perspective are Customer Management and Plan Creation. Plan creation can be further divided in to Illustration and Application. Illustration comprises of Product selection, Product Configuration and Consolidated Premium Calculation.

The Product configuration includes data capturing of Policy Holder information, Insured Information and Beneficiary Information.

The Application module comprises of Report creation and printing.

### Data View

Provided below is the tentative data model for the SPI portal. Detailed view of the data model is captured in the Excel file :<< hyperlink or Embed the ER Diagram XLS >>

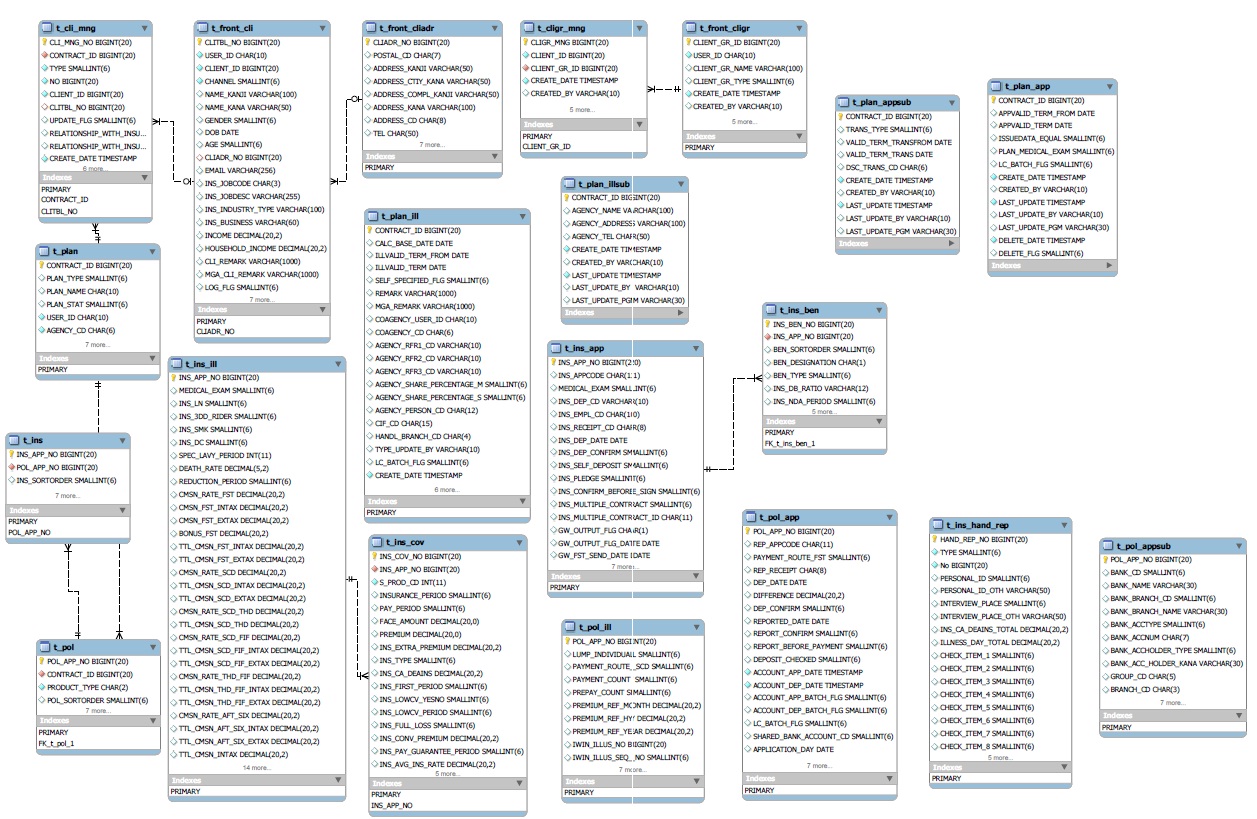


Fig 15: Data Model

As depicted the data model comprises of tables to capture the information pertaining to :

* Customer / Group
* Plan
* Illustration and
* Application

**Note:** The data model above is a tentative and is used to demonstrate the key entities that form the SPI services. For a detailed view of the data model, please refer to the link mentioned above.

### Security View

### Physical View

In order to support phased rollout, we need to consider two deployment models, one interim and one final TO-BE state.

The deployment model is based on Horizontal Scaling topology as depicted in the diagrams below. Highlights of the Topology are:

* Standard High Availability Design
* The design uses 2 Nodes federated in to one cell and clustered together
* The two nodes are usually (Not always) implemented in 2 geographically separated sites
* Each Site can take the full production load if necessary to minimize the impact of infrastructure outage
* IP traffic is routed and load balanced at either of the 2 locations
* Session affinity is used to route IP traffic to one site or the other
* Users do not switch between the sites unless there is a failure
* One Primary Database Server, Directory Server, Deployment Manager and Dispatcher (or IP Sprayer) are located in Site 1.
* Database Server, Directory Server, Deployment Manager are replicated to corresponding servers at Site #2 using native replication facilities.

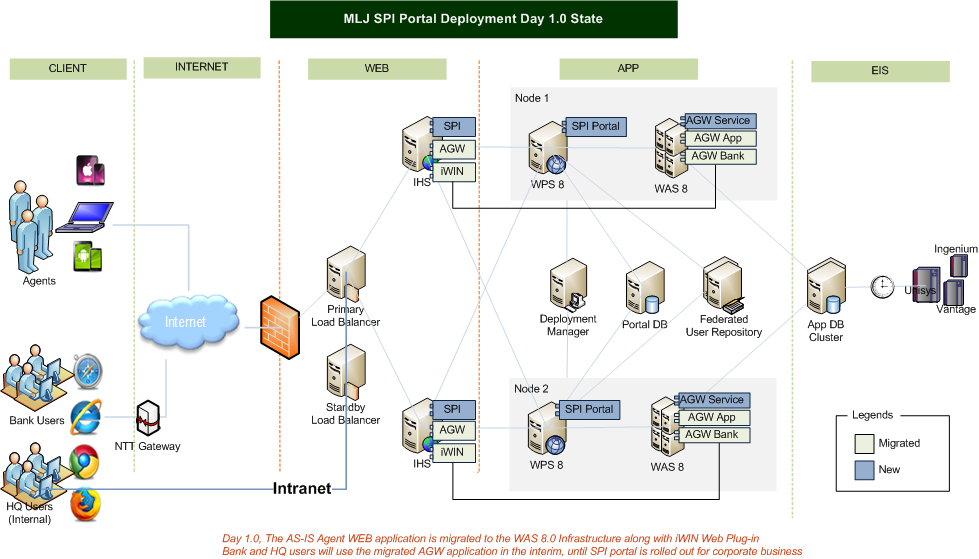


Fig 16: SPI Portal Day 1.0 (Interim State) Deployment View

In the interim deployment model, the AS-IS Agent WEB application is migrated to the WAS 8.0 infrastructure along with iWIN WEB Plug-in. Bank and HQ users will use the migrated AGW application in the interim, until SPI Portal is rolled out for corporate business. This deployment model will allow sharing of expensive software assets such as SVF.

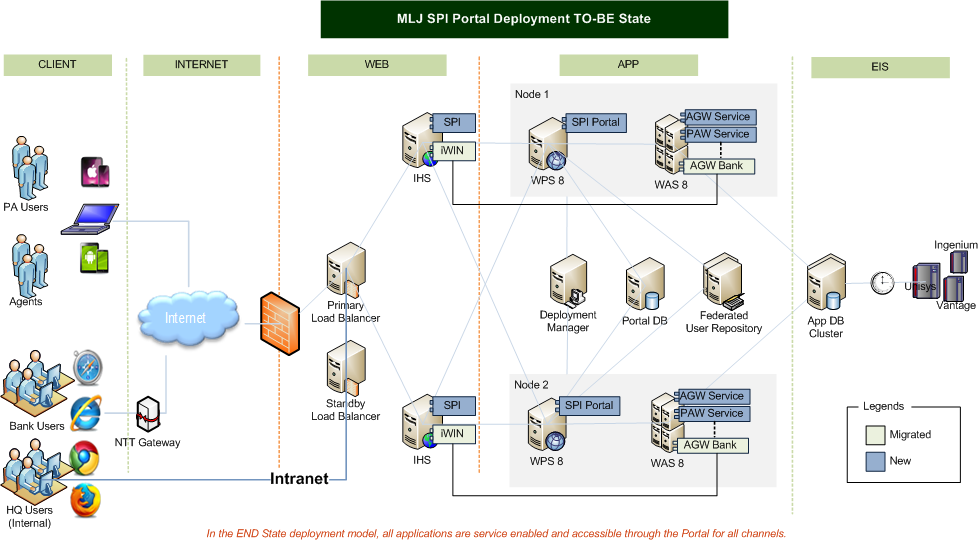


Fig 17: SPI Portal TO-BE state Deployment View

In the TO-BE sate, all applications are service enabled and accessible through the portal for all channels.

### Development and Test Environment

The illustration below depicts the Development and Test environment setup for the SPI Portal.

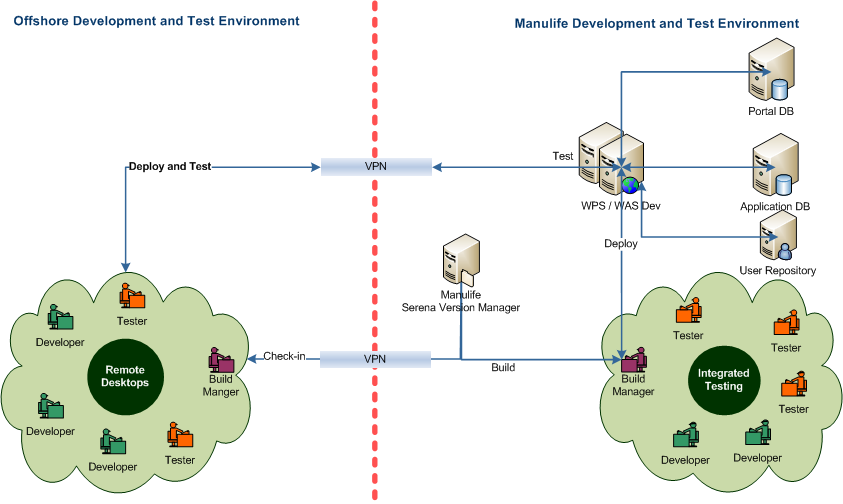
****

Fig 18: Development and Test Environment

The developer workstations of all the portlet developers will have Rational Application Developer installed, with integrated Portal Test Environment. The workstations of the Service component developers will have Eclipse with Tomcat installed.

The offshore developer will access their workstations in MLJ using Remote Desktop Connection via Citrix. Portlet developers can test their code on the integrated test environment. The Service Component developers can test their code in Tomcat.

Unit tested code will be checked-in to Serena Version Manager. ANT scripts will be used to build the WAS deployable. Tested portlet war modules can be directly deployed on to the DEV environment to perform an integrated testing. Integration tested portlet modules can then be promoted to TEST environment. Build Managers at Onsite / Offshore will be responsible for DEV / TEST deployment and ensure code quality.

# Risks, Assumptions, Issues, Dependencies and Constraints

### Risks

| **Ref No.** | **Risk Description / Mitigating Action** | **Risk**  **(Impact x Probability)** | **Impact**  **(1- Low**  **2 –Med**  **3 - High)** | **Probability**  **(1- Low**  **2 –Med**  **3 - High)** | **Owner** |
| --- | --- | --- | --- | --- | --- |
| 1 | Non-availability of Business Interfaces during development and testing of the Portal could pose a potential integration issue later / Can be mitigated to a certain extend by mocking the business services | 9 | 3 | 3 | Project Manager |
| 2 | Enhancements to existing application can proliferate due to the necessary changes to data structures introduced by new business requirements and business interfaces | 6 | 3 | 2 | Project Manager |
| 3 | Migration of existing user base to the Portal federation may become very involved and cumbersome, requiring co-ordination among infrastructure team and application owners. | 9 | 3 | 3 | Project Manager |
| 4 | Integration issues can be foreseen between existing applications and the portal | 6 | 3 | 2 | Project Manager |
| 5 | Co-existence of the AS-IS Agent Web and the Service Enabled version would introduce operational challenges. | 9 | 3 | 3 | Project Manager |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Table 31: Risks

### Assumptions

The following assumptions have been made during the creation of this document. If any of these are false then there will be change(s) to the requirements, subject to Change Control, once the requirements have been ratified.

| **Ref No.** | **Description / Impact** |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Table 32: Assumptions

### Issues and Unresolved Requirements

| **Ref No.** | **Issue/ Unresolved Requirement** | **Description** | **Impact**  **(1- Low**  **2 –Med**  **3 - High)** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table 33: Issues and Unresolved Requirements

### Dependencies

Dependencies of this project upon externals and dependencies of externals upon this project are listed here, together with impact if dependency not satisfied.

| **Ref No.** | **Description / Impact** |
| --- | --- |
|  |  |
|  |  |
|  |  |

Table 34: Dependencies

### Constraints

Any known factors that constrain potential solutions, such as budgets and timescales, are listed below.

| **Ref No.** | **Description** |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Table 35: Constraints

# References

### Standard Compliance References

### Best Practices References

# Appendices

### Offline Mode Options

One of the key requirements of the SPI portal solution is to provide provisions for offline mode access in the future. Extending the portal itself for offline mode access will not be a feasible approach. The key considerations for offline mode access are outlined in the section below:

### Considerations for Offline Mode Access

The important points that need to be considered for offline are:

1. How much of the SPI portal solution can we re-use
   1. The assumption as of now is that the UI part of SPI portlets can be re-used more than 90% as it is client centric.
   2. The URLs corresponding to Asynchronous calls to the portlet controller will have to be mapped to something in the client to get the JSON data
   3. Much of the Business Services Implementation of the SPI portal can also be reused while Syncing offline data with the server
2. What all functionality should be provided in offline mode
   1. We may not need the entire functionality to be made available in offline mode.
3. Should the UI be purely browser based
   1. How do we overcome the limitations of the Local Store in HTML 5 (5 MB)
   2. An approach involving a light weight web server in the client side will have to be formulated
4. How do we provide/implement the server side logic in the offline mode (Example: intermediate persistence, Reference Data etc)
   1. Will HTML 5 suffice or do we need any client side component to be pre-configured ?
   2. We will also have to explore IBM provided solutions for mobile and offline access ([IBM WebExperience Factory](http://www-03.ibm.com/software/products/us/en/ibmwebexpefact/),  [IBM Worklight](http://www-03.ibm.com/software/products/us/en/worklight/), [Lotus Expeditor](ftp://ftp.software.ibm.com/pub/lotusweb/portal/offline_portal_flyer.pdf), etc)
5. How do we synchronize the offline data to the server once in online mode (incremental / full update etc)

### Conceptual Solution Model

The illustration below is a conceptual solution model for the offline mode access:

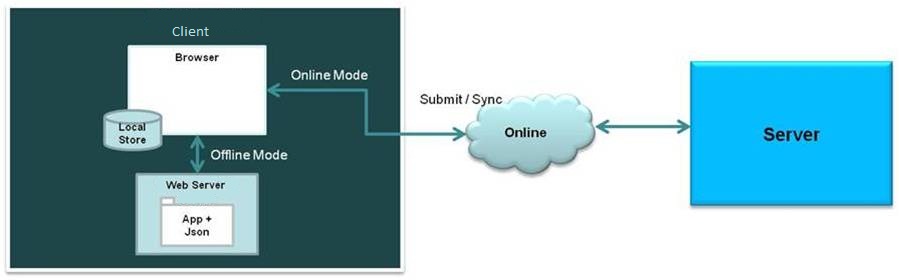


Fig 19: Offline Mode Conceptual Solution

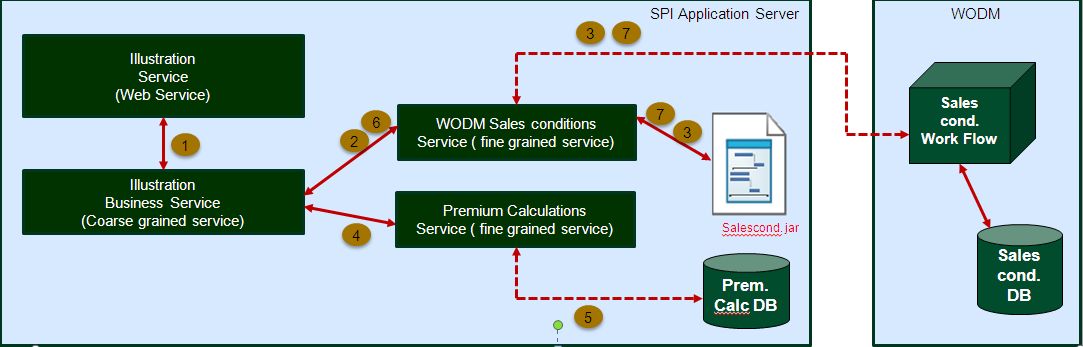
* The solution involves a light weight Web Server component in the client side capable of hosting the offline application in the client side.
* The solution leverages the Local Store of browser for storing offline data
* In the offline mode, the browse uses the local web server for all business functions
* The data is persisted in JSON data format in the WebServer
* Whenever the client goes online, the data is synchronized with the server

### Capacity Planning

### Testing Strategy

### Batch Mode

### WODM integration with services layer



WODM services are categorized into two services

* + - * Sales conditions check
      * Premium validations check

In case of multi product packaging, WODM services are requested once and executes for all products. Even if the validations fail during the execution, continues the execution for all the products.

**Sales conditions check**: Sales conditions check service executes the sales conditions & face amount for the given product.

**Premium validations check**: Premium validations check service executes the premium range checks & job code range checks for the given products.

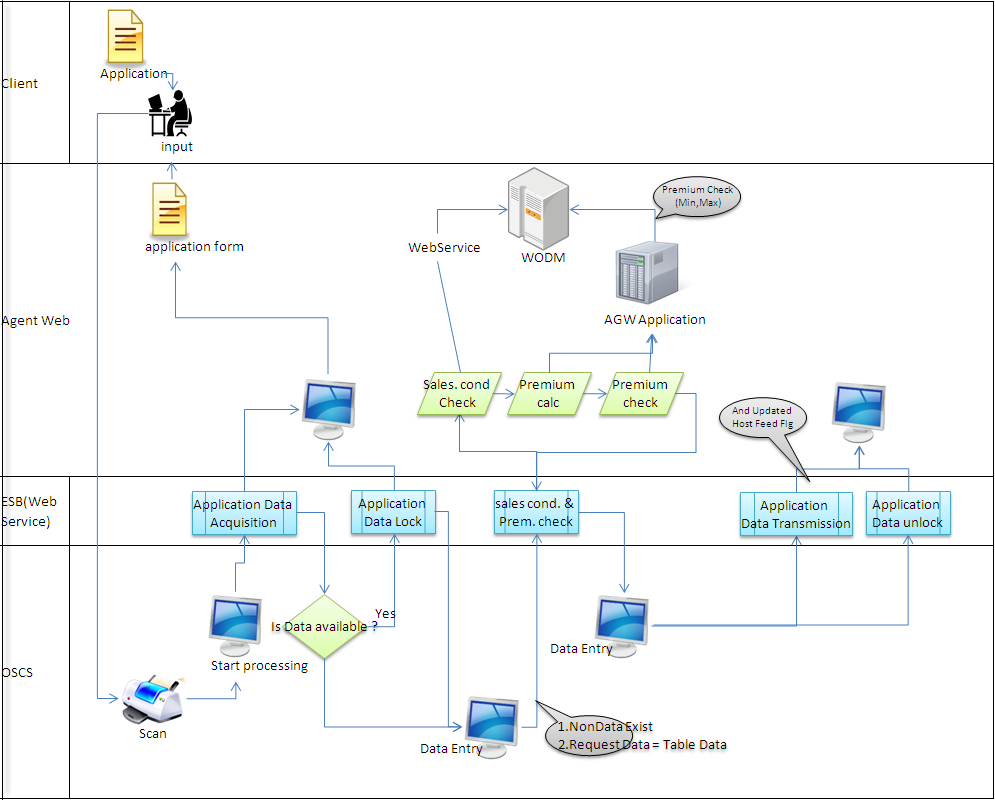
Illustration Web Service calls the Illustration Business Service to validate the sales conditions and premium calculations for the selected products.

1. Illustration Business service receives the request from Web Service and invokes the Sales Conditions check fine grained service to execute the sales conditions & face amount check.
2. Sales conditions FG service invokes WODM web service to perform the validations.
3. WODM responds the results in xml format ( success case return code as 0 & sales conditions validations failure case with error keys & descriptions )
4. If no errors during the sales conditions and face amount validations, invokes the premium calculations FG for premium calculations.
5. Premium calculations utility class returns the premium values to the FG service.
6. After the premium calculations, Illustration CG service invokes the Sales conditions FG to perform premium checks.
7. Sales conditions FG invoke WODM services to perform the premium checks.
8. Illustration CG service returns the response to Web servicer either the premium values or error keys with descriptions.

**Note: based on the performance, either web service call or bundling the jar file will be finalized to invoke the sales conditions.**

### Payment Gateway Interface and WODM Integration

Below diagram depicts the PGI interface for double entry.



**Service Provider:** Agent web

**Service Consumer:** OSCS

**Mediation:** ESB

Agent web exposes the below services through ESB for the application data modifications.

* Get Application Data
* Application Data Lock / unlock
* Sales Conditions & premium check
* Application Data feed

1. Once the application form is scanned, OSCS initiates the process.
2. Invokes Get Application Data service to verify whether the application is data is available in the agent web system or not.
3. If the data is available, invokes the Application data lock service to lock the plan (status change from 2 to 3) and sends the form for the data entry.
4. If the data is not available, sends the request to the data entry.
5. After the data entry, invokes the sales conditions & premium check service to calculate the premium. During the sales conditions check, it invokes pre sales conditions, premium & post sales conditions (premium range checks) and returns the result.
6. After invoking the sales conditions & premium calculations service, if any changes are required, sends for the data modifications and invokes Application data feed service to update the data in agent web system.
7. After the data upload, release the lock and changes the status from 3 to 4 at agent web system.